sceye SDK (OCX)



This SDK offers all the great basic features to scan documents with sceye. It comes with a preview window of what the user has placed in the scan-area of sceye and allows to control all the basic features like scan-mode, dpi, light and laser and all the other features of sceye. With the basic SDK all the comfortable features are available - and even better - it's free of charge.

For the more professional needs we offer also our Enterprise features like sceye-Autoscan, WTS-support and direct memory access for fastest image processing.

At a glance:	
Component:	OCX / .Net component
Component name:	
sceye_main.ocx	pure ocx with preview image and all scan-features
sceye_menucam2.dll	.Net component including sceye-main.ocx plus build in menu
InteropCamControl.dll	OCX wrapper for .Net component
Support for:	Visual C++, C#, Basic, any OCX-capable programming language
Requirements:	The SDK is for Windows operating systems (XP, Vista, Win7, Win8)
Note:	This component(s) delivers each scan as a single image file - color and greyscale as jpg file with definable compression - black and white scans as tif (CCITT) files.
Youtube video:	http://www.youtube.com/watch?v=wWGWyk0NUQY

At the following pages you'll find a full interface describtion of methods, events and registry switches as well as example code for different environments.

If you want to include sceye as an OCX into your application you can use the core component (sceye_main.ocx) or the more comfortable wrapper for the .Net component (InteropCamControl.dll). The second comes with all the features of the .Net component - see following lines.

If you are working with .Net (VB, VC, C#) we recommend to use the pure .Net component (sceye_menucam2.dll). It comes with an embedded menu (which can be hidded) and can automatically save and restore the last settings of the User. The integration this way is very easy and very powerfull the same time.

The methods and events are the same for OCX and .Net component.

- sceye SDK (OCX)
  - sceye SDK - Methods
    - Main methods
      - InitCameraExt
      - GetCameraDetails
      - GetConnectionStatus
      - PauseStream
      - ProceedStream
      - ShowMenu
      - ExitControl
    - Snapshot methods
      - SetColorMode
      - SetContrast
      - GetContrast
      - SetSaveDPI
      - SetSaveColorMode
      - SetSaveAddParam
      - GetLastFrameExt
    - Laser and Light and Brightness methods
      - LaserSwitch
      - GetLaserStatus
      - LEDSwitch
      - GetLEDStatus
      - SetBrightness
      - GetBrightness
      - GetAutoBrightnessSteadyState
    - Misc methods (less important)
      - RegisterUserLicence
      - SetSelection
      - GetSelection
      - WhiteBalance
      - SetAutoDetectionSensibility
      - GetAutoDetectionSensibility
      - SetAutoDetection
      - GetAutoDetection
      - GetAutoSelection2
      - SetAutoDetectionPauseObj
      - SetAnimation
      - ShowBrightnessSlider
      - ShowContrastSlider
  - sceye SDK - Events
      - Initialized
      - CamClosed
      - CamOpen
      - CamError
      - ConnectionStatusChanged
      - LiveStreamOn
      - LiveStreamOff
    - Enterprise features
      - NewObjectDetected
      - NotifyImageReady
      - NotifyImageReadyOnMemory
    - Misc Events (less important)
      - ReRegisterUserLicence

Registry settings
- sceye service
  - [HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/Sceye_service]
  - [HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/Sceye_service/Controlling]
- sceye ocx
  - [HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/sceye_ocx/AutoCrop]
  - [HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/sceye_ocx/Features]
- Twain related registry settings
  - [HKEY_CURRENT_USER/Software/SilverCreation/sceyeX/Twain]

Old and obsolete functions

EXAMPLES:
- Basic integration of the .Net component (example in VB.Net)
- Basic Integration of sceye_ocx (example in C#)
- sceye Autoscan Example

sceye SDK - Methods

Main methods

InitCameraExt

The most important function of the ocx. Its purpose is to build up required internal data structures and to start the threads for acquiring the live image. Any user of our SDK needs a valid Licence to initialize the ocx.
Parameter1 = Name of the Company + Product_Postfix
Parameter2 = valid number for the LicenceName

GetCameraDetails

P_ID = ProductID of the different models
7736 - 2nd and 3rd generation devices - these devices have (no serialnumber)
7737 - hot upgrades 7736 to the firmware of the 4th generation - (no serialnumber)
7738 - 4th generation device (with serial number)
7750 - 5th generation device with 10 megapixel sensor (with serialnumber)
Firmware - Version of the Firmware (just for internal use)
Serialnumber of the device - if available (devices of 4th generation or higher)

GetConnectionStatus

Returns the connection status of the camera. Important states are 0,3 and 4. State 1 and 2 are intermediate results for internal purposes.
retval = 0 / UNPLUGGED
retval = 1 / CONNECTED
retval = 2 / DOWNLOADED
retval = 3 / ALIVE
retval = 4 / ERROR
Hint: It is recommended to implement the ConnectionStatusChanged event. This function is especially for compatibility with older versions of sceye.

PauseStream

Stops drawing the preview window. The data transfer between camera and ocx continues to run in the background (autobrightness continues to work). The purpose of this function is to minimize processor demand when the control is not visible.

ProceedStream

Resumes the drawing of the preview window after a previous PauseStream. This function is normally used when the previously invisible control gets visible again.

ShowMenu

This method determines if the control shows the menu or not. Additionally you can define an application name. If you do the control keeps it current settings persistent between two program starts. The settings are stored in the registry. (Just supported by the .Net component and the ocx-Wrapper for it.)

ExitControl

The most important function for exiting the control. Its purpose is to stop the running threads and free the used memory.

Snapshot methods

SetColorMode

Sets the color mode for the preview image.
Parameter1: 1 = color, 2 = greyscale, 3 = black-and-white

SetContrast

Allows to set a contrast value between 0 and 100 - this contrast value is used for the live preview and the snapshot.

GetContrast

Returns the currect contrast setting.

Recommendation: We suggest using a Slider right under the Live-image. The contrast setting is a very powerful option to control the image quality and also very easy to use as the result is shown in the live-preview. Even if you don't want to perform a classic contrast algorithm we recommend to use the value 20. This makes you more independent regarding the light situation.

Background details:
The contrast value is not only performing a simple contrast function.
Value = 0 means no image modfication at all
Value 0-20 is enabling the dynamic white compensation (formally part of the SetSaveAddParam function)
Value 20-100 is adding a classic contrast function to the active dynamic white compensation

SetSaveDPI

Parameter1: resolution for the snapshot - valid values for SaveDPI: 100 / 200 / 300 (dpi)
Must be called before calling GetLastFrameExt.

SetSaveColorMode

Parameter1: color mode for the snapshot - valid values for SaveColorMode: 1 = color, 2 = greyscale, 3 = black/white
Must be called before calling GetLastFrameExt.

SetSaveAddParam

Sets some additional parameters (like Filter, Quality, Compression). This function must be called immediately before the function GetLastFrameExt.
Parameter1 is a parameter which combines several settings on a bit-level.
Bit0-3: compression rate of the jpeg file (color und greyscale) 1 (small file, low quality) to 10 (large file, very good quality)
Bit4: [0;1] [value: 16] dynamic white compensation
Bit5: [0;1] [value: 32] smooth filter
Bit6: [0;1] [value: 64] contrast enhancement
Bit7: [0;1] [value: 128] text filter
values above are not supported
example: jpg-compression 5 / dynamic white compensation 16 / contrast enhancement 64 - so the value for SetSaveAddParam = 5+16+64 = 85
Must be called before calling GetLastFrameExt.

GetLastFrameExt

This is the main function for taking snapshots. If the laser is enabled it will be turned off automatically during the snapshot and turned on again afterwards. The functions returns – determined by the AddParam parameter – immediately (asynchronously, after 1-2 sec) or after the processing of the snapshot (synchronously, can take some time depending on the filter settings). If the asynchronous processing is selected the ocx will fire a NotifyImageReady event after the processing of the snapshot is finished.
The user can proceed taking snapshots even before the NotifyImageReady event is fired. The tasks are stored in a fifo (first-in-first-out) structure and processed one at a time (with firing NotifyImageReady-Event for each processed picture).
ATTENTION: The NotifyImageReady-Event is not fired when you select synchronous processing (The GetLastFrameExt function returns when the picture process is finished.)
Parameter1: filepath, where the snapshot will be stored
Parameter2: method of post-process (0 = asynchronously, 1 = synchronously)

Laser and Light and Brightness methods

LaserSwitch

Parameter1: manually turns Laser on (state = 1) and off (state = 0).

GetLaserStatus

returns the current laser status - 1=on / 0=off

LEDSwitch

Turns the LEDs on (state=1) and off (state=0)

GetLEDStatus

Returns the current status of the LEDs - 1=on / 0=off

SetBrightness

Changes the brightness settings (shutter times) of the camera.
Parameter1: values for brightness - 1-27 (explicit values from dark(1) to bright(27)), 65536 (activates auto-brightness)

GetBrightness

returns the current brightness value. If auto-brightness is enabled it returns the brightness value (1-27) which is currently set by the auto-brightness function.

GetAutoBrightnessSteadyState

This function tells you weather autobrightness is still in progress changing the settings or the final brightness value is reached.
retval = 1 / Autobrightness ok
retval = 0 / brightness to dark
retval = 2 / brightness to bright
retval = -1 / error (no autobrightness active or other error)

Misc methods (less important)

RegisterUserLicence

Depending on the serial-number of a sceye and the DevLicence, a UserLicence becomes necessary. Register the UserLicence with this function.
retval = 0 - no camera connected
retval = 1 - Userlicence valid
retval = -1 - UserLicence not valid
Parameter1 = UserLicence
Calling this function without a connected camera returns 0. Please check for the ReRegisterUserLicence – Event which is fired if the User connects a camera. Depending on the return-values (valid user licence or not) other applications restrictions can be done.

SetSelection

Sets a region of interest on the image - same as a manual selection.
retval = 1 / Selection successfully
retval = 0 / size mismatch error
Parameter1: value between 0 and 344
Parameter2: value between 0 and 344
Parameter3: value between 0 and 460
Parameter4: value between 0 and 460

GetSelection

To get the region of interest - b.e. if the user made a manual selection on the preview image.
retval = 0 / no region selected
retval = 1 / regione selected - check the Parameters for the values
Parameter1: not used
Parameter2 & 3: values between 0-344
Parameter4 & 5: values between 0-460

WhiteBalance

There are 2 modes available to handle Whitebalance.
1) Automatic Whitebalance: By default the automatic whitebalance is enabled.
It can programmatically disabled by calling: ***CamControl1.SetSpecialFeatures(200, 0)***
It can programmatically enabled by calling: ***CamControl1.SetSpecialFeatures(200, 1)***
Alternatively the whitebalance can be calibrated by a white paper.
2) Calibrated Whitebalance
- disable first the automatic whitebalance
- Inform the user to place a white paper under the sceye
- Call: ***CamControl1.SetSpecialFeatures(7, 0)*** //this performs the manual calibration
- Inform the user that the calibration succeded

SetAutoDetectionSensibility

The method SetAutoDetectionSensibility accepts an integer value as parameter.
It determines how often a region must have the same detected corners in order to start capturing.
The following values are accepted:
3 - minimal value (very fast capturing, but lots of false detections)
5 - default value
10 - recommended maximum value
20 - absolutly maximum value

GetAutoDetectionSensibility

returns the current setting of the Sensibility...

SetAutoDetection

Parameter1: 0=auto-detection disabled / 1=auto-detection enabled / 2=auto detection and event (NewObjectDetected) enabled

GetAutoDetection

retval = 1 / autodetection enabled
retval = 0 / autodetection disabled

GetAutoSelection2

This functions gives you access to the results of the autoselection (also called autodetection) routine of the ocx.
Parameter1: returns the status of the autoselection - 0=autoselection not active - -1=no valid object found - 1=valid object detected
Parameter2: Angle of the detected object (just on the base of the green corners)
Parameter3-10: Coordinates of the 4 detected corners

SetAutoDetectionPauseObj

This functions sets the dimension and tolerance of the PauseObject. The PauseObject will be announced with another status (-1) by the NewObjectDetected-Event.

SetAnimation

Any snapshot can be animated in the ocx. You can choose the direction of the animation as well as the speed.
Parameter1: Animation time: 1=deactivated, 2=fast, 3=normal, 4=longer, 5=longest duration
Parameter2: Animation direction: 1=left, 2=right, 3=up, 4=down

ShowBrightnessSlider

Toggles the display of the brightness slider. (Just supported by the .Net component and the ocx-Wrapper for it.)

ShowContrastSlider

Toggles the display of the contrast slider. (Just supported by the .Net component and the ocx-Wrapper for it.)

sceye SDK - Events

Initialized
Is fired when control is initialized successfully. Normally not used.

CamClosed
Is fired when camera lid is closed.

CamOpen
Is fired when camera lid is opened.

CamError
Is fired when a camera error occurred. Normally not used.

ConnectionStatusChanged
Is fired when connection status changes. Both the previous status and the new status are provided.
For state values see function GetConnectionStatus.

LiveStreamOn
Is fired when Stream is started. For example: the user has opened the lid of the camera or StartStream was called.

LiveStreamOff
Is fired when Stream is stopped. Opposite to LiveStreamOn.

Enterprise features

NewObjectDetected
Is fired to indicate that a new object is detected under the camera. (To enable this event, the function SetAutoDetection must be called with the parameter 2).
Status: 1=new valid object found -1=Pause object found -2=invalid object found
ObjCounter: Each new valid object is getting a new number.
AutoBrightnessSteadyState: Indicates the status of the AutoBrightness - see also the interface GetAutoBrightnessSteadyState.
NumValidCorners: value between 0-4 - indicates the number of valid (green) corners.

NotifyImageReady
Is fired when post processing of the image (identified by the filename) is finished. (see chapter 3 – GetLastFrameExt)

NotifyImageReadyOnMemory
The event is fired after finishing all image optimization steps and before saving the image to a file. If you set the handle-flag on true, the ocx will not save the image.
If you have to do any post-processing with the image like OCR, multipage-tiff or something else, this is the fastest and most comfotable way to do it.
Important hint: You have to call the method GetLastFrameExt with the second parameter=1 which lets you perform a synchron processing (without using internal threads). You will just get the event in this context.

Note: Also the WTS support is just available with an enterprise license. However - for this feature no method or event is necessary. This feature operates transparently.

Misc Events (less important)

ReRegisterUserLicence
Is fired if the User connects a new camera. Status = 0 / invalid Userlicence --- Status = 1 / valid Userlicence

Registry settings

The sceye components are designed in a way that they will run with default settings if no registry-values are found. That's the default-situation at the end of the installation procedure.

Some changes of the users are stored in the registry under HKeyCurrentUser - Software - Silvercreations but another hundred options are invisible. These options allow a wide range of customization if needed. Some of these functions are described in this chapter.

A few words about the architecture:
Sceye-service is taking care for the direct communication with the HW. This service provides the preview-images and the snapshots for all other components like Twain and OCX.
The sceye_ocx / .Net component is the visible component in the sceye-explorer, the twain-driver and the customized solutions.
The twain-driver allows any twain supporting application the use of sceye right away. There are also some settings especially for the guiless-Twain-access.

Annotation: Any major release stores its registry settings into an own folder. Please replace sceyeX by the current version sceye5.

sceye service

[HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/Sceye_service]

- Firmwareversion = 0 [read only]
- BeepForDriverStartStop = 0 [0,1 beeps for debug purpose]
- TimeMeasurement = 0 [0,1 time-measurement for sceye-explorer]
- WriteLog = 0 [0,1 Write logfile into user-temp-folder / filename: sc_sceye_service.log]

[HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/Sceye_service/Controlling]

- ServiceStatus = 1 [0,1 read only]
- cmd-Shutdown = 0 [0,1 external way to shut-down service]
- ConnectionStatus = 0 [0,1 connection status of sceye-HW]
- RunningStatus = 0 [0,1 service running or pause]
- cam_arm_always_open = 0 [0,1 useful to ignore the arm-position-sensor]

sceye ocx

[HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/sceye_ocx/AutoCrop]

- SaveCornerDetection [0,1 debug purpose]
- MaxCutOff = 12 [0-n max pixels to cut off for autocrop]
- MinCutOff = 3 [0-n min pixels to cut off for autocrop]
- UseSecondDetection = 1 [0,1 process second corner detection]

[HKEY_CURRENT_USER/Software/SilverCreations/sceyeX/sceye_ocx/Features]

- DebugInfo = 0 [0,1 show debug info in preview window]
- TimeMeasurement = 0 [0,1 time-measurement for sceye-explorer]

Twain related registry settings

Not all functions of sceye are supported by the Twain-Interface. Especially the very important filter functions are not known by Twain. We allow using these filters also for a guiless-Twain-access, just by setting the following registry-keys.

[HKEY_CURRENT_USER/Software/SilverCreation/sceyeX/Twain]

- GUILESS_LASER [0,1 / on/off]
- GUILESS_AUTOCROP [0,1 / enables, disables autocrop]
- GUILESS_WHITEBALANCE [0,1 / enables, disables white booster]
- GUILESS_FILTER [0,1,2 / enables, disables filter, 0=no filter – 1=contrast filter – 2=smooth filter]
- GUILESS_MANUALSELECTION [0,1 / use a manual selection]
- GUILESS_MS_LEFT [0-width / left point]
- GUILESS_MS_TOP [0-height / top point]
- GUILESS_MS_RIGHT [0-width / right point]
- GUILESS_MS_BOTTOM [0-height / bottom point]
- GUILESS_COLORMODE [1,2,3 / 1=colour – 2=grey – 3=bw]
- GUILESS_RESOLUTION [100,200,300 / resolution]
- GUILESS_CONTRAST [0-100 / contrast (recommended: 30)]

Annotation: The last 3 guiless-settings are dominating a possible Twain-setting.
Attention: Using Autocrop in a GUILESS situation means you have no visual feedback of the Autocrop decision. Please make sure that the User is aware of this feature and that the scanning area is free of distracting objects.

Old and obsolete functions

These functions are just for internal purposes or the functions are unused and obsolete and they are just in the interface because of compatibility reasons.

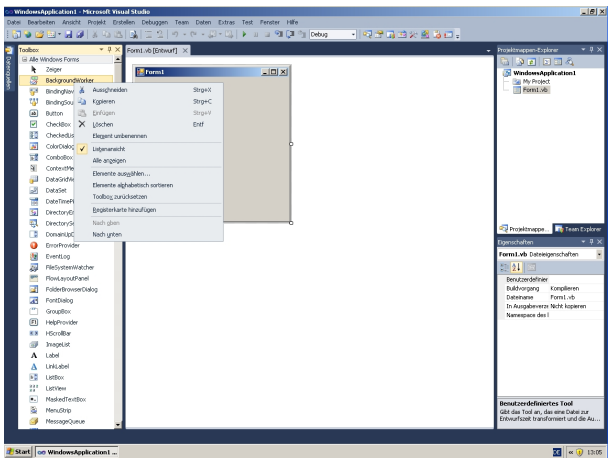
- SetLensZoom
- PerformReset
- void GetServiceStatus(long* nInstances, long* ConnectionStatus, long* ButtonStatus, long* StreamRunning, long* FrameCounter, long* BadFrameCounter);
- void SetSpecialFeatures(long index, long value);
- void EnableAutoLaser(...) - obsolete function, just available for interface compatibility reasons
- SetLightProfile - obsolete function - just available for interface compatibility
- long GetSerialNum(); - This is an older function - please use GetCameraDetails instead.
- GetShutterWidth
- SetShutterWidth
- SetImageHeight
- GetImageHeight
- SetImageWidth
- GetImageWidth
- CalibrateCam
- GetCalibrateAssistStatus
- SetMemoryBitmap
- StartLive / StopLive - obsolete - this function calls ProceedStream or PauseStream internally

EXAMPLES:

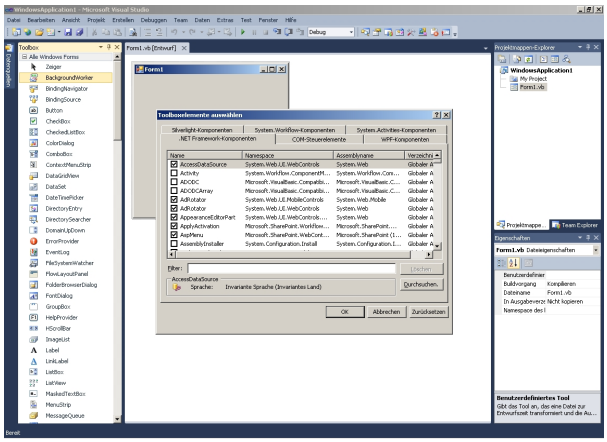
Basic integration of the .Net component (example in VB.Net)

Starting with version 5.7 of our software we provide a very easy and convenient way to integrate the Sceye camera in your own .NET project. Please follow these steps for a quick and easy integration of sceye into a demo-project.

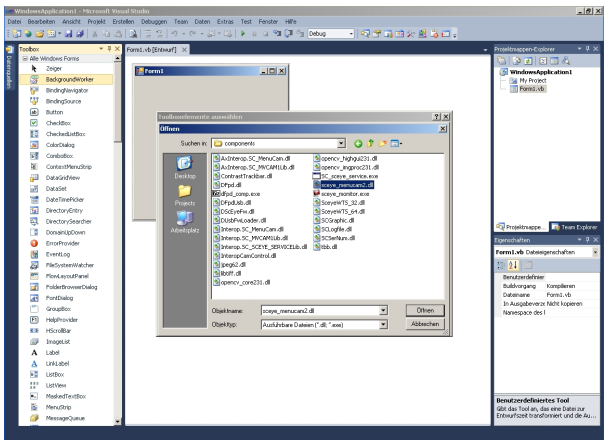
- Install the latest Sceye software on your development system
- Run the software once to verify it is installed correctly and working.
- Start a new project in Microsoft Dev Studio. B.e. use a Visual Basic Windows Forms project.
- Rightklick on any item of the toolbox



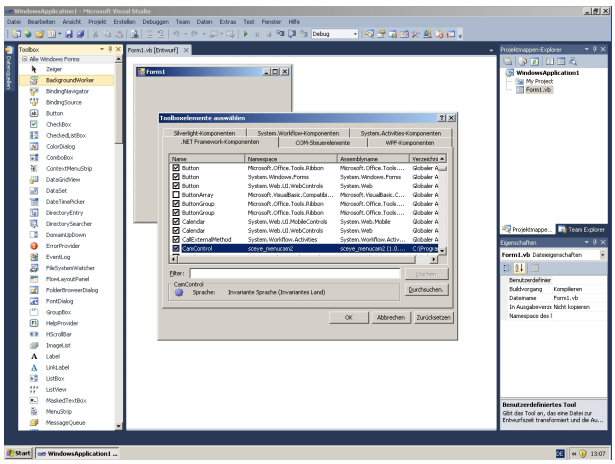
- Click on file selection.



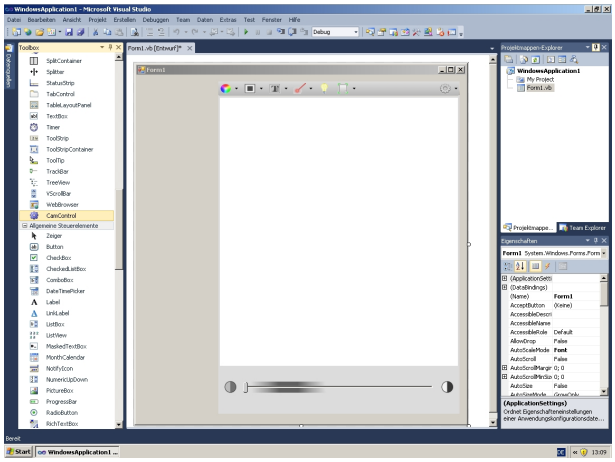
- Browse for the file sceye_menucam2.dll in the components folder of the sceye software.



- Finish control selection.



- Add the new control to the form.



- Add two buttons to the form.
  one for “Init camera”
  one for “Do snapshot”
- Add the code for the two buttons.

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        'put your licence code here
        Dim strLicenceName As String = "Your-Licence-Name"
        Dim strLicenceNumber As String = "Your-Licence-Number"

        CamControl1.InitCameraExt(strLicenceName, strLicenceNumber)
        CamControl1.StartLive()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

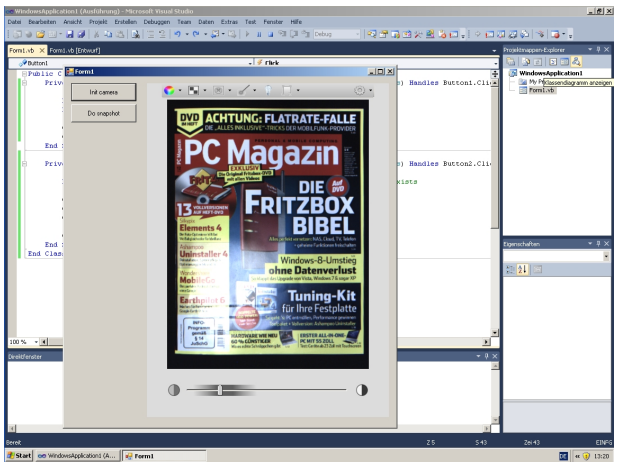
        Dim strFilename As String = "C:\temp\test_snapshot.jpg"     'make sure the folder exists

        CamControl1.SetSaveColorMode(1) 'color snapshot
        CamControl1.SetSaveDPI(200)     '200 dpi
        CamControl1.SetSaveAddParam(7)  'medium jpeg compression quality

        CamControl1.GetLastFrameExt(strFilename, 1)  'synchronous snapshot
    End Sub
End Class
```

- Run the project.

- Click on “Init camera”



- Click on „Do snapshot“
- Snapshot appears in the filesystem.

As you can see - with a few lines of code you can have a first integration within minutes.

Basic Integration of sceye_ocx (example in C#)

The most common used component is our standard ocx (filename: InteropCamControl.dll).
All features related to a single scan are implemented in this component and available through the interface.
The following lines are showing how to integrate the ocx by using the most important functions.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Text;
using System.Windows.Forms;
using System.Runtime.InteropServices;

namespace WindowsFormsApplication1
{

   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();
         string LicenceName;
         string LicenceKey;

         strLic1 = "YourLicName";
         strLic2 = "YouLicKey";
         sceye_ocx.InitCameraExt(ref LicenceName, ref LicenceKey);

       sceye_ocx.SetColorMode(3);          //b/w mode
       sceye_ocx.SetBrightness(65536);       //auto-brightness
       sceye_ocx.LaserSwitch(1);          //laser on
       sceye_ocx.LEDSwitch(1);            //led light on
       sceye_ocx.SetAutoDetection(1);       //autodetection on
       sceye_ocx.SetContrast(30);          //contrast 30 (0-100)

       sceye_ocx.StartLive();             //start live picture
       //Start and stop livepreview depending on the visibility of the sceye_ocx
       //StopLive is reducing the processor-time
      }

      ~Form1()
      {
       sceye_ocx.StopLive();
         sceye_ocx.ExitControl();
      }

      private void ScanImage_Click(object sender, EventArgs e)
      {
         sceye_ocx.SetSaveDPI(300);       //300 dpi
         sceye_ocx.SetSaveAddParam(85);     //filter and compression parameter
       sceye_ocx.SetSaveColorMode(3);     //save as b/w-image

         string strFilename;
         strFilename = "C:\\test.jpg";
         sceye_ocx.GetLastFrameExt(ref strFilename, 1); //perform snapshot syncron
      }
}
```

sceye Autoscan Example

Note: This feature requires an enterprise licence of the sceye-SDK. Otherwise the Event will not be fired.

```
Private Sub sceye_ocx_NewObjectDetected(ByVal sender As Object, ByVal e As
AxSC_MenuCam.__Cam_NewObjectDetectedEvent) Handles  sceye_ocx.NewObjectDetected

   If (e.status <> -1) Then  'Valid Object found
      Timer_Snapshot.Enabled = True

      If e.autoBrightnessSteadyState <> 1 Then
         System.Threading.Thread.Sleep(300) 'Sleep 300 ms
      End If

      If e.numValidCorners < 2 Then
         'less the two valid (green) corners detected
      End If

      'Perform snapshot or (not recommended)
      'start timer to change the context for performing the snapshot (recommended)

   ElseIf e.status = -1 Then '-1=Pause object found
      'Pause-Object detected - Perform special handling if necessary...
   End If
End Sub
```