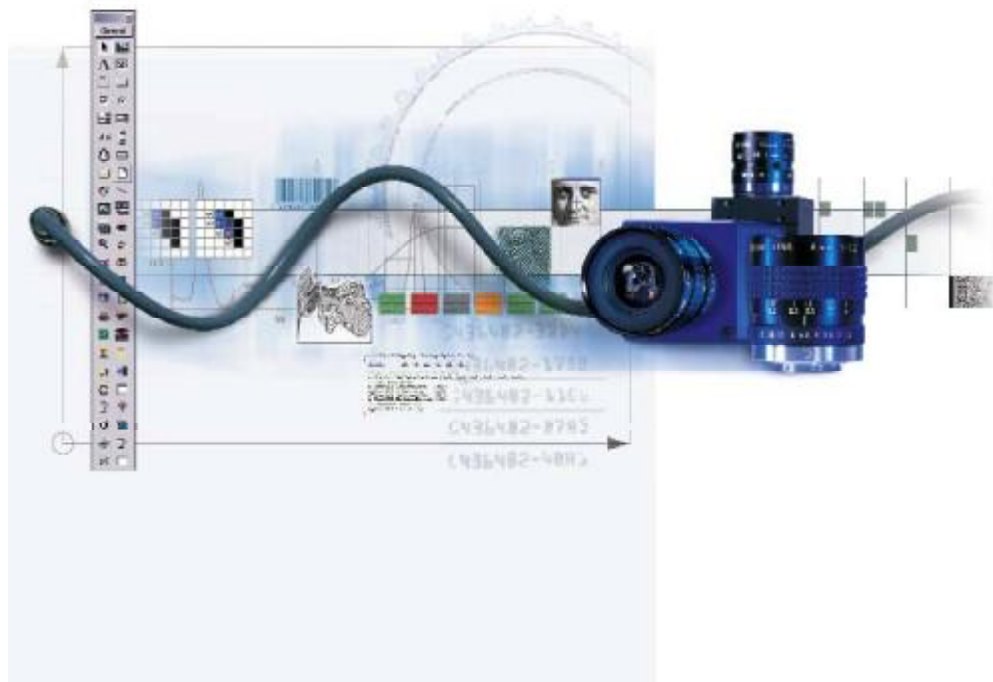


Getting Started Guide

► First steps with Common Vision Blox



STEMMER®
IMAGING

www.commonvisionblox.com

► **Copyright**

For technical support in the US, contact:

Image Labs International
(406) 585-7225 x 110 (800) 785-5995
www.imagelabs.com sales@imagelabs.com

Copyright © 2002 STEMMER IMAGING GmbH, Puchheim.

All rights reserved and subject to change.

STEMMER, Common Vision Blox, Windows, Visual Basic, Visual C++, C++Builder, Visual Studio.net, Visual C#, Delphi are registered trademarks.

Any duplication or transfer in printed, electronic or photographic form, whether in full or in part, or any translation into another language requires the prior written consent of STEMMER IMAGING GmbH.



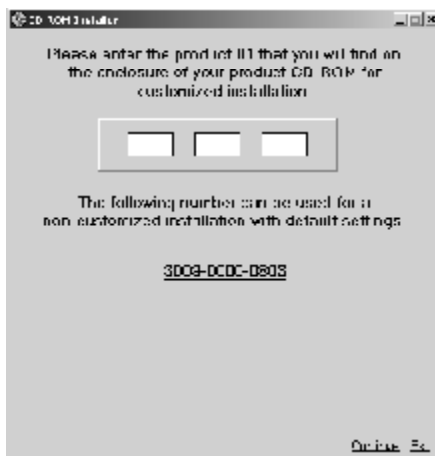
Installing Common Vision Blox	1
Software protection concept	3
Registering tools	6
What is Common Vision Blox?	7
Getting started	9
Your first Common Vision Blox application	10
Visual Basic®	10
Starting the Common Vision Blox Application Wizard	10
Selecting a stored image or a frame grabber	10
Creating and executing the application	12
Adding a tool to the application	13
Borland Delphi®	16
Starting the Common Vision Blox Application Wizard	17
Configuring the Wizard	18
Executing the application	20
Adding a tool to the application	20
Visual C++®	23
Starting the Common Vision Blox Application Wizard	23
Configuring the Wizard	23
Completing the application	26
Executing the application	27
Adding a tool to the application	29



Installing Common Vision Blox

Insert the Common Vision Blox CD into your CD ROM drive. If the Common Vision Blox installation program does not start automatically, you can call the file AUTORUN.EXE from the CD. You will be asked to supply the product ID that is printed on the CD. Alternatively, you can use the product ID suggested by the program which allows you to install the program with default settings.

This Getting Started Guide is based on the standard installation which is installed using the ID 3009-0000-0803. If you are using a product-specific ID, the stages described below may differ slightly in appearance.

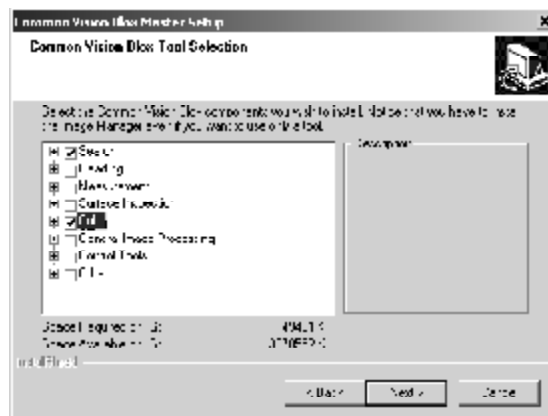


Select CONTINUE and, depending on the product ID you have entered, a customized CVB installation menu will be displayed. Read the “Installation Instructions” and then start the “Common Vision Blox Setup”.

The next thing to do is to enter the directory in which Common Vision Blox is to be installed and the name of the program group in which the shortcuts of Common Vision Blox files are to be created.

If you have any queries regarding the installation of the frame grabbers, consult the relevant Quickstarts and Release Notes or contact the technical support team of your supplier.

Now you can select the Common Vision Blox tools you wish to install.



Software protection concept

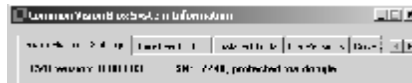
The software protection concept from Common Vision Blox allows protection of each individual tool and the application itself using the same software protection mechanism, for example, a hardware dongle.

The majority of CVB components will run in demo mode, but with restricted functionality. Virtually all functions are available, but from time to time dialog boxes appear reminding you that this is a demo version. In order to carry out tests under real conditions it is also possible to order a 14-day unrestricted demo license (see the section on “Node-locked licenses”).

There are a number of different methods of implementing software protection for CVB:

- ▶ using a hardware dongle from Rainbow
- ▶ using a “node-locked license” which is coupled to the PC hardware
- ▶ using a frame grabber, for example, CVA Leo
- ▶ using the smart camera CVS ThinkEye

In the “Common Vision Blox System Information” you can see whether the Image Manager is licensed. If this is the case, the serial number and the selected method of licensing will be shown. The CVB tools are licensed by generating a unique “magic number” using the CVB serial number.



Software protection using a dongle

The dongle is plugged into the parallel port of the PC and is supplied together with the Common Vision Blox Image Manager. The magic numbers of the CVB tools are generated on the basis of the serial number of the dongle.

Software protection using “node-locked licenses”

“Node-locked” means that the licensing of Common Vision Blox is not tied to a dongle, but instead to an item of hardware that is generally available on the computer, for example, the CPU, network adapter or the hard drive.

The “node-locked license” is requested and delivered by email. The magic numbers of the CVB tools are generated using the serial number and the license file with the magic numbers is also sent to the customer by email.

In order to generate a request for a node-locked license, start the “License-Commander” application. (This application can be found in the *START* menu after Common Vision Blox has been installed.)

This type of licensing only works under Windows 2000 or XP and also allows licensing for a limited period. At the moment, the following node-locked license options are available for Common Vision Blox:

1. *General node-locked license*

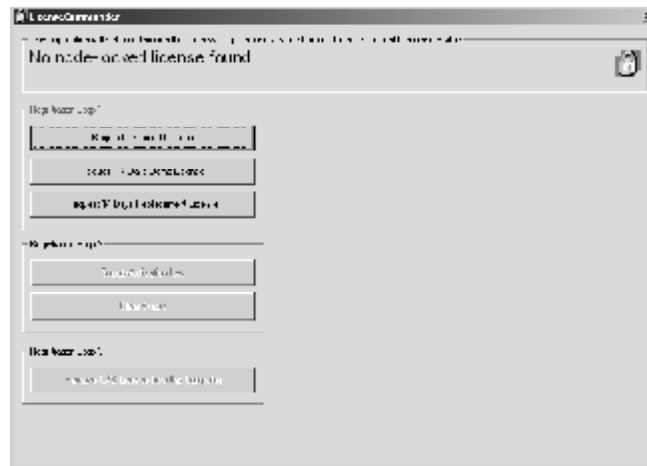
This is a license which works in a similar way to a dongle and is not limited to a specific period of time. The CVB serial number that is generated by this form of licensing is assigned by STEMMER IMAGING. This form of licensing is intended primarily for larger customers who wish to use Common Vision Blox on a large number of systems or in customer-specific applications.

2. *Demo license*

The demo license is designed for customers who wish to test one or more of the Common Vision Blox tools without restriction before purchasing the products. The license is valid for 14 days after generation and always produces the CVB serial number 999.

3. Dongle replacement license

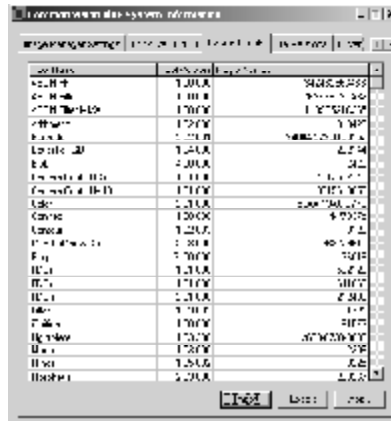
The dongle replacement license is designed to replace a CVB dongle that was purchased in the usual manner and which does not function correctly. During generation, the customer specifies the number of the faulty dongle, and STEMMER IMAGING produces a node-locked license based on the CVB serial number of the dongle. The license is valid for 14 days. During this time you will receive a replacement dongle from us.



More information about how to generate and activate node-locked licenses can be found in the Common Vision Blox online Help system.

Registering tools

If you try to call a function of a tool that has not yet been registered, a dialog box appears asking you to enter the magic number. You can also enter the magic number using the “Common Vision Blox System Information” program. This can be found in the *START* menu of Common Vision Blox.



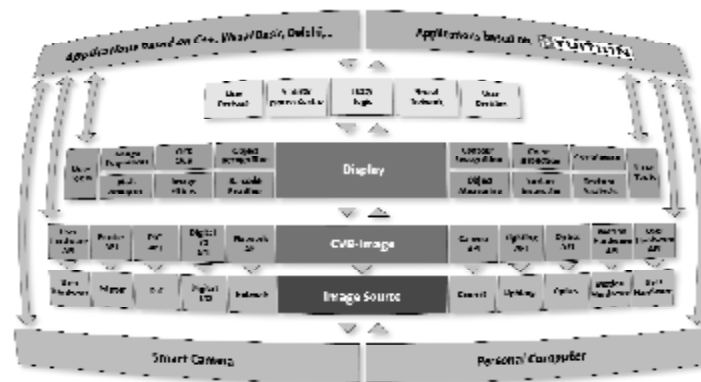
The “System Information” lists all the installed tools. When you double-click a tool, a dialog box appears in which you can enter the magic number. Now click the **APPLY** button to register the tool.

If you have been supplied with a license file (a “LIC” file) which contains the magic number, you can load this directly using the **IMPORT** button and activate it using the **APPLY** button.

What is Common Vision Blox?

Common Vision Blox is a package combining a large number of configurable software tools with which you can implement your own applications quickly and simply. Thanks to the tremendous increase in performance of PC technology, modern software algorithms now reach speeds comparable to or better than those that were previously achieved only with special image processing hardware.

Common Vision Blox is a universal software platform for machine vision applications and is based on powerful image processing algorithms. The tools offer a degree of speed, accuracy and reliability that can only stem from extensive integration experience. Common Vision Blox is ideally suited to OEMs, system integrators and end users who require a maximum of flexibility.



Ease of use, flexibility and high performance

Common Vision Blox makes use of the advantages of visual programming environments. Microsoft Visual C++®, Microsoft Visual Basic®, Microsoft Visual C#.Net®, Microsoft VC.Net®, Microsoft VB.Net®, Borland Delphi® and Borland C++Builder® form the basis for users who are thus able to create their own graphical user interfaces (GUIs) and integrate their application into these.

The **Image Manager** forms the cornerstone of Common Vision Blox and comprises **ActiveX controls** and **DLLs**. Both the drivers of all the supported frame grabbers and the tools are linked by the Image Manager. The tools are also available as ActiveX controls and DLLs.

Hardware independence

Common Vision Blox supports a wide variety of hardware components, whether they are illumination components, cameras, digital I/O components or frame grabbers. Hardware from various manufacturers can be added to an existing system at any time; only a few lines of code must be changed to do this. Thus, components can be combined individually as required and can be integrated even after the development phase has been completed. This open standard means that developers can concentrate fully on their application and therefore react more quickly to new or changed market conditions.

Open programming environment

New and existing algorithms can be incorporated into Common Vision Blox in the form of ActiveX controls or as DLLs. Since Common Vision Blox uses the programming environments Microsoft Visual C++®, Microsoft Visual Basic®, Microsoft Visual C#.Net®, Microsoft VC.Net®, Microsoft VB.Net®, Borland Delphi® and Borland C++Builder®, users can also access thousands of algorithms from third-party vendors and then use these in their own applications parallel to Common Vision Blox. With the aid of the Common Vision Blox software protection concept, new tools can be encrypted with a software key in order to protect the know-how invested.

Getting started

Well over a hundred **sample applications** with source code in all of the programming languages supported are installed together with Common Vision Blox. The sample applications demonstrate what Common Vision Blox and the tools can do. They can also be used as a foundation for your own applications.

The following chapter shows you how to use the **Common Vision Blox Application Wizards**, which enable you to create a basic framework for a Common Vision Blox application in Microsoft Visual C++®, Microsoft Visual Basic® and Borland Delphi® in a very short time.

The CVB package also includes programming examples and instructions on generating a Common Vision Blox application for the following languages: Microsoft Visual C#.Net®, Microsoft VC.Net®, Microsoft VB.Net® and Borland C++Builder®.

A printed **manual** and, in particular, the **online Help system** provide additional information about Common Vision Blox and how to create your own applications.

Your first Common Vision Blox application

Visual Basic®

Starting the Common Vision Blox Application Wizard

Start Visual Basic® and select “Common Vision Blox Application Wizard” from the NEW PROJECT dialog box.



Selecting a stored image or a frame grabber

In the next dialog box you can select a stored image (*.bmp), a video file (*.avi or *.mpg) or a frame grabber driver (*.vin in the “drivers” folder) with which the application is to be started.



ADD 'GRABBER' CONTROL

Adds the Common Vision Blox Grabber control to your form. The control offer simple access to the frame grabber and supports, for example, the switching of ports, as well as access to various frame grabbers and triggers.

ADD 'OPEN IMAGE' BUTTON

Creates a button on the form with which images or frame grabber drivers can be loaded.

ADD 'GRAB' CHECKBOX

Creates a check box on the form for activating grabbing by the frame grabber.

GET AREA IN IMAGE SNAPED

Determines for every new image whether the AOI (area of interest) only is to be used as the basis for further processing rather than the complete image.

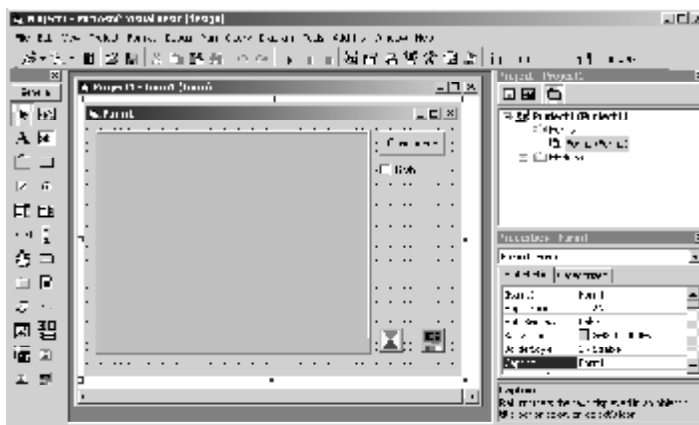
ADD IMAGE MANAGER LIB'S TO PROJECT

Adds the declaration modules to the project so that the DLL functions of the Image Manager can be used.

Click the NEXT> button to continue.

Creating and executing the application

The next dialog box confirms correct selection and creates the application when you click the FINISH button.



Start the application you have created by pressing <F5>.



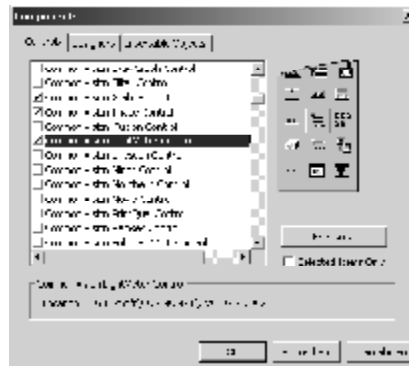
Adding a tool to the application

Using the “LightMeter” tool as an example, we would now like to demonstrate how easily tools can be integrated into the application. You must have the LightMeter tool installed on your system to be able to do this.

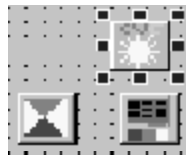
Click the list of tools with the right mouse button and select the COMPONENTS... item from the pop-up menu.



In the Components dialog box, select the Common Vision LightMeter control and confirm with OK.



Position the Common Vision LightMeter control on your form.



Press <F7> to view the code. Now insert the following lines in the *SetupUI()* procedure after the *TODO* comment.

```
CVLightMeter1.image = CVimage1.image  
CVimage1_ImageSnaped
```

The first line assigns an image object to the LightMeter control; the second line calls the procedure *CVimage1_ImageSnaped* in which the processing is to take place. The complete *SetupUI()* procedure should look like this:



We must now adapt the *CVimage1_ImageSnaped* procedure in which the processing is to take place. Insert the following lines after the line “TODO: add your processing here”:

```
CVLightMeter1.Execute  
Form1.Caption = CVLightMeter1.SVMean
```

The first line calls the *Execute* method of the *LightMeter* control; this executes processing. The second line serves to display the result in the application's name field. Here, we would like to display the mean brightness of the image. We therefore assign the *SVMean* property of the *LightMeter* control to the form's *Caption* property. The complete *CVimage1_ImageSnaped* procedure should look like this:



When the application has been started by pressing <F5>, the mean brightness of the image appears in the application's name field.

If you load the driver of a frame grabber instead of an image, you will see how the value changes for every new image.

It has taken just four lines of code to integrate a tool into our application!

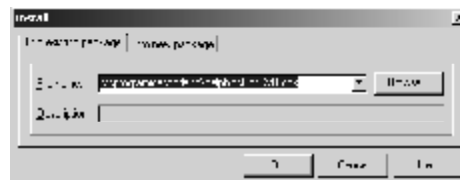
Borland Delphi®

Before you can start the Wizard, the ActiveX controls of the Image Manager must be linked into your development environment. Select the **IMPORT ACTIVEX...** item from the **COMPONENT** menu. Select "Common Vision Image Control" and select a palette page or enter a new name for it.



Click **INSTALL**.

In the **INSTALL** dialog box which is then displayed, select **INTO EXISTING PACKAGE** and confirm with **OK**.





Confirm compilation of the package in the next dialog box. Close the PACKAGE dialog box and click Yes in response to the query whether Delphi should save the changes.

Repeat the entire procedure with the “Common Vision Display Control” and the “Common Vision Grabber Control”.

Starting the Common Vision Blox Application Wizard

Start Borland Delphi® and then under FILE - NEW... the object gallery. Select the PROJECTS tab and mark COMMON VISION BLOX WIZARD.

Confirm with OK.



Configuring the Wizard

In the next dialog box, activate the OPEN BUTTON and GRAB CHECKBOX check boxes.



OPEN BUTTON

Places a button on your form with which an image or the driver of a frame grabber can be loaded.

GRABBER PROPERTIES

Places a button on your form with which the frame grabber properties can be set.

GRAB CHECKBOX

Places a check box on your form for activating grabbing by a frame grabber.

USE DIRECTX

Enables the use of DirectDraw® which allows images and overlays to be displayed quickly and without flicker. Under certain circumstances, problems may arise with DirectDraw® under Delphi.

Confirm with NEXT>.

In the next dialog box, specify the name of your project and the path under which it is to be stored.



When you have confirmed your entries with FINISH, a message appears to inform you that the resource file is being created. The project is now generated.

Executing the application

Start the application by pressing <F9>. To load an image or a frame grabber driver, click the OPEN IMAGE button and select an image (*.bmp), a video file (*.avi or *.mpeg) or a frame grabber driver (*.vin in the “Drivers” folder).



If you have loaded a frame grabber driver or a video file, you can display a live image by checking the GRAB box.

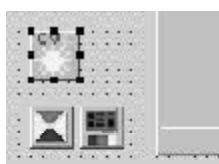
Adding a tool to the application

Using the “LightMeter” tool as an example, we would now like to demonstrate how easily tools can be integrated into the application. You must have the LightMeter tool installed on your system to be able to do this.

Import the Common Vision LightMeter control as already described above for the Image control and Display control.



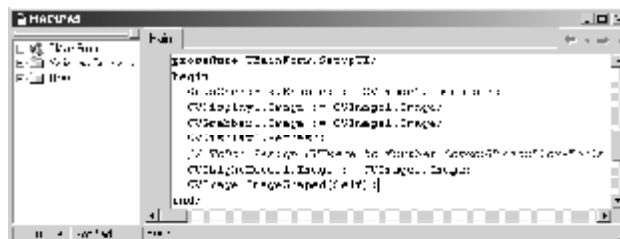
Now position the Common Vision LightMeter control on your form.



Insert the following lines in the *SetupUI* procedure after the *TODO* comment:

```
CVLightMeter1.Image := CVImage1.Image;  
CVImage1.ImageSnaped(Self);
```

The first line assigns an image object to the LightMeter control; the second line calls the procedure *CVImage1ImageSnaped* in which the processing is to take place. The complete *SetupUI* procedure should look like this:



We must now adapt the *CVImage1ImageSnaped* procedure in which the processing is to take place. Insert the following lines after the line *//ToDo*: Insert code to be executed for each snapped image”:

```
CVLightMeter1.Execute;  
MainForm.Caption := FloatToStr(CVLightMeter1.SVMean);
```

The first line calls the *Execute* method of the *LightMeter* control; this executes processing. The second line serves to display the result in the application's name field. Here, we would like to display the mean brightness of the image. We therefore assign the *SVMean* property of the *LightMeter* control to the form's *Caption* property. Since the *SVMean* procedure returns a floating-point number of the type *double*, but the *Caption* property expects a string, we must convert the value to a string using the *FloatToStr()* function. The complete *CVImageImageSnaped* procedure should look like this:

```
procedure TMainForm.CVImageImageSnaped(Sender: TObject);
begin
  (**** Event-Handler for ImageSnaped-Event ****)
  //ToDo: Insert code to be executed for each snaped image
  CVCLightMeter.Execute;
  MainForm.Caption := FloatToStr(CVCLightMeter.SVMean);
  CVDisplay.Refresh;
end;
```

When the application has been started by pressing <F9>, the mean brightness of the image appears in the application's name field.

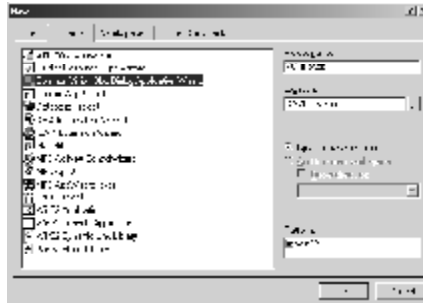
If you load the driver of a frame grabber instead of an image, you will see how the value changes for every new image.

It has taken just a few lines of code to integrate a tool into our Delphi application!

Visual C++®

Starting the Common Vision Blox Dialog Application Wizard

Start Visual C++® and select COMMON VISION BLOX DIALOG APPLICATION WIZARD from the FILE - NEW menu. Select a folder and specify a name for your project.



Configuring the Wizard

Select the options you require from Step 1 of 1 and then click FINISH.



DISPLAY SETTINGS

Some of the properties of the Common Vision Display control and of the control elements used can be specified here.

ENABLE DIRECTDRAW

Enables use of DirectDraw® which allows images and overlays to be displayed quickly and without flicker.

ENABLE SCROLLBARS

Activates the scroll bars for the display so that you are able to scroll through all parts of the image, e.g. when the image is zoomed.

GET AREA IN IMAGESNAPED

Determines for every new image whether the AOI (area of interest) only is to be used as the basis for further processing rather than the complete image.

ADD TOOL TIPS TO PROJECT

Activates the tool tips control. This enables you to assign short explanatory texts to all the control elements you use in your application, e.g. buttons, display etc.

PROCESSING IN IMAGE SNAPED

If online evaluation of all the images coming from the frame grabber is to be carried out, this can be catered for by adding the section of the program responsible for this evaluation to the *ImageSnaped* event of the Common Vision Image control.

NO PROCESSING

No program code is generated for processing.

PROCESSING USING LINEAR ACCESS

Program code is inserted for the *GetLinearAccess* function of the Utilities DLL. The function returns a pointer to the image data.

PROCESSING USING VPA

Program code is generated for the *GetImageVPA* function of the Image DLL. The function offers access to the Virtual Pixel Access Table and thus to the image data.

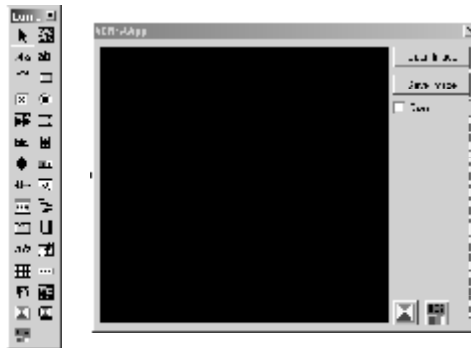
Completing the application

As a last step we must add the Common Vision Blox ActiveX components to the project. To do this, go into the PROJECT - ADD TO PROJECT - COMPONENTS AND CONTROLS menu. Double-click the REGISTERED ACTIVEX CONTROLS folder.

For the moment we only need the Common Vision Image, Display and Grabber controls in the application. Select each of the controls you need and click the INSERT button. Confirm the two Developer Studio queries that follow with OK.



The inserted controls can be used in dialogs only. In your work area select the RESOURCEVIEW tab and view the dialog generated. The Common Vision controls are now visible in the toolbar containing the control elements.



Member variables for the control elements have already been created by the Common Vision Blox Wizard in the MFC class wizard:

- ▶ *m_cvDisp* for the Common Vision Display OCX
- ▶ *m_cvImg* for the Common Vision Image OCX
- ▶ *m_cvGrabber* for the Common Vision Grabber OCX
- ▶ *m_bGrab* for the Grab check box

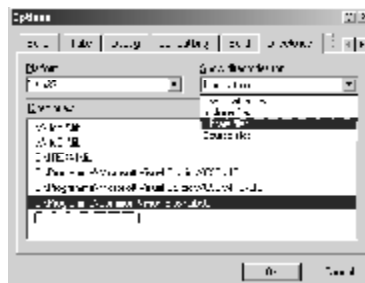
Executing the application

Compile and start the application by pressing the <F5> key. To load an image or a frame grabber driver, click the OPEN IMAGE button and select an image (*.bmp), a video file (*.avi or *.mpeg) or a frame grabber driver (*.vin in the "drivers" folder).

If you have loaded a frame grabber driver or a video file, you can display a live image by checking the GRAB box.



Make sure that you set a path to the Common Vision Blox folder \Lib\C in the TOOLS - OPTIONS - DIRECTORIES menu. If you do not, the compiler or the linker reports missing header or Lib files which means that an executable program cannot be generated.



Adding a tool to the application

Using the “LightMeter” tool as an example, we would now like to demonstrate how easily tools can be integrated into the application. You must have the LightMeter tool installed on your system to be able to do this.

Add the Common Vision LightMeter control to the project as already described above for the Common Vision Image, Display and Grabber controls.

Now position the Common Vision LightMeter control in your dialog box.

Call the MFC CLASS WIZARD to create the *m_cvLightmeter* member variable for the LightMeter control.



Go to the *SetupUI* function and insert the following lines after the *TODO* comment:

```
// TODO: Add your extra control notification handler code here
m_cvLightmeter.SetImage((long) m_Img);
OnImageSnapedCvImg();
```

The first line assigns an image object to the LightMeter control; the second line calls the procedure *OnImageSnapedCvImg* in which the processing is to take place.

We must now add the processing to the *OnImageShapedCvImag* procedure. We insert three more lines after the *TODO* comment. The complete procedure looks like this:



```
void CMainFrame::OnImageShapedCvImag...
{
    // TODO: Add your image processing code here.
    // For example, you can use the cvtColor function to convert the image to grayscale.
    // The following code snippet shows how to use the cvtColor function:
    // cvtColor(img, img, CV_BGR2GRAY);
    // The following code snippet shows how to use the mean function to calculate the mean brightness:
    // double mean_brightness = cv::mean(img)[0];
    // Finally, you can use the SetWindowText function to display the result in the title bar:
    SetWindowText(TEXT("Mean Brightness: %f"), mean_brightness);
}
```

The first line after the *TODO* line calls the *Execute* method of the *LightMeter* control; this executes processing. The second line serves to display the result in the application's name field. Here, we would like to display the mean brightness of the image. We therefore assign the *SVMean* property of the *LightMeter* control to the title bar of the dialog box. Since the *SVMean* procedure returns a floating-point number of the type *double*, but the *SetWindowText* function expects a string, we must pass the value as a string.

When the application has been started by pressing <F5>, the mean brightness of the image appears in the application's title bar.

If you load the driver of a frame grabber instead of an image, you will see how the value changes for every new image.



It has taken just a few lines of code to integrate a tool into our Visual C++ application!



Please contact your local distributor.

You can find a list of all distributors
of Common Vision Blox under
www.commonvisionblox.com
in the menu Sales/Distributors.

STEMMER[®]
IMAGING

► www.commonvisionblox.com



■ **STEMMER IMAGING GmbH**
■ **Gutenbergstr. 11**
■ **D-82178 Puchheim/Germany**
■ **Phone +49 (0) 89 / 80 90 2-0**
■ **Fax +49 (0) 89 / 80 90 2-116**
■ **E-Mail: info@imaging.de**
■ **www.stemmer-imaging.de**

STEMMER[®]
I M A G I N G