Instructions for Use

vcwin 2.33

Programming instructions for the Operating software vcwin

999.994.306.10-en-2.11

Impress

Publisher / Manufacturer	Vision & Control GmbH Mittelbergstraße 16 98527 Suhl, Germany Telephone: +49 (0) 3681 7974-0 Telefax: +49 (0) 3681 7974-33 *www.vision-control.com*
Number of the document	Programming instructions for the Operating Software vcwin 2.33 999.994.306.10-en-2.11
Date of first issue	05.12.1996
Date modified	2021-02-10
Copyright	© Vision & Control GmbH 2021

Copyright

Validity

The software is to be used for the purpose intended with the vision systems specified by Vision & Control GmbH.

By vision systems we mean:

- Intelligent cameras of the pictor series Nxxx
- Intelligent cameras of the pictor series M12xx/M14xx/M16xx/M18xx
- Intelligent cameras of the pictor series M24, MxxE, M48E Measure
- Intelligent cameras of the pictor series T300
- Multi-camera systems of the vicosys series 2300/2400/2480
- Multi-camera systems of the vicosys series 4300/4400
- Multi-camera systems of the vicosys series 5300/5400

1 FOREWORD

This programmer's manual describes the operating software vcwin, an external test program editor for the use with the vision systems (intelligent cameras and multi-camera systems) by Vision & Control GmbH.

In this manual, the use of the operating software vcwin and the most important functions are described.

The available functionality depends on the the vision system that is used. This document describes the options for full function scope.

Please refer to the hardware manual for more detailed information about the vision systems.

As well as the software, this programmer's manual is regularly improved and extended. The current version can be found on the home page at www.vision-control.com.

TABLE OF CONTENTS

2 THESE INSTRUCTIONS OF USE

We recommend introductory training at the Vision Academy. Before you start writing your own programs you should be familiar with the way the software works. The instructions of use provide a good basis.

Using the table of contents at the beginning or the index at the end of the manual you will quickly find the menu item or command that you want to know more.

Part 1 - Elements of the user interface

Part 1 describes the structure of the user interface.

The individual elements (menu bars, comand windows, docking windows ...) and their functions are explained.

Part 2 - Command Reference

The command reference describes the commands available in the operating software's **Command Selection**. The commands are structured into the following groups:

- Image commands
- Locate commands
- Evaluation commands
- Control commands
- Obsolete commands

Part 3 - Working with the Software

This section provides additional instructions for working with the operating software. The following chapters explain e.g. creating web pages with vcwin or working with a WebHMI.

Part 4 - Appendices

The appendix contains a description of the principles of detection and have program listings for a number of demonstration programs.

Additional Instructions of use

Instructions of use vision system

Please refer to the instructions of use delivered with each vision system manufactured by Vision & Control GmbH. These documents describes the resources available for your vision system.

ADVICE

As a rule, the various vision systems do not all have the same range of functions and do not support the full functional range of the operating software.

Instructions of use vicorem

These documents describes how the commands used are parametrized with the transfer between vicorem and the vision system. This manual is necessary for writing your own program modules.

3 CONVENTIONS OF PRESENTATION

The formatting and the symbols in this document help you to use the instructions of use and the device quickly and safely.

Product name

The instructions of use are valid for the products listed at the beginning. If not referred to expressly by the specific product name, the products described are referred to as "devices".

Advice

ADVICE
Indicates tips for users and useful additional information.

Enumeration

Indicates a listing of issues or possibilities:

Heading or topic of the enumeration

- Example list item 1
- Example list item 2

Operating steps

Listing of work steps, the given order of which must be followed. Numbering starts at 1. for each individual sequence.

1. Example work step 1

➔ Result work step 1

2. Example work step 2

Cross-references

Cross-references help you to make quick reference to particular sections of the manual, providing valuable supplements of information. The cross-reference shows you the page of the relevant section. Example: *see "These instructions of use", page 10*

Links

Links lead to documents outside the instructions for use. Expressly, no guarantee or liability is accepted for the accuracy and security of these documents (such as Internetpages). Links are only active in online-help and the PDF version and with a connection to the Internet. Example: *www.vision-control.com*

Spelling

Commands, menus and dialogues are highlighted in bold. The spelling **Utilities** indicated the menu **Utilities**.

References to subordinate entries are indicated by arrows.The spelling **Utilities > I/O Test** indicates the command **I/O Test** in the **Utilities** menu.

The character formatting TEXT Indicates instructions, commands, and names that you need to enter as shown.

Buttons are marked with square brackets. [OK] indicates the OK button.

Images and Tables

Figures and tables shown as such are numbered consecutively. Individual details in figures are marked by item numbers and lines. Each item number is explained in the legend accompanying the figure.

4 INTRODUCTION

4.1 Getting Started

4.1.1 Using the software

vcwin is an external test program editor for Vision & Control GmbH supplied vision systems (intelligent cameras and multi-camera systems).

You use vcwin to create, change, test and start test programs. cutable. vcwin runs under Microsoft Windows.

Previous Experience

It is assumed that you are familiar with other applications which make use of the Microsoft Windows environment. Programming skills are not necessary. We recommend introductory training at the Vision Academy. Before you start writing your own programs you should be familiar with the way vcwin works. You should be able to follow the example programs and have a good idea of the range of functions available within vcwin.

Available Functions

vcwin inquires which functions the vision system supports and activates/deactivates menu items and options accordingly during connection.

vcwin only displays the functions, commands and command options that are supported by your vision system. Under *"Function and Command Reference (Firmware Overview) ", page 416* you will find a short list. Please contact us for more information.

Optional Commands

Various commands are optional and are only available for certain vision systems, hardware and firmware versions. Information concerning this is available from your supplier.

Command codes

ADVICE
All the command codes are described in the vicorem reference manual. However, it is possible that your vision system cannot interpret all the commands. Therefore, please always refer to the hardware manual for your vision system.

Also add short, self-defined commands to the test program using **Command Selection > Control > Direct Code Input** as HEX code. Please ensure that your vision system – e.g. via firmware upgrade – can interpret these commands.

Available Languages

The program interface can be switched between English and German.

Scope of Delivery

- Installation DVD
- Programmer's manual vcwin
- When buying a vision system from Vision & Control GmbH respective manuals are provided on DVD

4.1.2 Communicating with Vision Systems

Operating vision systems with a PC

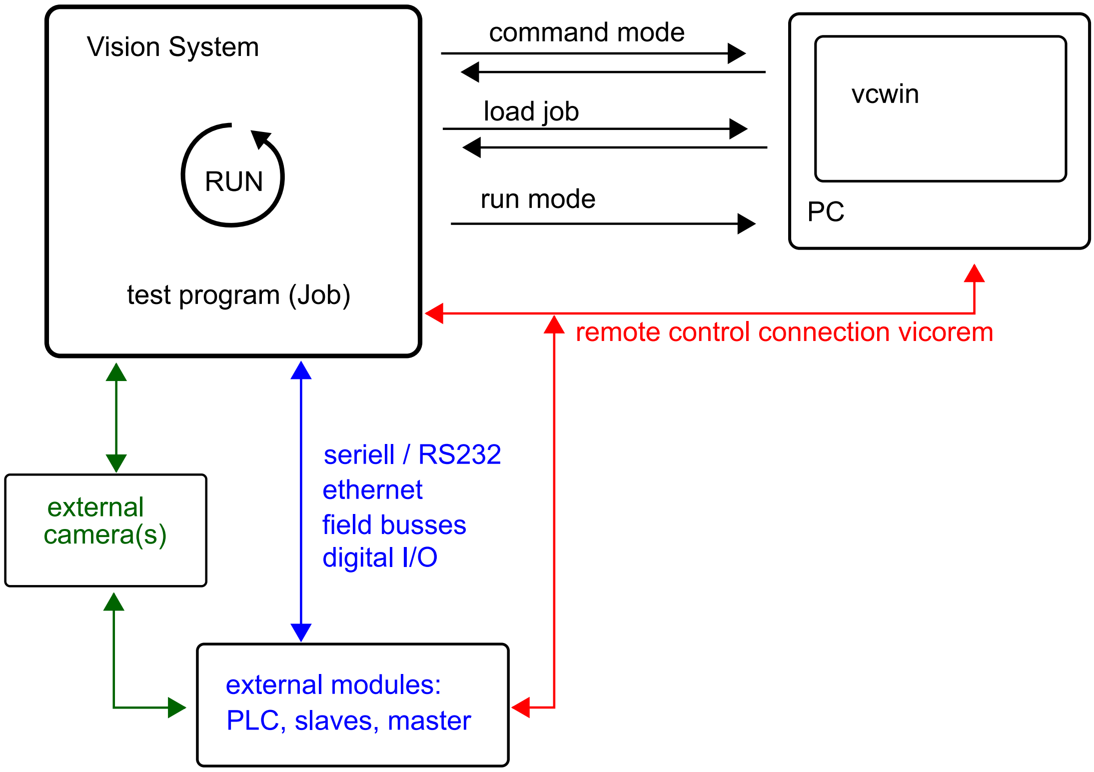


Image 1: Block diagram of a vision system with possible types of editor and link to internal processes

Vision systems communicate with a processing environment in one of these ways:

- Serial / RS232
- Ethernet
- Digital I/Os
- Several fieldbuses

Communication via vicorem

vicorem is a block-oriented protocol used for communication between the vision system and the master system (operating software vcwin, PLC, ...).

The operating software vcwin sets all programs in this code.

Format:

- 2 byte block length
- 1 byte command recognition (ID)
- n byte command parameter
- 1 byte checksum

Example:

vcwin command:

```
Determine Point 0 (Image); Max. Gradient [10, 1, 2];
Dark/Light Search Arrow: [249, 286/109, 259]
```

vicorem-remote code (hexadecimal):

```
[ 10 00 00 00 01 00 F9 01 1E 00 6D 01 03 02 07 0A 02 01 00 00 ]
```

ADVICE

More information about vicorem and its use can be found in the vicorem manual. Please contact our support.

Communication via PC

The operating software or another compatible external editor is installed on the PC.

ADVICE

External editors are programs that run on a host computer and use remote communication to link to the vision system.

Communication is characterised as follows:

- Programs can be read into the vision system and tested via the remote interface from the PC.
- The results are sent back from the vision system to the PC.
- Tried and tested programs are transferred from the PC to the vision system and will run there automatically.

Communication between the operating software and Vision Systems

The operating software use the *command mode* to call any commands, to test and modify them and then to create the program text. The test program is either loaded into the vision system's RAM or directly as a file into the flash (*load job*).

In *run mode* (RUN) the program is executed on the vision system. The operating software can log single functions (e.g. send image for the user interface).

Command Mode

In the command mode, the vision system works as a client of the PC. The communication is characterised as follows:

- The PC sends single commands to the vision system.
- The vision system executes each command.
- The vision system returns a result (success, failure or test value) to the PC.
- Results are processed and evaluated by the PC.

Job Mode (Load Job)

- The PC loads a complete test program (job) into the processing system using the remote control interface vicorem.
- The program is converted into a for the vision system understandable language (e.g. ASCII).
- The processing system stores the program on the flash or RAM.

Run Mode

- The vision system transfers the program from the flash to the RAM and executes the program cyclically.
- An interaction with the PC will take place only by means of special commands.
- Special commands or functions send the results to the PC. e.g. Send Image, Send Measuring Values, Online Debugging

Communicating with PLC Devices

The vision system can also be connected via the signal I/O port or the RS-232 interface, ethernet or fieldbus interface to Programmable Logic Control devices (PLC). The PLCs send or request event data and can thus be integrated into the test runs. Direct control of actuators is also possible.

External modules can also control the processing system by remote control interface vicorem.

4.1.3 Installing and starting

Minimum requirements of the computer

- Operating systems: Windows 7, Windows 8, Windows 8.1, Windows 10
  - Both 32 and 64 bit versions (Windows RT is not supported)
- DVD drive (for installation from DVD) or Internet connection (for installation after download)
- Minimum 1 GB free RAM
- Monitor with a resolution of at least 800 x 600 pixels
- Ethernet interface
- Serial interface (optional for sending and receiving via RS232)

Installing Vision System

ADVICE

Please refer to the instructions of use of the vision system for the exact connection of the vision system.

Connect the vision system to the host computer with one of the following methods:

- via a serial cable to a free serial interface
- via a ethernet cable with a free ethernet port

Optional:

If the vision system has a monitor interface, connect the video screen to the vision system with a video cable.

Connecting cables, e.g. PLC, can be connected to the parallel I/Os (In- and Outputs) of the vision system.

Installing vcwin

Execute **vcwin setup** from the installation medium and follow the instructions. The vcwin software will create an own icon on the Windows desktop.

Start and Quit vcwin

The vcwin software is started by either double-clicking on the vcwin icon or from the Windows start menu by calling **Start > Programs > Vision & Control GmbH > vcwin> Vision & Control vcwin.**

Terminate vcwin with ALT+F4 or with the menu item **File > Exit**.

Designating the Port in vcwin

After having started vcwin, you should designate the port to which the vision system is connected by calling **Communication > Interface** from the menu. For more information, see the section *see "Interface", page 75*.

Connecting vcwin with a Vision System

Before starting program teach-in, you must connect the vision system by using the command **Communication > Connect**. The vcwin software initialises the connection to the vision system and automatically activates all the commands that the vision systems supports.

4.1.4 From Single Commands to Test Programs

Summary of Commands

Test programs consist of a list of commands. The commands are enumerated chronologically and contain a single executable instruction each. Comments can be inserted to help make the program more readable.

Test Program Size

With vicosys systems, the maximum program size is limited only by the Flash memory. pictors allow a maximum program size of 512kB.

ADVICE

It must be considered here that, due to different coding, the size of a vcwin program differs from the actual size of the test program on the vision system. You can see the actual size of the test program under **System Preferences > System Resources**.

Because the individual test commands can have different vicorem lengths, the maximum program size cannot be given exactly as the number of test program lines. As a rough guide, a maximum program size of 512kb corresponds to around 7000 lines.

Number of Test Programs

The number of test programs is limited only by the total size of the flash memory.

Image Acquisition Commands

Configure the image acquisition commands for controlling the acquisition of visual data by the vision system or the external cameras of the vicosys. This group of commands contains:

- Image acquisition
- Configuring shutter mode
- Synchronous flash
- Selecting video mode, e. g. live image, memory image, overlay display and image acquisition
- Deleting image information
- Setting acquisition page, work page, demo page, display page
- Displaying configuration of image-memory pages
- Image preprocessing
- Image difference
- Color binarization
- Color conversion
- Utilities for vicosys
- Line scan camera

Locate Commands (Investigation and Detection)

Configure test procedures for the analysis of test object properties with the detection commands. The results are interpreted as quality criteria and the geometry variables determined are saved. This group of commands contains:

- Testing geometry (points, lines, circles, contours, blobs, edges)
- Testing gray value and brightness
- Recognizing patterns, Data-Matrix code and characters

- Counting pixels
- Compensating brightness
- Checking focus
- Teaching-in, searching and identifying objects
- etc.

Evaluation Commands

With evaluation commands, you combine already determined geometries and results, and configure the corresponding outputs in the test program. This group of commands contains:

- Calculating geometries from existing geometry variables (points, lines, circles, contours, distances, angles, angular positions)
- Evaluating and issuing results
- Superimpose parameters
- Counting
- etc.

Control Commands

With control commands, you coordinate how the test program interacts with other equipment, such as a PLC and handling systems. This group of commands contains:

- Checking and branching the program
- Controlling the inputs and outputs of the camera
- Communicating with robots and transforming coordinates
- Calibrating
- Sending measured values and results via several interfaces
- etc.

Obsolete Commands

Obsolete commands are commands that have been superseded by newer commands. Although these commands are no longer supported, they are found in some versions yet, due to backward compatibility of vcwin.

ADVICE

The commmands in the category **Obsolete** should no longer be used for test programs, see also *see "Obsolete commands", page 388*.

Checklist for Planning Tests

Test programs are similar to any other programs in that they consist of a list of commands. The commands are enumerated chronologically and contain a single executable instruction each. Comments (only with contents) can be inserted to help make the program more readable.

Establishing Procedures during a Test

- Decide which contours and points should be included to determine geometry.
- Establish the best order.
- Define the screen output: verbal messages (e.g. errors, status etc.), values, superimpose with geometric figures.
- Define the requirements for the communication (PLC, Ethernet and serial): start signal, standby message, error messages (general, special), output of values.

Assigning Variables according to the Test Plan/Drawing

- Adherence to systematology (e.g. points to circle 3: P31, P32, P33).
- Establish areas in which variables only temporary required will be stored so that they can be overwritten.
- Use the **Comment** command for the program documentation.

Structuring Test Programs Efficiently

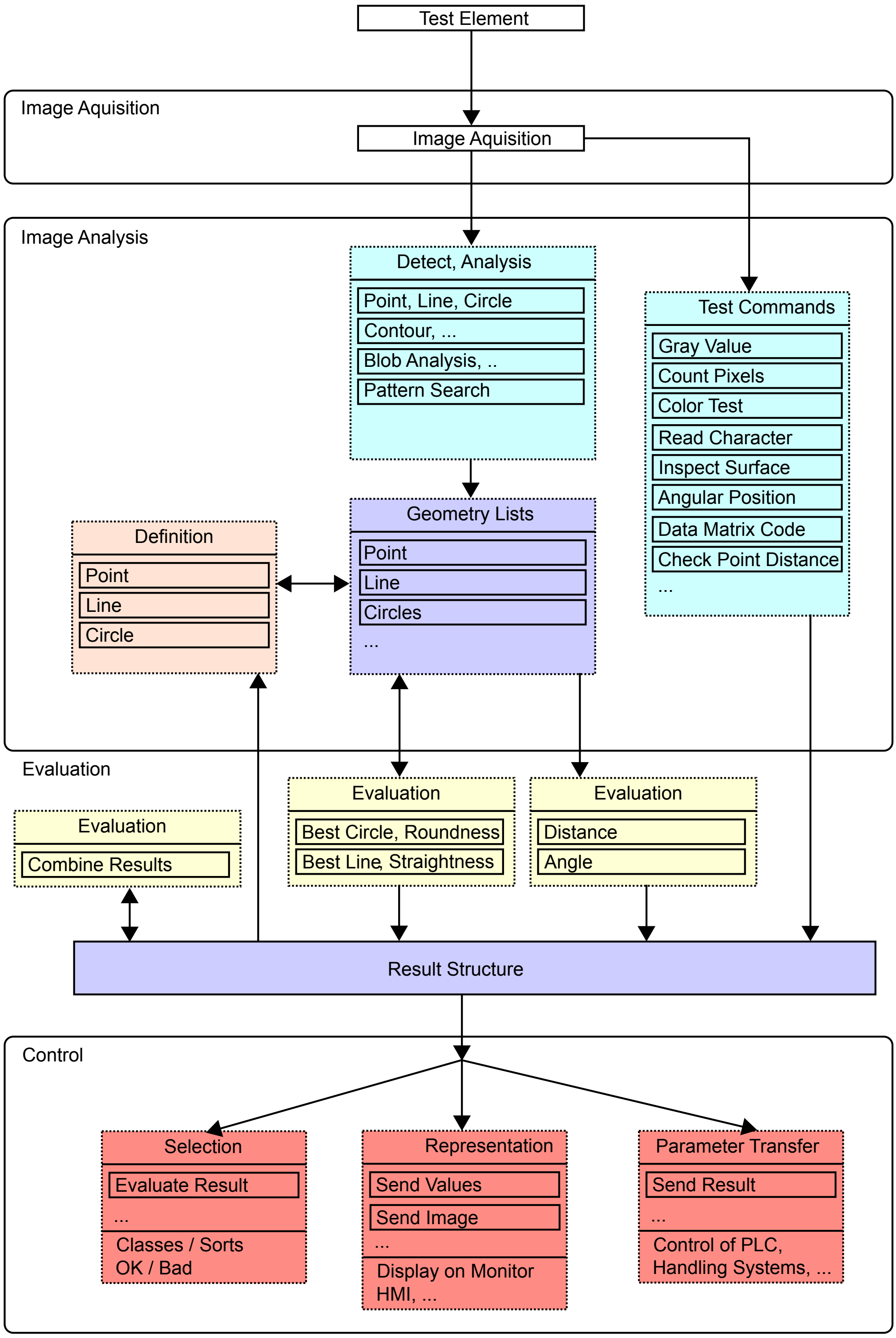
- Find repetitions in the program and rewrite them as loops with break conditions.
- Find often-used routines in the main body of the program and rewrite them as subroutines.

Use the Support Available

- Please take note of the demonstration programs included in the appendix. These programs can be used as frameworks for adapting to individual tasks.
- Please take advantage of the up-to-date know-how available at training courses run at the Vision Academy.

Combining Commands and Command Groups

Find important command groups and processes for program generation from the following diagram.



Running Test Programs

Step 1: Image Acquisition

The image captured by the vision system is stored in an image page in the image memory.

Step 2: Image Analysis

Use the following options to describe the test objects:

- geometric variables and the geometric relations between them
- qualitative attributes of the test objects

The determination of geometric variables and the check of quality properties are carried out by the detection command. New geometric variables and further numeric measurement results can be calculated from the immediate geometric results.

Step 3: Evaluation

Evaluation of results can take on several forms:

- comparing results in dialog box with nominal values
- using the Evaluate Result command
- using the Combine Results command

Step 4: Control Commands

Control commands and the commands for calibration and justification guarantee that the program can be properly adjusted to the existing hardware environment, e. g. the parameterizing of the communication to industrial robots.

Moreover, several communication options are available:

- sending values over serial/RS232
- sending values over ethernet
- sending signal states over the digital I/Os
- Communication via several field busses

Variables and Elements in Programs

You can see the type and number of commands that are supported by your vision system displayed in a window when connecting the vision system.

Reference Geometry

Parameter sets describing the geometry of test objects can be saved so that the data can be re-used later on. The parameter sets can contain points, lines, circles, contours and results.

Result Numbers / Result Structure

A sequence of commands leads to arithmetic results or test results which can be stored under result numbers and are available for further processing. All the results together form a result structure.

4.1.5 Checklist for Operating the Software

Preparation for Editing Test Programs

Execute the following steps prior to editing programs:

1. Install the vision system. Follow the instructions for use for the corresponding vision system. Install the user software vcwin.
   >> *see "Installing and starting", page 15*
2. Start vcwin.
3. Register the interface between the host computer and the vision system in vcwin by calling **Communication > Interface**.
   >> *see "Interface", page 75*
4. Connect the vision system by **Communication > Connect**.
   This causes the vcwin menu to contain only commands which are supported by the connected vision system. The vision system will be in Command Mode.
   >> *see "Operating vision systems with a PC", page 13*
   The vision system can now be remote-controlled from the host computer.

5. Select **Live Image** in the **Video Control Panel** *see "Docking Window Video Control Panel", page 121*, then position the test object and adjust the sharpness of the vision system image on the control monitor or alternatively in the monitor window *see "Docking Window Monitor Window", page 118*. Also ensure that the aperture is correctly adjusted.
6. Now select the **Memory Image** option in the **Video Control Panel**.
7. Click on the **Grab Image** button to load the current image into the image memory page.

Inserting and Editing Commands

1. Move the cursor to the position in the command list following which you want the new command to be inserted.
   >> see "Command Display", page 116
2. To insert a new command: Select the required menu entry in the command selection.
   To edit an existing command: Double click on the program line you want to change.
3. In the case of Locate commands you should define the detection windows or beams directly on the control monitor or monitor window.
   see "Teaching in Test, Detection and Search windows", page 26
4. Enter the input and output parameters of the result structure or geometry structure as well as measured values and tolerances or select these from the displayed resources.
5. Test the command with the [Test] button. If this returns the desired result e.g. it finds an edge in the video image, go on to item 6. If not, adjust the parameters or the detection/ test window.
6. Press the [OK] button. The command will be added to the command list as text in vcwin.

You can use commands also in the following manner:

- copying
- cutting
- deleting
- saving in a special file (as a block)
- commenting out

Testing the Program and Transferring it to the Vision System

1. Test in Command Mode using the commands in the **Utilities** menu.
   >> see "The Utilities Menu", page 68
2. If this test is successful, copy the test program into the main memory of the vision system by calling **Communication > Transfer**.
   >> see "Transfer", page 73
3. In run mode, test by calling **Communication > Run**.
   >>see "Disconnect", page 73
4. If there are no errors, connect the vision system by calling **Communication > Connect**. The vision system returns to the Command Mode.
5. Transfer the test program into the vision system by calling **Communication > Transfer** or with the [**Transfer**] button. The test program is stored in the vision system's own file system.

Special Features

Vision Systems with Flash give you the opportunity to transfer programs via **System Preferences > Save File System in Flash**.

4.2 Customize interface

You can modify and customize the presentation of the programming window. In order to do so, use the docking windows and toolbars.

4.2.1 Customize Docking Windows

The windows within the user interface can be freely positioned, grouped with other windows in indexes, and be shown or hidden as requested by the user. These are special docking windows which can be docked on other elements in the interface (e.g. other windows or the borders of the programming window). When docked, they are anchored at this position but can be relocated to any other position at any moment.

This works with the help of control elements which are displayed when you start to relocate a docking window (*see "Image 2: Control elements"*). When you place the window above a control element, a transparent blue background shows you where the window will be positioned. You then just have to release the mouse to finally relocate the window to the desired position.

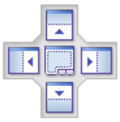


Image 2: Control elements

To group several windows into a tab window, you have to place the windows directly on the central control element of the other window so that the target window gets the transparent blue background before you drop the current window. The respective tab indexes will be generated automatically, with which you can display the grouped windows. There can only be one active window at a time.

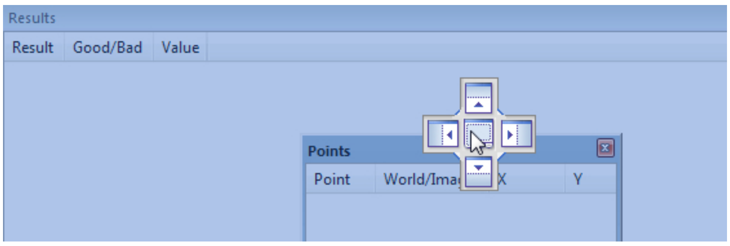


Image 3: Group two windows

You can also place docking windows freely floating above the actual user interface. Therefore you simply have to move the respective window over a free space on the interface and not over a control element. The window will then be detached from the programming window and placed accordingly. To show or hide windows, you can use the options under View > Toolbars and Docking Windows. Here you can choose which windows and toolbars are to be displayed and which sould be hidden. Additionally, you can hide a window temporarily by moving it automatically into the background. To do this, click on the little pin icon in the top right of the window's title. The window will then be moved in to the Customize interface 2 background and be replaced with a tab index at the border of the programming window. When you move the mouse cursor over the tab index, the window will be shown as long as you don't move the mouse out of the window. In order to display the window permanently again, you have to click on the pin icon once again. This function is particulary useful for netbooks or laptops with low resolutions.

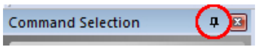


Image 4: Hide a window temporarily

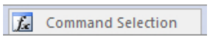
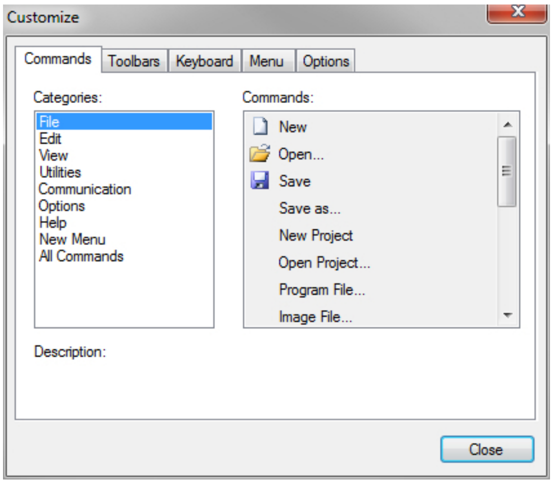


Image 5: Tab index of a temporarily hidden window

4.2.2 Customize Toolbars

With the menu **View > Toolbars > Custzomize**, the user can make further adjustments to the user interface. More information is shown below.



This dialog is divided into five tab indexes. The first two, **Commands** and **Toolbars**, allow the user to customize the toolbars and menus, the index Keyboard enables the assignment of hotkeys to single commands, with Menu you can reset the user interface to defalut, and Options offers settings for the display of quick info texts.

Customize Toolbars and Menus

While the Customize menu is open, you can change the postion, arrangement or alignment of almost any interface item (including icons, lists, menu points and entries) via drag and drop. Therefore the index Commands lists all commands and options that are executable within the operating software interface in separate categories. For example, if you want to move a command that is by default only available in a menu into a toolbar, you simply pick it from the list and put it into the desired toolbar with drag and drop. You can also pick the command directly from the interface.

In the tab index **Toolbars**, you can delete, rename, reset, or create toolbars. When you create a new one, you can fill it with commands from the **Command** tab.

Create toolbar

You create a new toolbar as follows

1. Go to the **Toolbars** tab and click on the **New** button.
2. Type in a name for the new toolbar and confirm it with **OK**.

   ⇨ An empty toolbar gets created on the middle of the screen.
3. Go to the Commands tab, choose a category and pick a command.
4. Click on the command and move it with the button pressed to the new toolbar.

5. Alternatively you can add commands directly from the interface with drag and drop. Choose a menu entry or icon and move it into the new toolbar.
6. Repeat steps 3 and 4 resp. 5 to add more commands. When finished, you can dock the new toolbar to the others by moving it next to them.

Edit toolbar

You edit a toolbar as follows

- **Add commands**: See above, steps 3, 4 and 5.
- **Delete commands**: Right-click on the item and select the context menu entry **Delete**. Alternatively you can move the item into a disabled area in the interface (marked by the x at the bottom of the mouse cursor) via drag and drop.
- **Change an items appearance**: Right-click on the item and select the context menu entry **Button Appearance**. Now you can modify the icon graphic, edit the displayed text and adjust further options.
- **Reset toolbars**: Go to the **Toolbars** tab in the Customize menu, mark the toolbar in the list and click on the **Reset** button. To reset all toolbars, click on the **Reset All** button.
- **Rename toolbars**: Go to the **Toolbars** tab in the Customize menu, mark the toolbar in the list and click on the **Rename** button.
- **Delete toolbars**: Go to the **Toolbars** tab in the Customize menu, mark the toolbar in the list and click on the **Delete** button.

Customize hotkeys

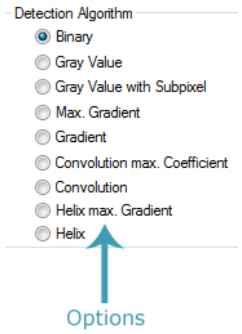
1. Go to the **Keyboard** tab.
2. Select a command category.
3. Mark the command that you want to assign a hotkey.
   ⇨ The current hotkey is shown under "Current Keys".
4. Type in your own hotkey into the "Press New Shortcut Key" text area. If the new hotkey is already token, type in a different one.
5. Click on the **Assign** button.

4.3 Introduction to Commands

4.3.1 Dialog Elements

In this section, you can find information regarding important elements that are needed in order to parameterize commands.

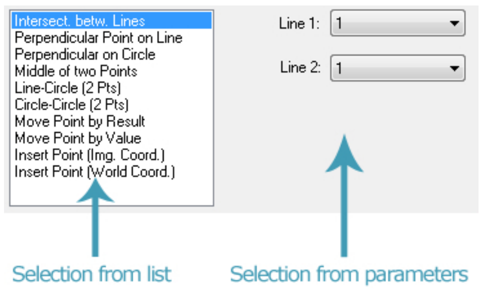
Options Dialog for Detection Algorithm



Use this dialog type for selecting the detection algorithm. Proceed as follows:

1. Select the desired option.
2. Configure the parameters using the arrows at the right-hand border or enter the values directly.

Options Dialog for Calculating Geometric Variables



Use this dialog type for calculating geometric variables. Proceed as follows:

1. Select an entry in the left-hand list either by mouse or the cursor keys.
2. Highlight one of the options displayed in the center part.
3. Configure the parameters displayed at the right-hand by using the arrows at the right-hand border or enter the values directly.

Selection Dialogs for Variables, Strings and Results

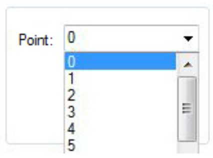
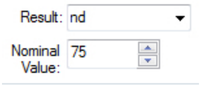


Image 6: Dialogs for Element Numbers and Geometry Elements (Examples)

vcwin uses drop-down menus (left-hand image) and element lists (right-hand image) for selecting geometrical elements, strings and results. The numbers of previously defined geometrical variables and results can be selected or entered directly by keyboard. Use this type of dialogs for loading, saving and transferring elements.

To deactivate the selected element choose or type {-1} respectively {nd}. This means that this element will not be loaded, saved or transferred on execution of the command. The elements available are displayed as black characters in the listings, elements that are not available are displayed in gray.

Dialog Box for Determining Results



Use this type of dialogs for specifying results and the corresponding tolerance tests.

What is the Result Structure?
The Result Structure is the totality of the result variables used. The result variables, for example, save whether the result is good or bad. Simultaneously, they always save the first value determined.

Input Fields
The individual fields should be considered in combination and have the following meanings:

- **Result (RES)** determines the number under which the result is saved in the result structure. The result is GOOD if the actual value assumes values within the tolerance range and BAD otherwise.
- **Nominal** and **Tolerances** determine the nominal value and the tolerance range for the parameter to be checked.

4.3.2 Teaching in Test, Detection and Search windows

Basics

Test-, Detection- and Search windows are used to:

- Test window: ... to specify the subregions of the image to be checked
- Search window: ... to define search areas in the image
- Detection window: ... to enclose objects to be detect

Geometries

Depending on the command, the test and search windows can assume different geometries.

You can select a default geometry or use the geometry of a test window.

Detection windows each use only a specific geometry, e.g. geometry detection beam at the command *Locate Point*.

Default geometries

- Standard geometries are in their shape fixed defined geometries (see also: *"Geometries", page 30*). They are selected and configured directly in the command.

Test windows

- Test windows are variables that describe geometries. Test windows are generated by a command in the test program.
  The test window geometry is taken from the variables of the test window at the command execution time. If a position tracking has been set, it will be applied at the time the command is executed.

Teach-in Tab

The test or detection windows are located in the *Teach-in* tab. For commands where a reference object has to be found (e.g. Pattern Search), there is a *Teach-in Pattern* and a *Search Window* tab instead of the *Teach-in* tab.

Furthermore, there is "Teach-in Mask" (Mask Check) and "Teach-in Distortion" (Image Equalization). The difference to the tab *Teach-in Pattern* lies in the additional indication of the grey value threshold.

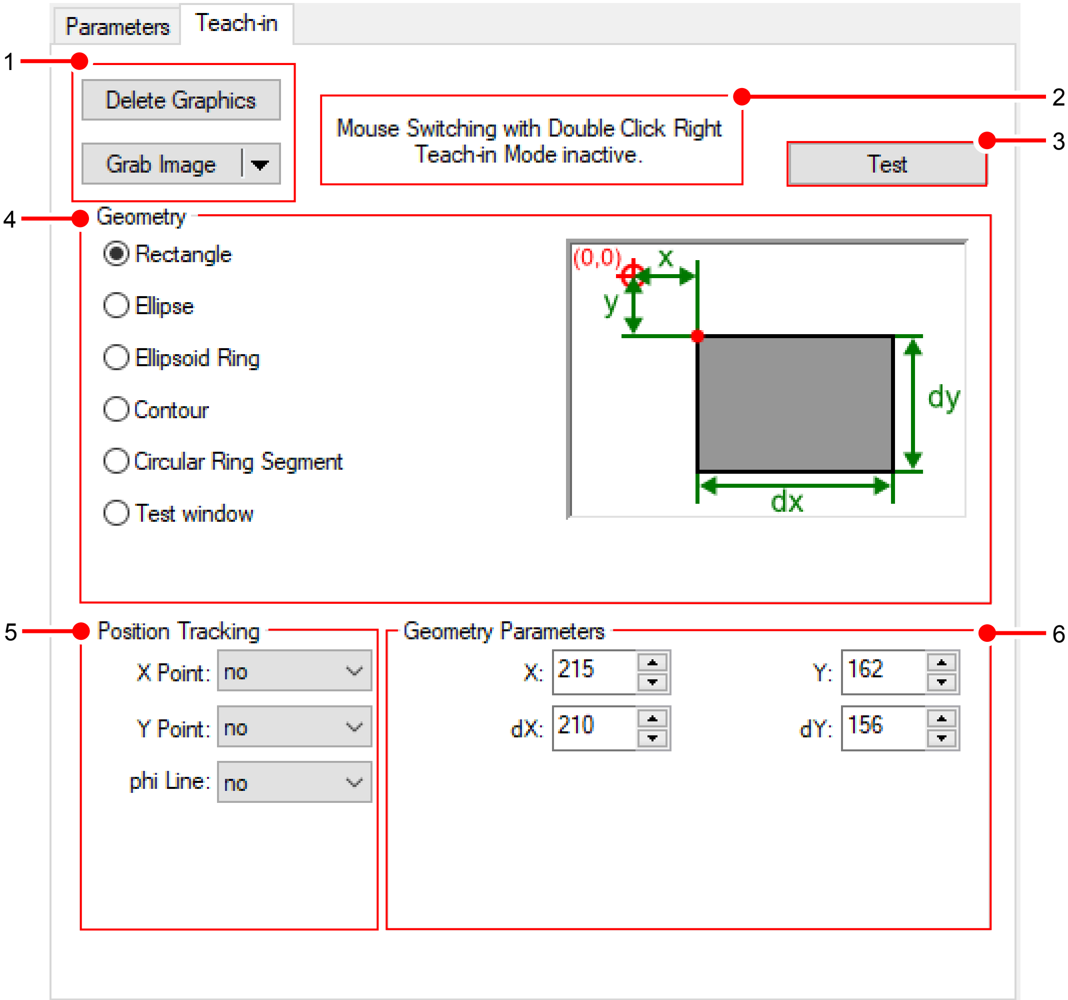


Image 7: Teach-in tab

The dialogue offers the following principal possibilities:

1. Deleting all graphics in the current image using [Delete Graphics]. As well as taking a picture with [Grab Image]. If the vision system consists of several individual cameras, select the desired camera using the arrow in the drop-down list.
2. Switch to the teach-in mode to parametrize the test, detection or search window in the video image (recommended).
3. Execution of the overall test, identical to the [Test] button in the *Parameters* tab. Selecting the geometry of the test, probing or search window. Indication of the position tracking.
4. Selecting of the geometry of the test, detection or search window. Indication of the position tracking.
5. Specify the Position Tracking.
6. Parametrization of the test, detection or search window for teaching in the window dialogue by numerical input of the coordinates or selection of the contour resp. test window variable.

Only options that are usable in the command are displayed. Unusable parameters are either greyed out or not displayed.

Teach-in via Video Screen

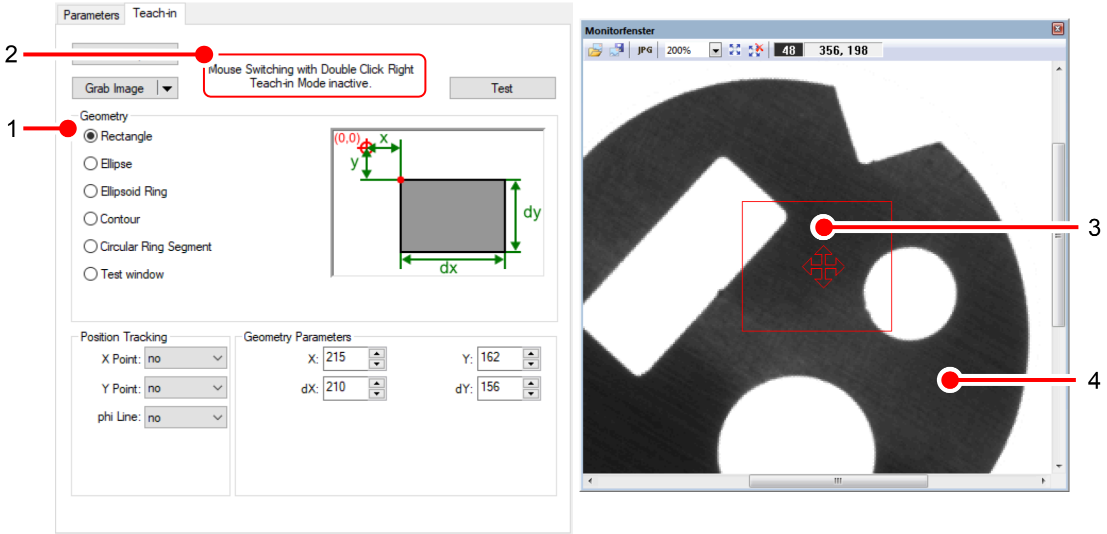


Image 8: Teach-in test and detection window in the Monitor Window

1. Select the geometry. Some functions have fixed geometries.
2. Switch to learning mode: Double-click with the right mouse button in the dialogue to switch from the edit window with the mouse pointer into the video image (Monitor Window or the external monitor).
3. In the video image, move the mouse pointer over the outer contour and corners of the detection window or into the window until the mouse pointer changes into a small double-headed or quad-headed arrow (handle). For some functions, a rotation arrow appears. Modify the dimension and position of the window using the mouse. In order to do this, move the mouse whilst holding down the left mouse button:
   - at the *corners* ... to resize the window in diagonal direction
   - at the *edges* … to move the side borders perpendicular to the window limits
   - *centrally* … to move the window
   - on an *arrowhead* ... to turn the window

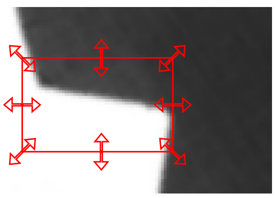


Image 9: handle corner/ edge

Image 10: handle centre

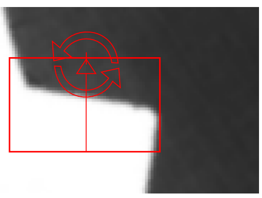
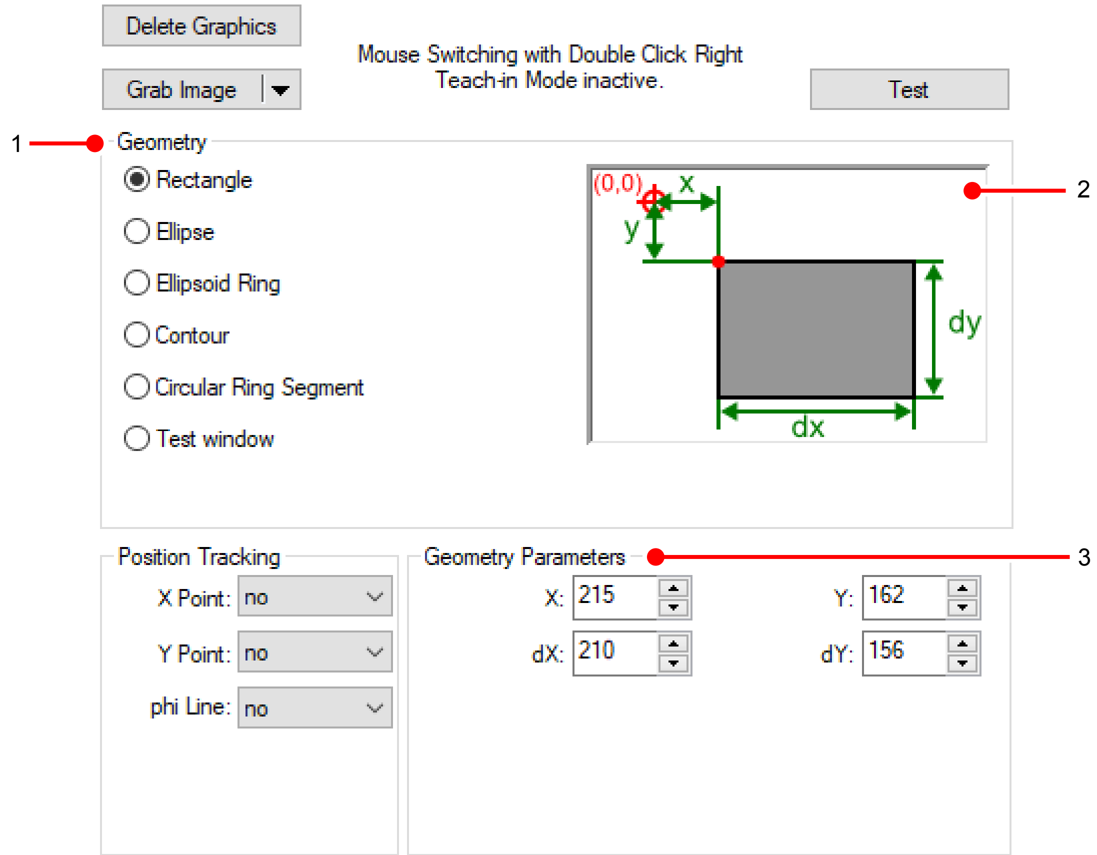


Image 11: handle arrowhead

4. Some windows have detection beams e.g. Locate Circle. With a double- click on the arrowhead, the detection direction can be changed.
5. Double-click on the right mouse button again to return to the edit window.

   ⇨ The changes made in the video image are adopted by the dialogue window (e.g. new coordinates).

Teach-in via tab



1. Select the geometry. Some functions have fixed (special) geometries.
2. Depending on the selected geometry different values are required (see also: *"Geometries", page 30*). The window displays these geometry parameters. All numerical information refers to image coordinates and are counted from the upper left corner of the screen or from the reference point.
3. Enter the values for the geometry parameters.
   - Change the values as needed using the arrows on the right-hand side of the input fields or enter them directly.
   - For contour or test window, select the appropriate variable from the drop-down list.

Teach-in Pattern / Mask / Distortion

The "Teach-in pattern", "Teach-in mask" and "Teach-in Distortion" tabs have a similar structure to the "Teach-in" tab. They are used with the following commands:

- Image > Image Equalization
- Locate > 360° Pattern Search
- Locate > Edge Based Object Search
- Locate > Search Pattern
- Locate > Mask Check
- Obsolete > Advanced Pattern Search

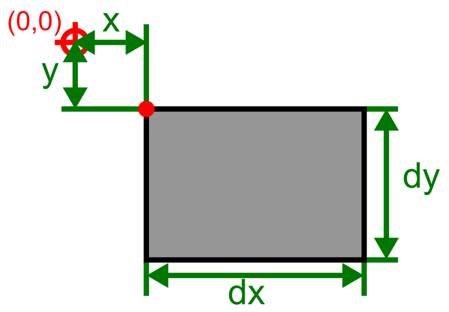
With the button [Read Pattern] the image content, which is located in the geometry defined here, is saved.

With the button [Read Mask] the image content, which is located in the geometry defined here, is binarised according to the indicated grey value threshold and then saved.

With the button [Teach Distortion] the image content, which is in the geometry defined here, is binarised according to the specified threshold value. The points are searched and the dot matrix is saved.

4.3.3 Geometries

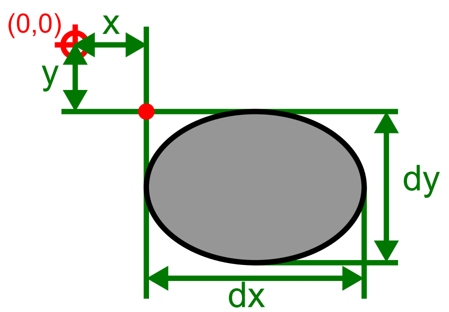
Standard Geometry:
Rectangle



Geometry Parameters:

- X: X coordinate of the upper left corner
- Y: Y coordinate of the upper left corner
- dX: Width of the rectangle
- dY: Height of the rectangle

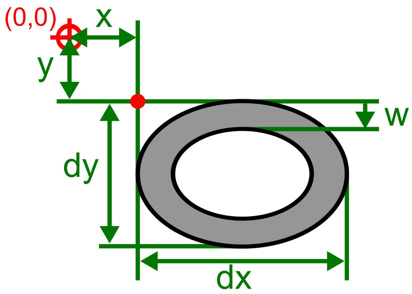
Standard Geometry:
Ellipse



Geometry parameters:

- X: X coordinate of the upper left corner of the bounding rectangle
- Y: Y coordinate of the upper left corner of the bounding rectangle
- dX: Width of the ellipse
- dY: Height of the ellipse

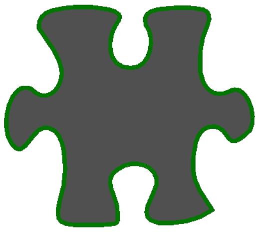
Standard Geometry:
Ellipsoid Ring



Geometry parameters:

- X: X coordinate of the upper left corner of the bounding rectangle
- Y: Y coordinate of the upper left corner of the bounding rectangle
- dX: Width of the ellipse
- dY: Height of the ellipse
- w: Width of the ellipsoid ring, indication from the outer edge

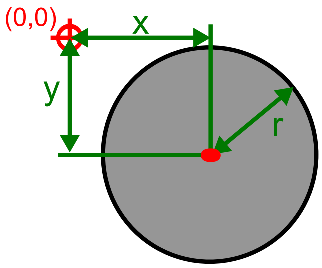
Standard Geometry:
Contour



Geometry parameters:

- Contour: Number or Name of the contour
  (must be present on the vision system, for example Locate > Contour)

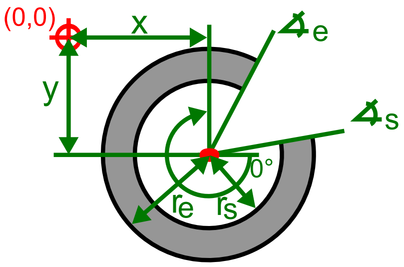
Standard Geometry:
Circle



Geometry parameters:

- Center Point X: X coordinate of the center point
- Center Point Y: Y coordinate of the center point
- Radius: Radius of the circle

Standard Geometry:
Circular Ring Segment



Geometry parameters:

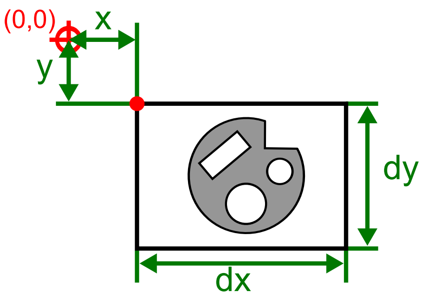
- Center Point X: X coordinate of the center point
- Center Point Y: Y coordinate of the center point
- Start Radius: Start of the circular ring segment
- End Radius: End of the circular ring segment
- Start Angle: Starting angle of the arc (sector) in degrees
- End Angle: Ending angle of the arc (sector) in degrees

only at Locate Edges on Circle

With this command, in addition, the search direction is given.

- Clockwise:
  The under Parameters selected edge transitions are searched only in a clockwise direction.
- Counterclockwise:
  The under Parameters selected edge transitions are searched only in a counterclockwise direction.

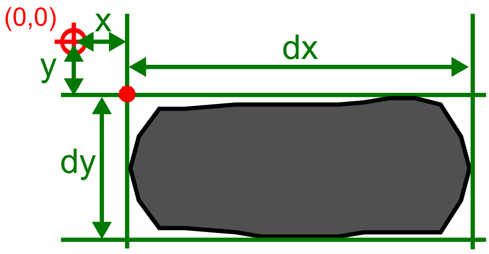
Teach-in Geometry:
Mask Check



Geometry parameters:

- X: X coordinate of the upper left corner
- Y: Y coordinate of the upper left corner
- dX: Width of the taught mask
- dY: Height of the taught mask

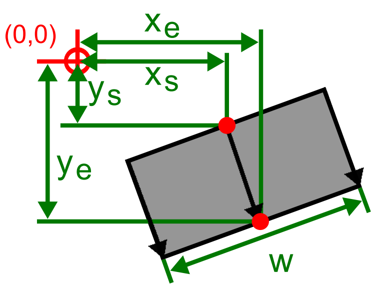
Teach-in Geometry:
Image Equalization



Geometry parameters:

- X: X coordinate of the upper left corner of the taught distortion
- Y: Y coordinate of the upper left corner of the taught distortion
- dX: Width of the taught distortion
- dY: Height of the taught distortion

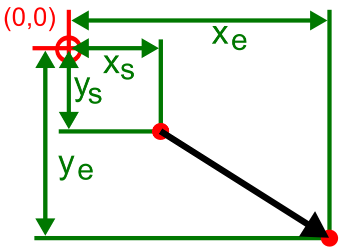
Detection Geometry :
Locate Line



Geometry parameters:

- Start X: X coordinate of the starting point of the detection arrow
- Start Y: Y coordinate of the starting point of the detection arrow
- End X: X coordinate of the ending point of the detection arrow
- End Y: Y coordinate of the ending point of the detection arrow
- Width: Width of the detection window
- Quantity of Detection Points: The quantity of detection points corresponds to the detection arrows. These are evenly distributed over the width. On each detection arrow a detection point is searched. From the detection points found then the line is formed. The more detection points are used, the more accurate the line forming is, but the longer the computation takes time.
- Determine whether all detection points must be found so that a line is formed from the single detections.

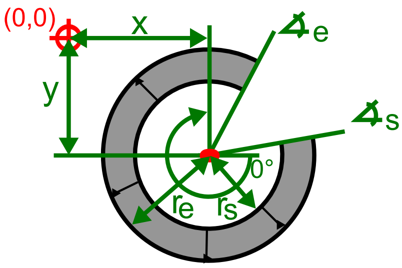
Detection Geometry :
Locate Point



Geometry parameters:

- Start X: X coordinate of the starting point of the detection arrow
- Start Y: Y coordinate of the starting point of the detection arrow
- End X: X coordinate of the ending point of the detection arrow
- End Y: Y coordinate of the ending point of the detection arrow

Detection Geometry :
Locate Circle

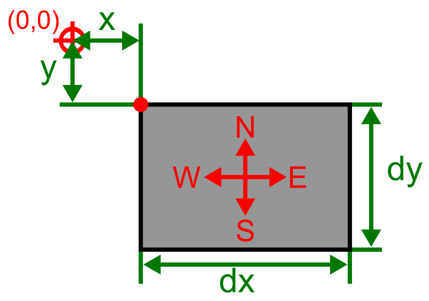


Geometry parameters:

- Center Point X: X coordinate of the center point
- Center Point Y: Y coordinate of the center point
- Start Radius: Start of the circular ring segment
- End Radius: End of the circular ring segment
- Start Angle: Starting angle of the arc (sector) in degrees
- End Angle: Ending angle of the arc (sector) in degrees
- Quantity of Detection Points: The quantity of detection points corresponds to the detection arrows. These are evenly distributed over the circular ring segment. On each detection arrow a detection point is searched. From the detection points found then the circle is formed. The more detection points are used, the more accurate the circle forming is, but the longer the computation takes time.
- Determine whether all detection points must be found so that a circle is formed from the single detections.

If the start radius is larger than the end radius the search direction changes from inside -> outside to outside -> inside.

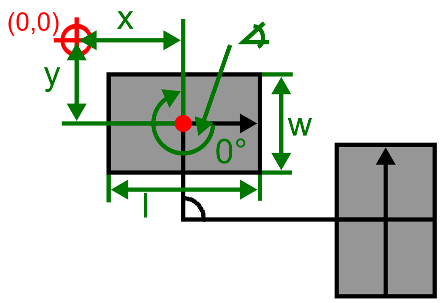
**Detection Geometry :
Locate Contour**



Geometry parameters:

- X: X coordinate of the upper left corner
- Y: Y coordinate of the upper left corner
- dX: Width of the rectangle
- dY: Height of the rectangle
- Search Direction:
    - N(orth): bottom-up
    - E(ast): left-to-right
    - S(outh): top-down
    - W(est): right-to-left

**Detection Geometry :
Locate Angle**



The geometry parameters are entered separately for both Detection Windows.

Geometry parameters:

- Center Point X: X coordinate of the center point of the detection window
- Center Point Y: Y coordinate of the center point of the detection window
- Length: Lenght of the detection window
- Width: Width of the detection window
- Angel: Rotation of the rectangle around the center point

4.3.4 Position Tracking of Objects

Principles of Position Tracking

Position tracking is used if the object to be measured – with the detection beams and/or window – can be moved or rotated in relation to the initial teach-in. A differentiation is made between:

- Position tracking in the X/Y directions
- Angular position tracking

Principle

During teach-in with vcwin , the relative coordinates of the test window are calculated from the fixed window coordinates and the compensated coordinates of the reference points and lines. These coordinates can be changed in the entry masks of the relevant dialog boxes.

General Procedure

1. Search for clearly defined points or a line in the video image at which the position of the detection beams and window should remain constant during the testing.
2. Insert commands for determining these points/this line into the test program.
3. When teaching the image processing command with position tracking, select the points/ line detected under point 2 in the dialog fields for the position tracking.

Commands Where Position Tracking is Available
The following commands' tech windows can be tracked:

Commands	X/Y Tracking	Angular Position Tracking
Blob Analysis	×	
Data-Matrix Code	×	
Angular Position	×	
Rotation Based on Moments	×	×2)
Superimpose	×	
Advanced Pattern Search	×	
Color Blob Analysis	×	
Test Color	×	
Focus	×	
Locate Line	×	×2)
Gray Value	×	
Test Brightness Percentage	×	
Brightness Offset	×	
Locate Edges on Circle	×	
Transition at Line	×	
Locate Contour	×	
Locate Circle	×	
Mask Check	×	×2)
Save Pattern	×	
Search Pattern	×	
Inspect Surface	×	×2)
Search and Identify Object	×	
Count Pixels	×	×2)
Locate Point	×	1)
Read Character	×	

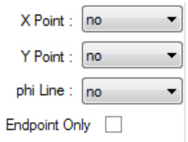
Instructions for angular position tracking
1) When detecting with gradient and convolution algorithms, the detection beams can only be rotated in steps of 45°.
2) In the case of test windows, only the central points are rotated. The window alignment in the X-Y directions is maintained.

Input Dialog

ADVICE

If the reference points/lines used for tracking are to be expressed in pixels (image coordinates), they have to be defined for the same image page and the same camera being used, otherwise you will get a type error.

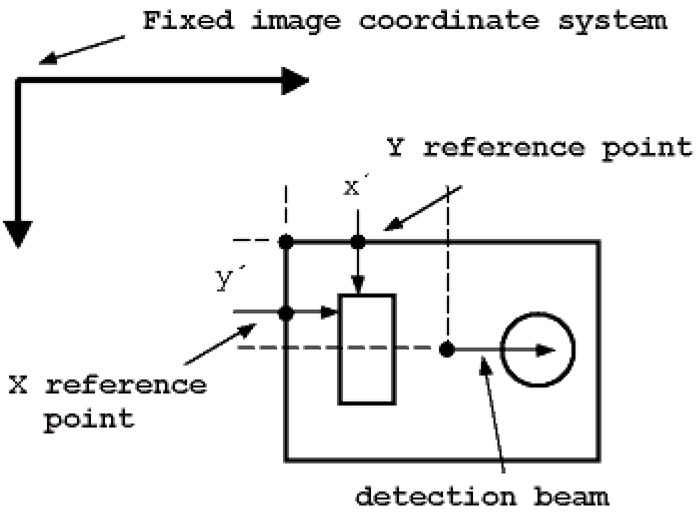
Use the following parameters for position tracking of test windows and detection beams:



Parameters	Description
X Point	The object will follow the X coordinate of the reference point.
Y Point	The object will follow the Y coordinate of the reference point.
X and Y Point	The object will follow both X and Y coordinates of a single reference point or two different reference points.
phi Line	The object to be followed is followed at a fixed angle to the given line.
Endpoint Only	Position tracking uses only the apex of the detection beam as a reference point. The base always remains at the same image position. In this way the position tracking of a point is possible. Only available for Locate Point. If you do not want any tracking you should enter either {1} or {-1} here. Points/lines used for tracking must already be defined in the point structure. They will then appear as resources in the selection window for **X Point**, **Y Point** and **phi Line**.

Position Tracking in X/Y Directions

Basics



Position tracking in x/y direction

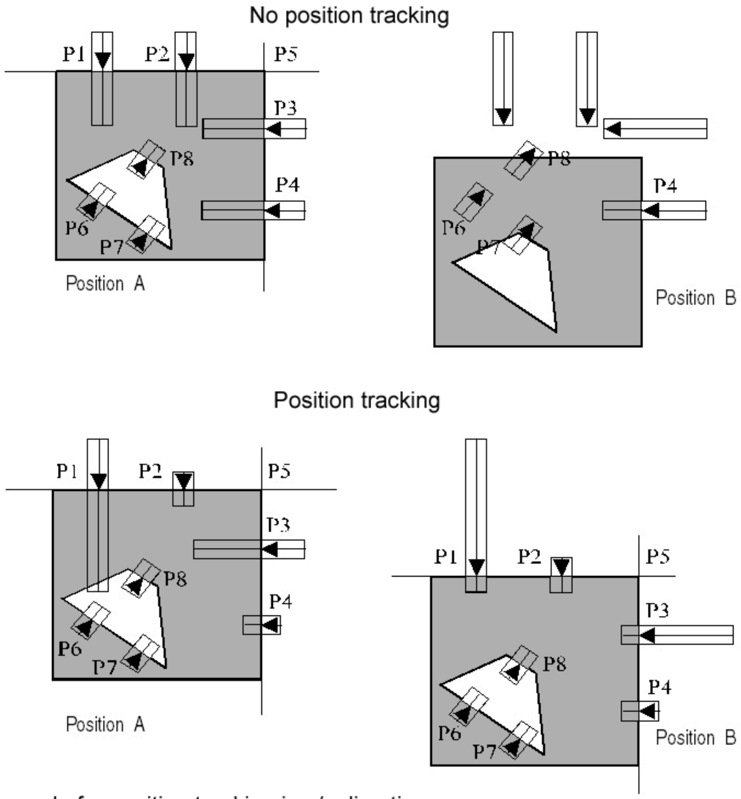
With position tracking in X/Y direction, the detection beams and/or the central points of the test windows are fixedly assigned to a new (moveable) coordinate system.

The orign of the new coordinate system consists of the X coordinate of the X reference point and the Y coordinate of the Y reference point.

During the position tracking, the relative position (X´, Y´) of the test windows and detection beams to the reference points always remains constant.

If position tracking is not enabled, the coordinate of the reference point in the corresponding direction is zero.

Example

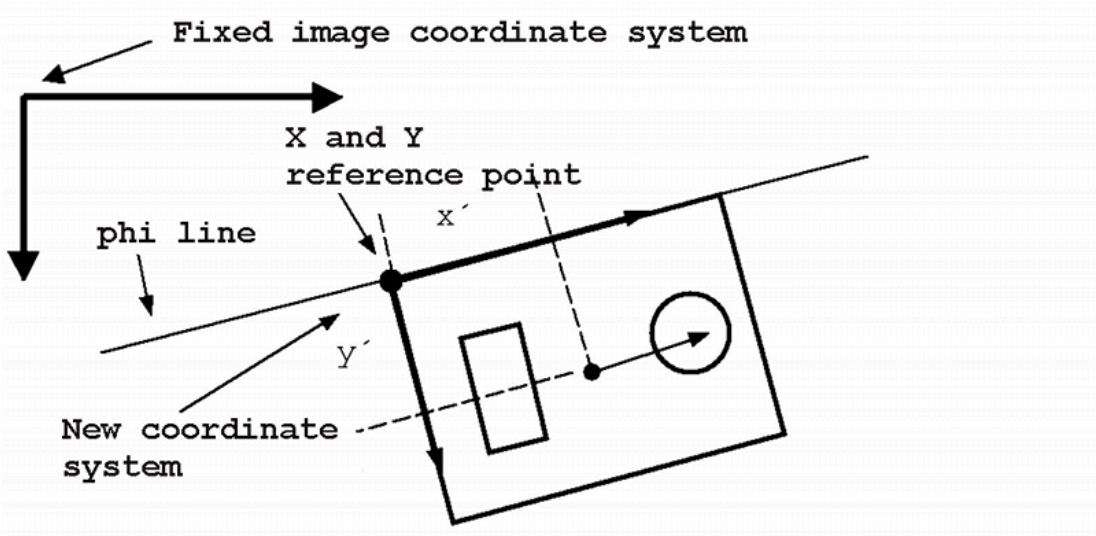


Example for position tracking in x/y direction

Angular Position Tracking

ADVICE
• The angular position tracking should be linked with the X/Y position tracking, whereby the same X and Y reference points should be used. • In order to achieve angular position tracking up to 360°, the phi line must be defined by two clearly defined points. Lines which are determined by the Locate Line command can only be used for angular position tracking up to 90 degrees.

Basics



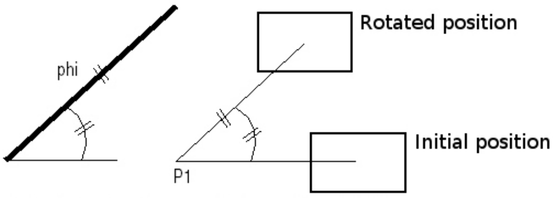
You assign the detection beams and/or middle points of the test windows to a new (rotatable) coordinate system using angular position tracking.

The X axis of the new coordinate system runs parallel to the reference line (phi line). The zero point lies in the reference point.

To track the angular position

1. Determine a reference point.
   This can be, e.g. the center of gravity from **Blob Analysis** and **Locate Contour**.
2. Execute the **Angular Position** command.
   Use the reference point for the angular position to compensate for movement of the object in the image.
3. Use the angular position line from point 2 as phi line for following commands.
   These commands – e.g. Count Pixels – must use both the reference point and the phi line to compensate.

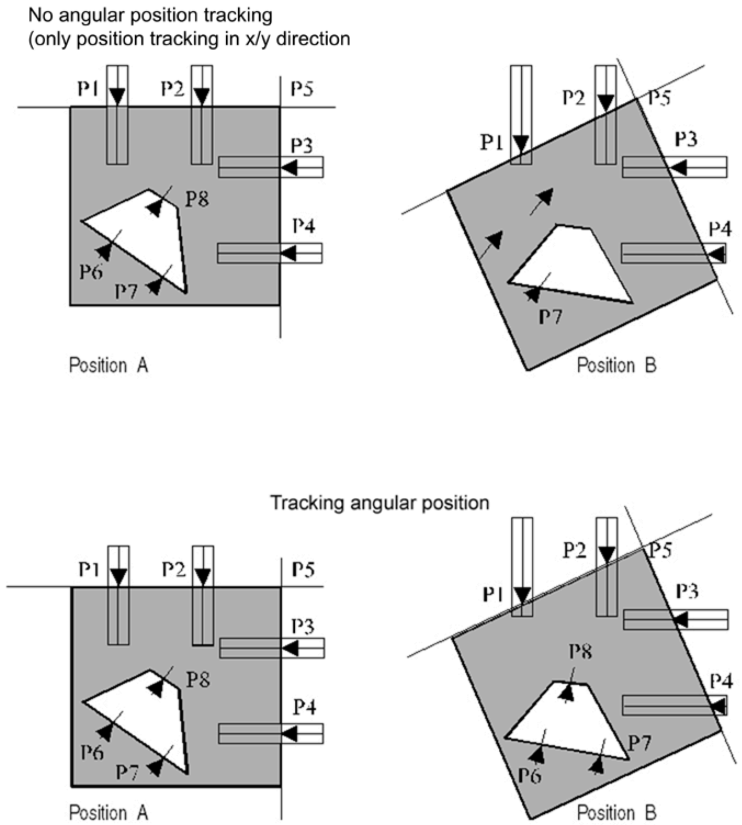
A Special Case for Angular Position Tracking



Only the center point of the window is rotated.

The rotary angle is produced from the angle between the position tracking line and the X axis. The direction and size of the window are maintained.

Example



Example 3: Tracking angular position

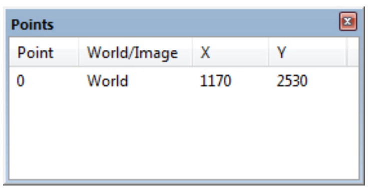
4.3.5 Checking Variables

Before using values of previously saved variables in new commands, ensure the variable contain correct values. If not, the test function in "Teach-in" will raise an error.

To check variables

1. Review the program using **Utilities > Test Step** until the point is reached where the variables concerned are used.
2. Check the variables in the results list, points list, line list, circle list or string list. You can use these variables for teach-in within further commands.

Example for points



4.3.6 Command Test

Button [Test]

Click the [Test] button in order to test the commands during teaching-in. You should do this before adding a command to the command list.

The test displays the following:

- Whether the command was **successful** or **unsuccessful**.
- The calculated values and results.

4.3.7 Online Debugging

Online debugging is a tool for diagnosing and locating errors in test programs. The function enables you to efficiently optimise your commands and programs and thus lays the foundation for successful test procedures.

For this purpose, the current test program is transferred to the RAM of the vision system and executed there. In contrast to run mode, communication with vcwin remains intact and the device transmits the line numbers of the currently executed lines/commands. The user can therefore see which commands have already been processed and respond accordingly. If the program on the vision system is waiting for an external signal (LineIO, PortIO, Port Control, Synchronous Flash, etc.), the cursor remains in the corresponding location. Through the transfer of information between the vision system and vcwin, debugging works with only a very slight delay in comparison to run mode.

To better monitor the progression of online debugging, the user can insert breakpoints. These points can also be linked to certain conditions. If a breakpoint is triggered, processing by the vision system is stopped and the evaluation of the commands processed up to that point in time are displayed in vcwin.

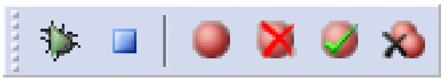


Image 12: "Debugging" toolbar

Starting, stopping and continuing debugging

To start online debugging, click either Utilities > Start/Continue Debugging or the symbol in the toolbar. The current program is then transferred to the RAM of the vision system, and existing programs in RAM are replaced (this does not apply for programs in the FLASH). As soon as debugging starts, a dialog for controlling the process appears. During the debugging process, you cannot use functions or commands in vcwin.

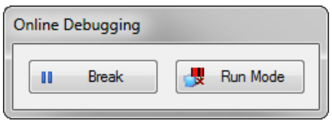


Image 13: Dialog while debugging

If you click the Break button in the dialog, the debugging process is stopped and you can work in vcwin again. The command currently being processed is indicated via an arrow symbol in the command display. In the debug history, the previous commands (including test results and duration) are listed (max. 50,000 commands), and the yellow arrow in the command display shows the line where debugging was stopped. You can utilise this "paused" state to check geometry elements or adjust individual commands, for example. While debugging mode is active, you can only change commands. They cannot be deleted or added. To continue debugging, click **Start/Continue Debugging** again.

If you click the Run Mode button, the connection between the vision system and vcwin is broken and the test program is continued in the run mode of the vision system.

To exit online debugging and switch back to command mode, click the blue stop symbol in the "Debugging" toolbar or click **Utilities > Stop Debugging**.

Breakpoints

Use breakpoints to stop your test program at specific points while debugging and evaluate the previous results. With conditional breakpoints, you can also specify that a program is only stopped if a specific condition has been fulfilled. If the debugging process is stopped by a breakpoint, the program sections processed up to that point are evaluated and displayed in the debug history. To continue debugging from the same point, simply click **Utilities > Start/ Continue Debugging** or the symbol in the "Debugging" toolbar.

The following breakpoint types are available:

Type	Symbol	Function
Unconditional breakpoint		Interrupts the running program before processing the line marked with the breakpoint.
Conditional breakpoint (good)		Only interrupts the running program if the preceding command was tested as "good". Otherwise, debugging is continued without interruption.
Conditional breakpoint (error)		Only interrupts the running program if the preceding command was tested as "bad". Otherwise, debugging is continued without interruption.

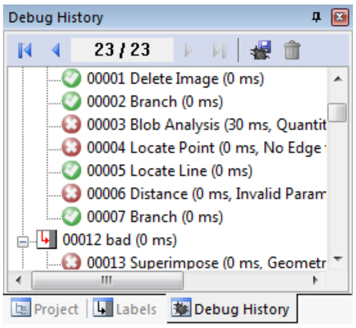
Breakpoints

A breakpoint can only be set within the command display. For this purpose, you must select the desired line and then insert the breakpoint. The breakpoint is always processed BEFORE the line in which it was inserted. This means that when you set an unconditional breakpoint for a command with number 00005, the program stops after processing command 00004. If you set a conditional breakpoint at 00005, the test result of command 00004 determines whether debugging is stopped or continues running.

There are a number of procedures for setting or deleting a breakpoint:

- Symbol in the toolbar: Click the corresponding icon in the "Debugging" toolbar. Click it again to delete the breakpoint.
- F9 shortcut key: Press the F9 key to switch between the various breakpoint types or remove the breakpoint.
- **Menu Utilities > Breakpoint**: Click the Breakpoint entry in the Utilities menu to switch between the various breakpoint types or remove the breakpoint.
- To delete all breakpoints, click the Delete All Breakpoints symbol in the toolbar or click **Utilities > Delete All Breakpoints**.

Debug History



The debug history shows the results of the processed commands of the current debugging process. It represented the last 50.000 commands. They are divided into pages of 100 entries. Use the blue arrow buttons at the top to switch between categories.

The individual entries are comprised of a command number, command designation, the time required and the error type (if present) in the brackets. If the time specification is 0 ms, less than 1 ms was required to process the command. It is not possible to add up times to determine the total time, as other processes (e.g. triggers) are carried out between commands.

Type	Symbol	Function
First/ Last Page		Goes to the beginning/ end of the debug history.
Previous/ Next Page		Goes to the previous/ next page of the debug history.
Number current page/ all pages	3 / 5	Display the current page / total number of pages.
Save Debug History		Stores the debug history on the PC.
Clear Debug History		Clears the debug history.

Type	Symbol	Function
Command "good"		The command has been rated as "good".
Command "bad"		The command has been rated as "bad".
Label Start		Starting point of the program.
Label Branch		This label indicates the command **Branch** with option "**Set Label**". All subsequent commands up to a return or a new label are grouped together under this label. For more info: *"Branch", page 367*.
Label Return		Indicates a return from subroutine. Any commands that were executed up to the next return or to the next label are listed under this entry. For more info: *"Branch", page 367*.

If you are in paused debug mode, you can quickly access the individual commands using the debug history. Clicking an entry once selects the corresponding line in the command display, and double-clicking it opens the command dialog in the case of commands, where you can conveniently make changes to parameters.

4.4 Structuring programs

For reasons of clarity and the avoidance of redundant sections, test programs are structured. The programs are subdivided into subprograms or program sections which are linked together according to specific rules. The most important program structuring techniques are test program branch and subroutine engineering.

Advantages:

Advantages:

- Programs are becoming clearer and shorter.
- The individual subroutines or program modules can be easily exchanged if necessary.
- Reduced change effort. Changes in the subroutine affect all areas of the program from which the subroutine is called.

4.4.1 Program branch

Test program branches are used in test programs in connection with the subprogram technique or with marks.

The following control commands can be used to branch the program

- Branch
- Port Control
- Indexed Branch
- Line I/O
- Port I/O

Program branches can be configured

- as unconditional branch (always branches)
- as a conditional branch depending on the result of a previous command or command block or the status of an input

Test program branch via the device inputs

The test program branches take place here via a status change or a signal at the device inputs. The commands Line I/O, Port I/O and Indexed Branch can be used for this.

Test program branching via internal result variables

In order to branch test programs via the value of a result, a result must be linked to a value and the result of this operation must be queried. The linking operation is successful if the result of the operation is the same as the target result or is within the set tolerances.

Test program branching via external interfaces

Results can be set externally (e.g. PC) via the interface with the Link results command. The test program branches to a label or a subprogram depending on the value set.

4.4.2 Subroutine technique

The subprogram technique is used when commands that are arranged in a certain order are used several times in a test program. The commands are stored in a program block that can be called up using its name.

Types of programming subroutines

- direct programming of subroutines
  These subroutines jump back to the entry point in the main program by *Return*.
- calling subroutines by jumping to labels
  For subroutines with labels, an additional label is required, which should be jumped to after the subroutine has been processed.

Subroutines can be called conditionally and unconditionally.

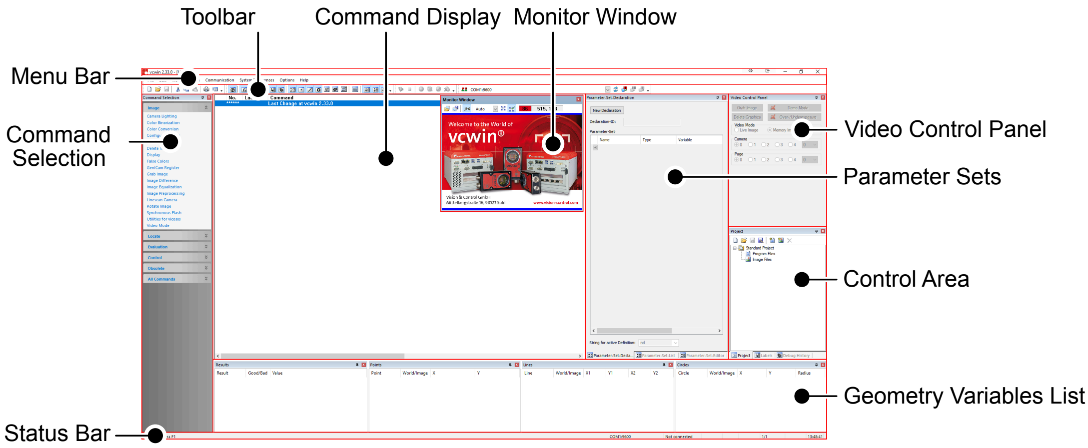
Conditional calls can be made by following operations:

- successful command
- failed command
- successful block
- failed block

Subroutines can be nested in one another, i.e. one subroutine can call another. The return to the main program takes place with the *Return* instruction. This instruction must be given.

5 PART 1 - ELEMENTS OF THE USER INTERFACE

5.1 Overview over the Programming Window



The user interface is opened on the first start with the programming window. The programming window allows, among other things, the establishment of a connection to the vision system, the creation of test programs and the parametrization of commands.

Elements of the Programming Window

Menu Bar

- Area with drop-down menus in which commands are grouped by category.
- Categories: File, Edit, View, Utilities, Communication, System Preferences, Options, Help

Tool Bars

- Free-floating bars with menu options and several commands.
- Categories: Default, Docking Windows, Debugging, Communication

Status Bar

- Area where system and program information is displayed.
- Categories: File, Edit, View, Utilities, Communication, System Preferences, Options, Help

Command Display

- Area in which the editable commands of the test program are displayed and edited.

Docking Windows

Monitor Window

- Window for displaying live or memory images, as well as the overlay.

Command Section

- List of all image processing commands, sorted by categories.
- Categories: Image, Locate, Evaluation, Control, Obsolete, All Commands

Video Control Panel

- Window for controlling the image acquisition in command mode.

Control Area

- Window for managing image processing projects, program flow control and online debugging.
- Categories: Project, Labels, Debug History

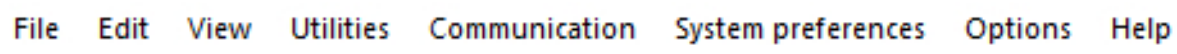
Parameter Sets

- Window for declaration and definition of Parameter-Sets.
- Categories: Parameter-Set-Declaration, Parameter-Set-List, Parameter-Set-Editor

Geometry Variables List

- Window for displaying the geometry variables determined during program execution.
- Categories: Results, Points, Lines, Circles, Strings, Contours, Test Windows

5.2 Menu Bar



The menu bar is located in the upper left corner of the Programming Window. It is an area with a drop down menu where commands are grouped by categories.

The menu bar can be individually customised (*see "Customize Toolbars", page 23*).

In the default setting, the menu bar contains the following menus:

Menu	Description
File	Functions related to the management of test programs. *see "The File Menu", page 49*.
Edit	Tools for editing an open test program. *see "The Edit Menu", page 53*.
View	Showing and hiding individual objects of the user interface as well as changing the surfaces. *see "The View Menu", page 62*.
Utilities	Tools for testing and setting up test programs. *see "The Utilities Menu", page 68*.
Communication	Tools for organizing the communication between PC and vision system. *see "The Communication Menu", page 72*.
System Preferences	Tools for system setting and configuration of the communication of the vision system. *see "Menu System Settings", page 81*.
Options	Tools for changing the appearance of the user interface and the configuration of the user management. *see "The Options Menu", page 101*.
Help	Calling the help and the program information. *see "The Help Menu", page 107*.

5.2.1 The File Menu

The **File** menu contains ommands that relate to complete test programs.

In the drop-down menu you will find:

- Commands for creating, opening, closing, and saving test programs
- Tools for handling projects
- User information, as well as print settings and page setup
- Recently opened files (if present)
- Command to terminate the operating software

5.2.1.1 New

File > New is used for creating a new empty test program.

Any test program currently open will be closed automatically. If data has not been saved after being edited, you will be given the chance to save.

5.2.1.2 Open

Use **File > Open** to open an existing test program. The program to be opened can be either selected from the selection box (path: to the right, file name: to the left in the menu) or typed into the File Name field.

Files with the ***.vc** and ***.*** extensions can be loaded.

Only a single test program can be open at any one time.

Open the selected program with the [Open] button.

If you have already opened a test program which has not been saved since the last time you edited it, you will be given the chance to save it.

Options during Opening

Preview: After the program name is selected, further information will appear by using the preview option of your Explorer.

- Name of the user
- Creation date
- Date of the most recent edit
- Comments

The according information is written during editing into the test program via **File > Info** and is saved together with the program itself by calling **File > Save**. >> *"Info", page 51*

After Opening

After opening a new test program, you can insert, modify or delete commands with the Editor.

5.2.1.3 Save

File > Save is used for saving the current test program and all program-specific settings to disk. While you are creating/editing test programs, any changes you make will only change the data in memory.

The **Save** command stores the edited test program, file information and all relevant settings to harddisk, overwriting the original. The test program remains open.

ADVICE

Newly created and edited test programs occupy volatile memory in the computer. This means that the data will be lost if the computer is switched off. To ensure that the edited test programs will be available the next time you start the system, you should always remember to call **File > Save** before switching off.

5.2.1.4 Save as

With **File > Save as** you can save a copy of the opened test program under a different name. The original program is kept under the original name.

Unsaved changes are not written back to the original program, but only to the created program copy.

5.2.1.5 New Project

Open a new empty project with **File > New Project**.

"Docking Window Project"

5.2.1.6 Open Project

You open an already saved project with **File > Open Project**.

see "Docking Window Project", page 123

5.2.1.7 Insert

With **File > Insert**, you can add an **image file** or **program file** to the current project.

Select **File > Insert**, followed either by an **image file** or a **program file** and then the required files in **Open** dialog box. The files added are displayed in the Project Window.

Use the Project Window icons or context menu as alternative to add image files or program files to your project.

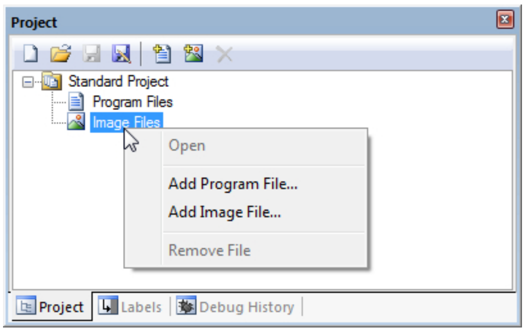


Image 14: Project Window context menu

5.2.1.8 Save Project

You save a new project or the changes to an edited project with **File > Save Project**.

see "Docking Window Project", page 123

5.2.1.9 Save Project as

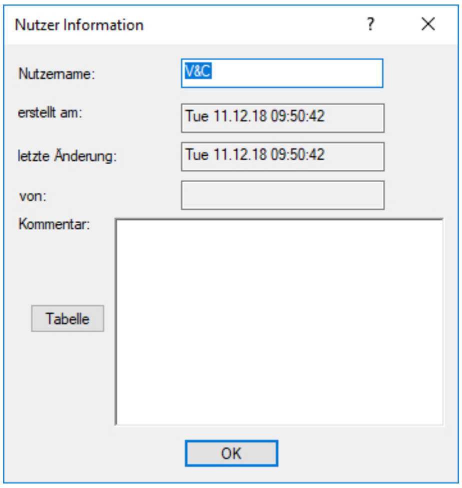
With **File > Save Project as** you save an existing project or the changes in an edited project under a new name in any directory.

see "Docking Window Project", page 123

5.2.1.10 Close Project

Close the current project with **File > Close Project**.

5.2.1.11 Info



The **File > Info** command is used for adding further information about the program:

- Name of the user
- Creation date (display only)
- Date of the most recent edit (updated)
- From: Name of the editor (updated), if the user logged in via **Options > User Login**.
- Comment
- Table with maximum 16 fields for additional information

All this information is saved together with the test program whenever the **Save** or **Save As** commands are called.

5.2.1.12 Print

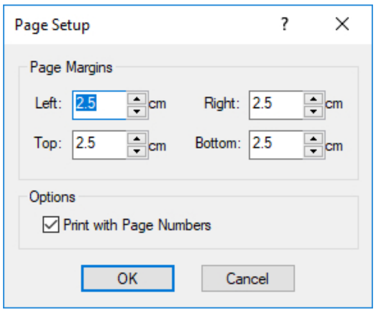
The **File > Print** command opens the operating systems standard print dialog box so that the current test program list can be printed. For more information about printing documents please refer to your operating system manual and the printer manual.

5.2.1.13 Print Preview

The **File > Print Preview** command opens a preview for printing the current test program lists.

Use this dialog box to check the layout prior to printing. Click the [Print] button at the top left in the dialog box to start the print procedure.

5.2.1.14 Page Setup



With **File > Page Setup** you set the width of the margins and the optional output of the page number for the printout of the test program.

The entered parameters are saved in an initialization file.

New test programs automatically use the given default settings for margins.

5.2.1.15 Print Setup

This menu item opens the printer setup menu of the operating system.

5.2.1.16 Exit

With this menu item you exit the user interface.

If you select the **Exit** command without first saving all changes, the user interface will ask you if you want to save your changes.

Select [Yes] if you want to save your changes, or [No] to close the user interface without saving the changes.

5.2.2 The Edit Menu

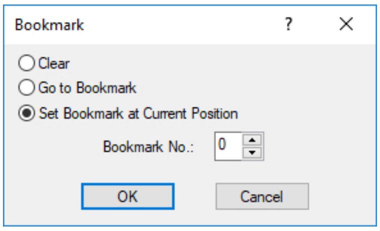
The **Edit** menu contains functions used for editing an open test program.

In the drop-down menu you will find:

- Bookmarks to find specific parts of the program, the command to change dialogues, and command selection for classification of user right linked commands.
- Options for searching text, labels or commands
- Commands to cut, copy, paste, delete and undo
- Command to save and load a command block
- Option to enable and disable individual commands
- A comment function for documentation • the name assignment of variables

5.2.2.1 Bookmark

Overview



Bookmarks provide quick access to command lines of a test program. Each bookmark receives a number from 0 to 100. Bookmark numbers can be assigned several times.

Create bookmarks

1. In the command window, select the command line you want to bookmark.
2. Open the dialog via **Edit > Bookmark**.
3. Activate the option "Set Bookmark at current position".
4. Enter a bookmark number.
5. Close the dialog with [OK].

The bookmark number is written to the command line before the command.

```
No.    Label  Command
00001         Standard Image Acquisition with camera 4
00002         Bookmark 0:
              Determine Line 3 (World Coordinates); 10 Single Detec-
              tions to ..
              Search Arrow: [557, 154/472, 154];
              Max. Gradient [5, 1, 1]; Dark > Light
```

TIP
Use the keyboard shortcut [CTRL] + [0 ... 9] to quickly assign a bookmark number from 0 to 9 to the selected command line.

Switch to bookmarks

1. Open the dialog via **Edit > Bookmark**.
2. Activate the option "Go to Bookmark".
3. Enter a bookmark number.
4. Close the dialog with [OK].

The program switches to the command line.

TIP
Use the key combination [ALT] + [0 ... 9] to quickly switch to the bookmark number.

Remove bookmarks

1. In the command window, select the command line from which the bookmark should be removed.
2. Open the dialog via **Edit > Bookmark**.
3. Activate the option "Clear".
4. Close the dialog with [OK].

The bookmark number is removed.

5.2.2.2 Edit

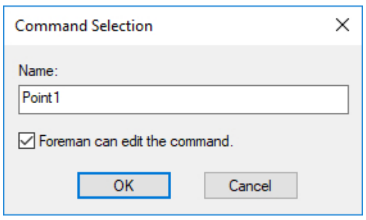
In order to open the corresponding teach-in dialog box, in which all associated parameters can be edited, use **Edit > Edit**. Alternatively, you can open the teach-in dialog box as follows:

1. Select the program line.
2. Double click on the line or press the Enter key.

5.2.2.3 Command Selection

With **Edit > Command Selection**, the programmer can decide whether the current command may also be changed by the foreman. This is recommended e.g. for commands relating to setting tolerances.

Under **Name**, enter a relevant name.



The commands that can be changed by the foreman are indicated in the program window by an [M].

```
No.    Label  Command
*****         Last Change at vcwin 2.17.136
00001  Start
00002         ForemanCommand Determine Line 25(M):
              Determine Line 25 (World Coordinates];
              30 Single Detections to 53 Pixels; Search Arrow:
              [69,102/..
```

If later, a user logs on with **Foreman** rights, he is then shown only the commands in the program that he has a right to change.

```
No.    Start  Command
*****         >>> Edit Selected Commands <<<
00002         ForemanCommand Determine Line 25
```

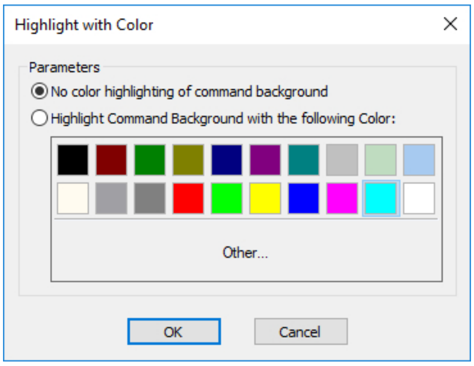
The foreman changes commands as follows:

1. Double-click on a command.
2. Change the parameter in the dialog box.
3. Save the program.

5.2.2.4 Highlight with Color

Overview

With **Edit > Highlight with Color** you can highlight commands in the Command Window with color. The formatting is stored in the program.



Highlight commands with color

1. Mark the commands in the Command Window.
2. Open the dialog via the **Edit** menu **> Highlight with Color**.
3. Select the background color. The "Highlight Command Background with the following Color" option is automatically enabled.
   a) Choose from 20 predefined colors or
   b) Click on the button [Other...].
   Choose a color from the Windows system dialog. Close the dialog with [OK].
4. Close the dialog with [OK].

The marked commands are highlighted with the selected color.

Remove colored marking

1. Mark the commands in the Command Window.
2. Open the dialog via the **Edit** menu **> Highlight with Color**.
3. Activate the option "No color highlighting of command background".
4. Close the dialog with [OK].

The background color of the commands will be unmarked.

TIP
You can turn off the colored markings of the program temporarily. Deactivate the option "Display Highlight Colors" in the **Options** menu. This setting is a function of the operating software and is not stored in the program.

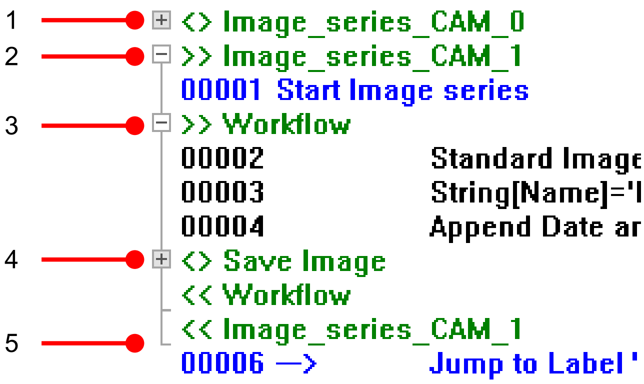
5.2.2.5 Group Commands

Overview

The function **Group Commands** is located in the **Edit** menu.

With **Group Commands** you can group any number of consecutive commands into a group. The formatting is stored in the program.

Elements of a Group

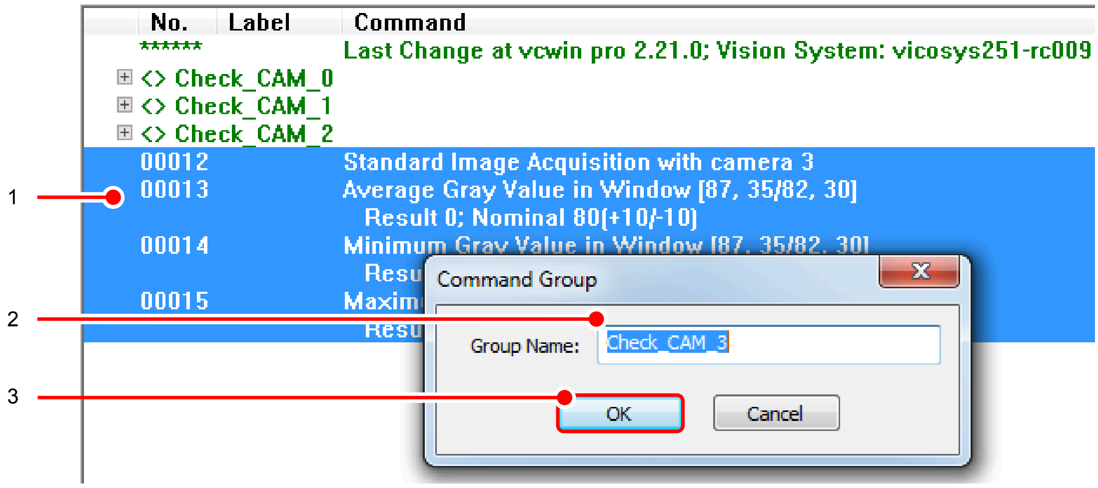


1 ⊞ and <>*Groupname* indicate a collapsed Group. Click ⊞ to expand the group.
2 ⊟ and >>*Groupname* indicate the start of an expanded group. Members of a group are indicated by a solid line. Click ⊟ to collapse the group.
3 ⊞ indicates a collapsed subgroup.
4 ⊟ indicates an expanded subgroup.
5 └ and <<*Groupname* indicates the end of an expanded group.

Working with groups

- Each group is defined by the beginning and the end of the group. In the expanded state, the group name is at the beginning of the group and at the end of the group. Group names can be assigned more than once.
- When selecting collapsed groups, all commands in the groups are automatically selected. In this case, the function will apply to all commands of the group (e.g. delete, search).
- When selecting the start group name or end group name of an expanded group, functions like delete, cut, copy, or "Save Block in File" cause that the other related group name is deleted, cut, copied or stored too. The content of the group is not affected by the functions.
- Multiple groups can be combined to a new group.
- In a group sub-groups can be created.
- New groups must include complete existing groups.

Create a group



1. Mark in the command window the commands that should be grouped.

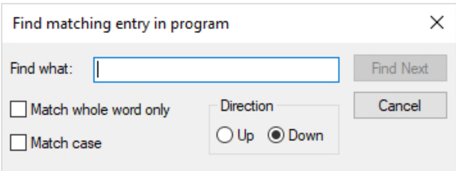
   Open the dialog via **Edit** menu **Group Commands**.
2. In the dialog, enter the name of the command group.
3. Close the dialog with [OK].

   The selected commands are grouped together.

Ungroup a group

1. Expand the group to be ungrouped. When deleting a collapsed group the commands in the group will be deleted.
2. Select the group name and press the [Delete] key on your keyboard.

5.2.2.6 Find



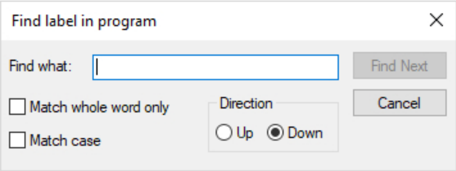
With **Edit > Find**, you open a dialog box, then type the character string you want to search for.

Searching starts at the current line and will stop when the character string is found or the end of the program is reached.

5.2.2.7 Find More

With **Edit > Find More** you continue a search procedure.

5.2.2.8 Find Label



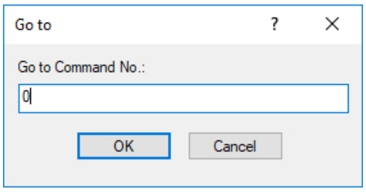
The search for a label applies to the entire program.

1. Enter the name of the label in the field *Find what:*
2. Start the search with [Find Next].
3. Focus will jump to the label if the search is successful.

The search starts at the cursor position and, depending on the search direction, runs through to the beginning or end of the program or to the searched mark.

Depending on the setting, a distinction between uppercase and lowercase letters is made.

5.2.2.9 Go to Command



The search for a command number includes the entire test program.

1. Enter the command number you want search for in the input field.
2. Confirm.
   The entered command is selected.

Searching starts at the first line and runs until the specified command or the end of the program is reached.

5.2.2.10 Cut

You can copy sections of open test programs into the Windows clipboard using **Edit > Cut**. This removes the cut section from the test program. The cut section can then be re-inserted into the open test program in the position required via **Edit > Paste**.

5.2.2.11 Copy

You can copy sections of open test programs into the Windows clipboard using **Edit > Copy**. The copied section can then be inserted into the position required in the open test program via **Edit > Paste**.

5.2.2.12 Paste

You can insert sections of test programs from the clipboard into the position required in the open test program with **Edit > Paste**.

5.2.2.13 Delete

Remove sections of a test program using **Edit > Delete**.

5.2.2.14 Undo Command

With **Edit > Undo Command** you can remove the last commands that were inserted into the program or reinsert the previously deleted commands.

5.2.2.15 Save Block in File

Commands in an open test program can be saved as a block of commands under a new program name. In order to do this, all the commands required to be saved should be marked. The original program is not changed in any way.

5.2.2.16 Insert File

Edit > Insert File is used for inserting a program block or a complete test program into the program currently being edited. Select the file in the shown dialog box.

Option

Preview: If you use the Preview function of the Windows dialogue, the following additional information appears:

- Name of the user
- Creation date
- Date of the most recent edit
- Comment

You can insert this additional information with **File > Info** into the test program and then save it with **File > Save-Info**.

5.2.2.17 Activate / Deactivate

Deactivate and activate the commands currently highlighted in the test program for test purposes using **Edit > Activate / Deactivate** respectively. Deactivated commands are ignored by subsequent program testing.

5.2.2.18 Insert Comment

Enable the documentation of commands using **Edit > Insert Comment**. Select the command line under which a comment should be entered and open the Comment window with **Edit > Insert Comment** or with [F2].

Comments are shown as separate lines. In the dialog, you can write multi-line comments with up to 1000 characters per line, which will then be inserted in each case as a single line in the command display. *see "Comments", page 117*

The colour of a command is green (standard). The color of the commentary text can be altered via **Options > Application Settings**.

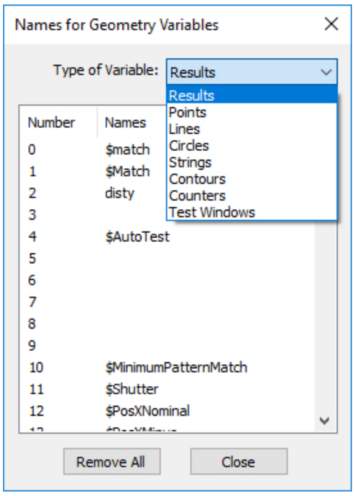
5.2.2.19 Names for Geometry Variables

You can assign identifiers (names) to the geometry variables with **Edit > Names for Geometry Variables**. Use this function in order to make your test program more readable.

You can also set the names for geometry variables during actual test command parameterization.

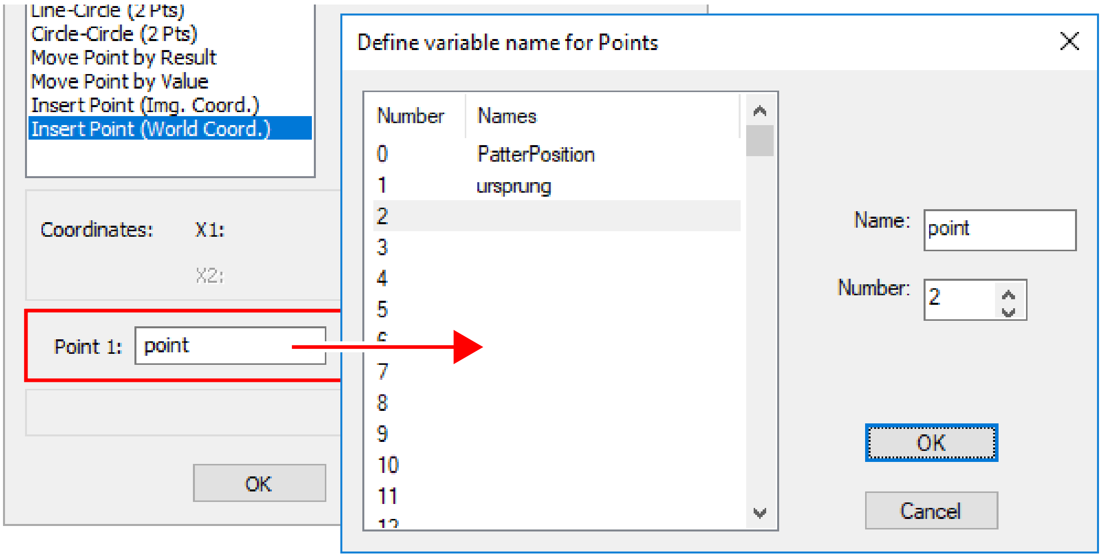
The geomery variables are subdivided into variable types. Each variable type has its own namespace.

Define names via Menu Edit



1. Select **Edit > Names for Geometry Variables.**
2. Select the Type of Variables from the drop-down menu.
3. Click on the geometry variable that you want to name on the **Names** column, next to the **geometry variable numbers**.
4. Enter the name.

Define Names in Dialog

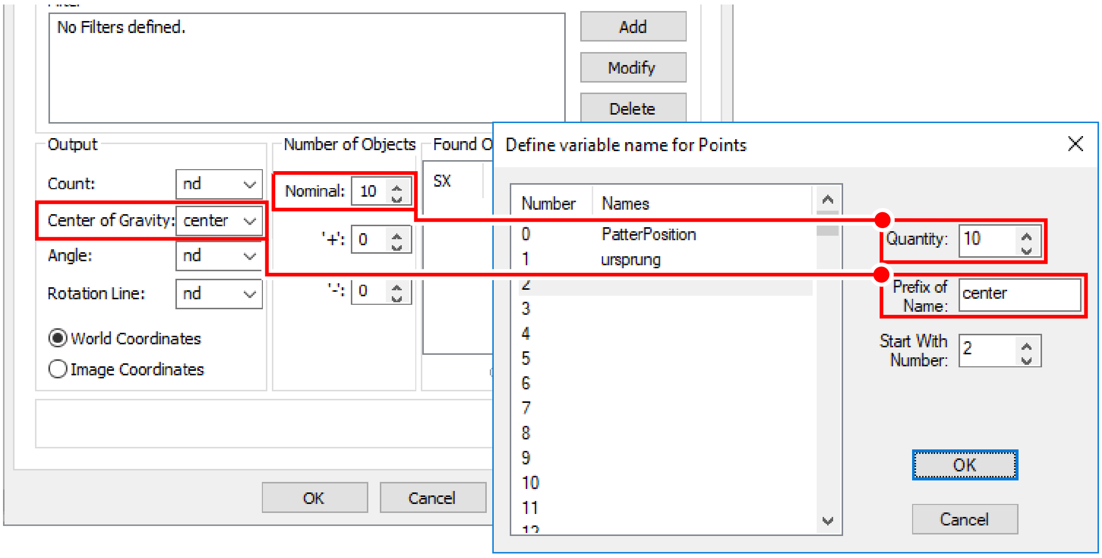


1. Enter a new name into the geometry variable output field in the dialog, then click in an other output field.
   A window for defining a new variable name appears. You see names that are already in use and you can define a new name here.
2. In this dialog, enter the name of the variable in the textbox **Name**.
3. Choose among the number of the geometry variable that you want to rename.

Define names in dialog - Multiple definition

Some commands allow, to determine and save several variables of one parameter in one step. For example **Rotation based on Moments > Multiple Objects**.

The variables get their name automatically by the entered name and an index that is attached.



Parameters	Description
Quantity	Number of indexing. The Quantity is the sum of the nominal value and the highest positive tolerance.
Prefix of Name	Prefix of the name. It is taken from the declaration section (here: Center of Gravity). But it can be adjusted.
Start With Number	From this position in the variable list, the results (Prefix of name and index) are stored.

5.2.3 The View Menu

The **View** menu allows to display or hide user interface objects and switch to the user interface.

In the drop-down menu you will find:

- switching to the user interface
- a list of all docking windows and options for showing and hiding them
- a list of all toolbars and options for showing and hiding them
- Option to show and hide the Status Bar

5.2.3.1 User Window

With **View > User Window** you switch to the respective selected user window

The User Windows are specific user interfaces for different tasks. You can select among the following options:

- Standard User Window … used for the execution of completed programs
- Measuring User Window … used for good/bad evaluation and measurement statistics

The user window can be determined in the programming window via **Options > Application Settings > Look.**
>> *see "Application Settings", page 102*

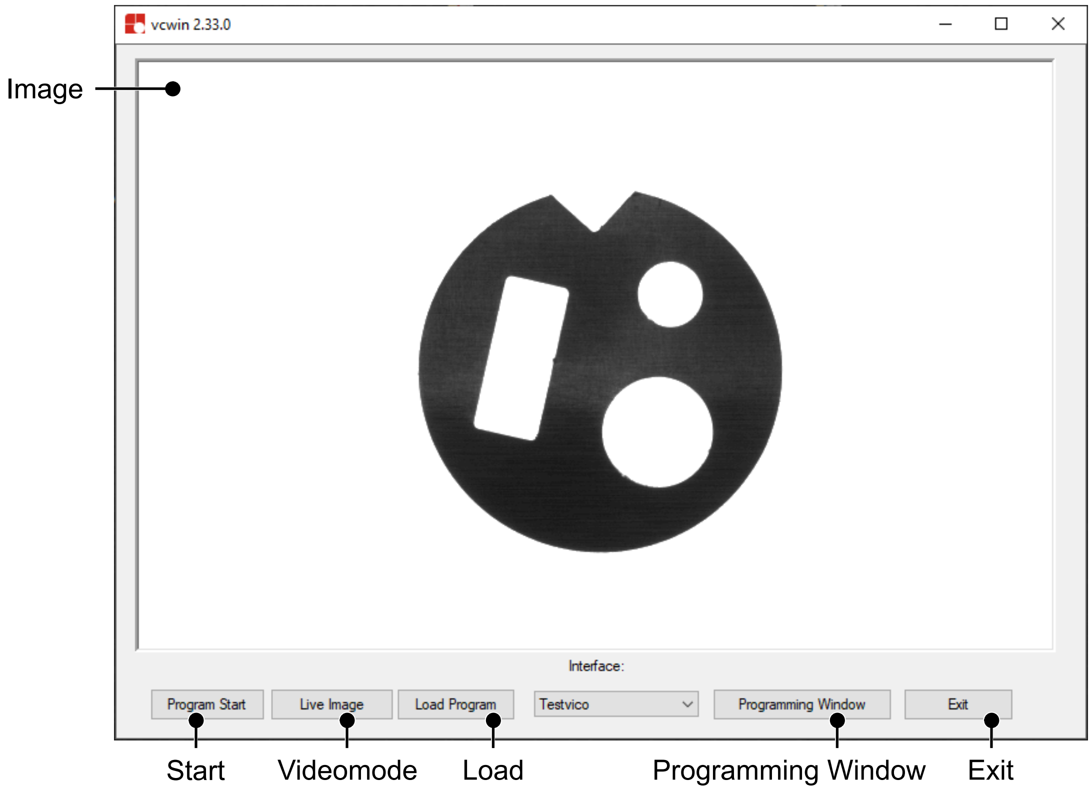
5.2.3.1.1 Standard user interface

The standard user interface is a tool with that the image processing task can be visualized, during the runtime of the vision system.

With the standard user interface, you can:

- start and stop the currently loaded program
- alternate between live- und memory image

- display of transmitted images
- load a program on the vision system
- switch to the programming window (user right based)



5.2.3.1.2 Measurement User Window

The measurement user window is a tool with which an image processing system manages, controls and archives a machine application. The statistics that are displayed show the relationship between positive and negative test procedures.

The user window maintains information independently from the programming window. The user window filters a great deal of data out of the test program when it is loaded. In order that the user window can work correctly, it is therefore essential that the test program in the image processing system is identical with the test program last loaded by the user window.

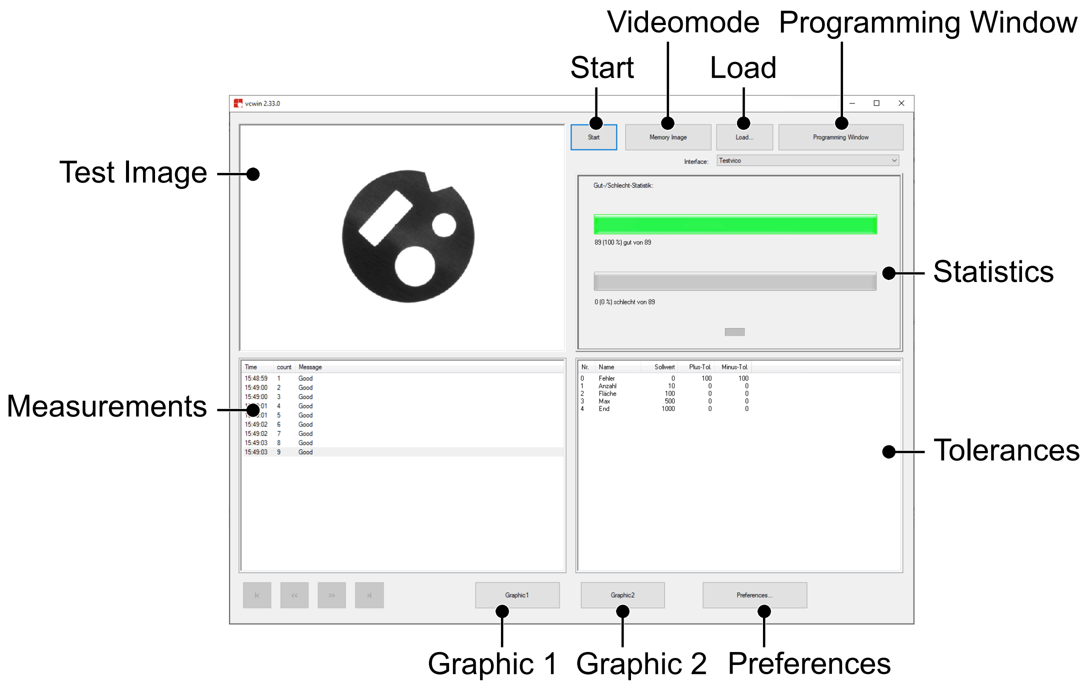


Image 15: Dialogue Measurement User Window

Load

The **Load** button transfers a test program optionally to working storage or to the image processing system's flash. During this process, the user window extracts information from the test program about results and images to be sent.

Start/Stop

With the **Start** and **Stop** buttons, you start the test program that is currently loaded or interrupt it. **Stop** also initialises the vision system.

Programming window

With the **Programming Window** button, you arrive at the programming dialog. This button can be protected by a password against unauthorised access (*see "User Management", page 105*).

Image window

The image window (upper left) contains the latest test images that were transferred during run time. All detections and superimpositions are shown in this window.

Measurement values

In the measurement values window (lower left), data and parameters determined during the test program are displayed.

The **Graphic 1/ Measurement Values** button lets you choose to have the measurement values displayed either as a table or as a graph. In the graphic representation, green is used for good evaluations and red for bad evaluations.

It is always displayed only one measurement value. You can set them by using the **Preferences** button.

Tolerances

In the tolerances window (lower right) the tolerance areas parameterised in the test program are shown. The relevant parameters are:

- Upper and lower tolerance thresholds
- Nominal size
- Good measurement values (green)
- Bad measurement values (red)

The **Graphic 2/ Measurement Values** button lets you choose to have the measurement values displayed either as a table or as a graph. In the graphic representation, green is used for good evaluations and red for bad evaluations.

It is always displayed only one measurement value. You can set them by using the **Preferences** button.

Change Tolerances

To adjust tolerances double-click on the corresponding measurement line. The dialogue **Measuring Values - Edit Presents** opens. Set the desired new values here.

The tolerances are changed and stored in the respective program on the PC. Upload the program with the button [Load] to the vision system, to apply the changes.

Bar chart/statistics

The bar chart (upper right) gives you a statistical representation of the process variables. It contains good/poor statistics for a percentage-based evaluation of the determined values.

Settings dialog

The settings dialog allows the user to save events and images in the computer that were received during the process.

Measurement values and statistics can be reset by the user if necessary. In addition, automatic reception of images and data during program start can be enabled or disabled by means of a check mark.

Save measurement values

If the **Save Data** option is enabled, measurement values that were received are stored in a text file with the name **Log File** or with a user-defined name, as the case may be. If no path is given, the storage location for the data relates to the installation folder. Measurement values are stored line for line in the text file.

Each line contains the following information:

- Date & time: (coded in time_t: seconds since 1.1.1970)
- Block ID
- Overall result (0: good, 1: bad)
- List of measurement values

Saved measurement values can be loaded back to the user window via **Settings > Load Measurement Values**. To allow the data to be interpreted correctly it is essential that the test program that was last loaded is identical with the test program used for saving the values.

The current statistics are deleted with the **Reset good/bad statistics** button.

Saving images

Images are saved in the path supplied under **Settings > Image Files**. A file prefix defines the first part of the file name. The rest of the file name is made up from the date and time. The following options can be selected for the image file:

- Displays of the images
- Save image
- Circular buffer (maximum 10,000 images)

Display images

In order to display images with the user window, the **Send Image** command must be used in the test program (*see "Send Image", page 335*). The user window detects the settings for this command when loading the test program and displays the image after it is received. The **Settings > Display** option in the **Settings > Image Files** dialog must be enabled in order to display the images.

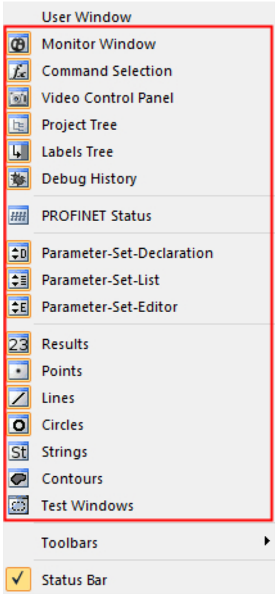
Display results

Results can be displayed in the user window by using the **Evaluate Result** command in the test program with the send option enabled (**Send Measurement Values**) and by configuring the interface accordingly. When transferring via the serial interface (**/com**) the setting must be made according to that of the image processing system in **Programming Window > Communication > Interface**.

For communication via Ethernet, the image processing system must be set as the TCP server on port 8500. This is done via the **/ethernet** selection, in the field for the measurement window data. The settings for Port and IP remain empty. The overall result from **Evaluate Result** will also be sent to the user window. The overall result is used to produce the displayed statistics. The block ID must be between 0xD0 and 0xDF for the display in the user window. The names of the parameters (e.g. **Test, Test2 and Test3**) as well as target values and tolerances are interpreted by the user window. The names of the parameters are displayed as headings. The target values and tolerances are used for the graphic representation.

5.2.3.2 Docking Windows

In the menu **View** you can show or hide all available docking windows.

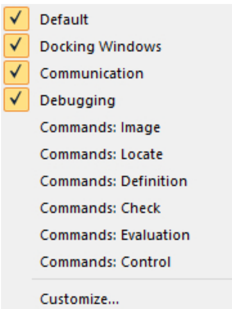


The individual docking windows are described in the chapter *"Docking Windows", page 118*.

A description of how to handle docking windows can be found under *"Customize Docking Windows", page 22*.

5.2.3.3 Toolbars

In the menu **View > Toolbars**, you can show or hide all available toolbars.



Die einzelnen Symbolleisten sind im Kapitel *"Toolbars", page 107* beschrieben.

A description of how to handle toolbars can be found under *"Customize Toolbars", page 23*.

5.2.3.4 Status bar

In the menu **View > Status bar** you can show or hide the status bar at the bottom of the user interface.

The individual elements of the status bar are described in Chapter *"Status Bar", page 114*.

5.2.4 The Utilities Menu

The **Utilities** menu contains tools for testing and setting up test programs.

In the drop-down menu you will find:

- Online Debugging
- Options for testing of commands and the step test
- Options for testing of command sections and programs
- Functions to set and clear breakpoints
- Image Report
- Options for image transfer (send and receive)
- Options for resetting of geometry variables
- Functions for testing the I/O hardware

5.2.4.1 Start/Continue Debugging

Start/continue the online debugging. (*see "Online Debugging ", page 42*).

5.2.4.2 Stop Debugging

Stop the online debugging (*see "Online Debugging ", page 42*).

5.2.4.3 Test Command

ADVICE
Use the time protocol in the status bar to estimate the duration of commands and to locate perpetual commands.

Test the first selected command using **Utilities > Test Command**. The result of this test is displayed in the status bar.

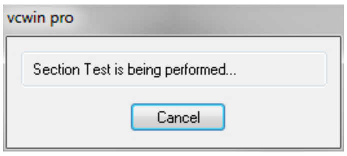
5.2.4.4 Test Step

Test a program step-by-step using **Utilities > Test Step**:

1. Highlight the command with which you want start the test.
2. Call **Utilities > Test Step** or press the F7 key. The command line is tested. The result appears to the left in the status bar and the command marking steps to the next command line.
3. Repeat step 2 as often as required.

The function includes the command call as well as displaying the results of the commands in command mode.

5.2.4.5 Test Section



You test a marked section of the test program with **Utilities > Test Section.**

The section is run repeatedly until it is actively stopped. The result of the test is displayed in the status bar.

The function includes the command call as well as displaying the results of the commands in command mode. Program control commands are not interpreted by the host computer, i. e., branches are not implemented in testing.

Use the **from Current Command** function to test the section from a breakpoint to the end of the section.

5.2.4.6 Test Program

The entire program can be tested via **Utilities > Test Program**. The result of this test is displayed in the status bar. On reaching a breakpoint, the breakpoint is noted and the program halts.

Use the **from Current Command** function to test the section from the command line marked until either another breakpoint is reached or to the end of the program.

5.2.4.7 Breakpoint

Use **Utilities > Breakpoint** to set a breakpoint in the test program. Multiple breakpoints are possible in the test program. Breakpoints are highlighted in colour in the program.

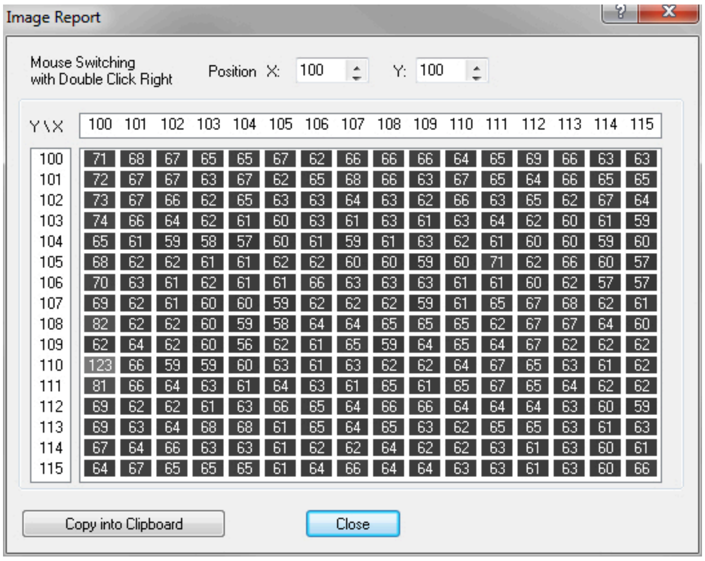
During **Online-Debugging** and when testing the test program with the commands **Utilities > Test Section** or **Utilities > Test Program**, the program execution is stopped at the breakpoint.

To deactivate a breakpoint, unmark it in the test program.

Breakpoints are ignored in the run mode.

5.2.4.8 Image Report

Image Report allows you to display the gray values of each pixel in a defined area.



You can define the image window by entering the position numerically or directly with the mouse.

To enter the position with the mouse on the video screen and to switch between the PC monitor and the video screen, double click with the right-hand mouse button. The grey values of your position are determined and displayed immediately. For better visualization the values are underlayed with the corresponding gray value of each pixel.

Parameters	Description
Copy to clip-board	This is to copy the report files to the clipboard. Then the data can be pasted directly into Microsoft® Excel, for example.

5.2.4.9 Receive Image from Vision System

With **Utilities > Receive Image from Vision System**, you save the contents of the image memory of the vision system in a JPG or BMP file..

The JPG quality is set to 75% by default. You can adjust the compression level in the **Options > Application Settings > Advanced** menu *see "Application Settings", page 102*.

Under *Save as Type,* you can specify whether the image is saved as BMP or JPG, as well as with or without overlay.

5.2.4.10 Send Image to Vision System

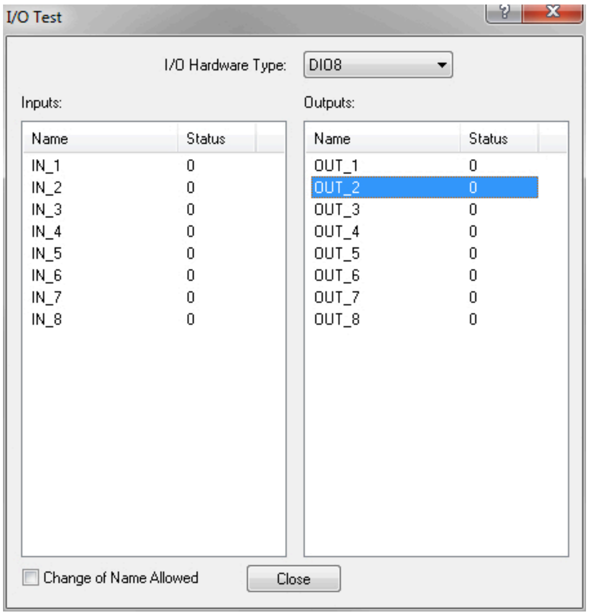
With **Utilities > Send Image to Vision System** you load a JPG or BMP file into the image memory of the image processing system.

5.2.4.11 Reset Geometry Variables

Use **Utilities > Reset Geometry Variables** to set all geometry variables which have previously been determined by the program execution or tests to **undefined** in the vision systems memory. The display of the geometry list is then deleted.

When programming the next free element is always offered for selection, therefore you should reset the geometry variables before creating new programs.Storage locations occupied by old geometry variables are skipped during programming and are not available in the current program.

5.2.4.12 I/O Test



Use **Utilities > I/O Test** to test the digital inputs and outputs e.g. when connecting to a PLC. You can assign defined signal levels or impulses to the outputs for this purpose. All input lines are scanned repeatedly.

You can do the following tests:

- Test the state of an input.
- Switch output signals manually.

Input Parameters

Parameters	Description
Hardware Type	Select the hardware type for the I/O port. The options shown depends on hardware you use. Choose between PS8 (4 digital input and 4 digital output), DIO4/6 (4 digital input and 6 digital output), DIO4 (4 digital input and 4 digital output), DIO8 (8 digital input and 8 digital output) or DIO16 (16 digital input and 16 digital output).
Inputs/Outputs	Display of inputs and outputs and their states.
Change of Name Allowed	If activated the user may determine own names for the inputs and outputs. To do so, click on the input or output you want ro rename and click again shortly after. Now type in the new name and confirm it with [ENTER]. The name is saved in the operating software.

I/O Testing Method

The method of testing outputs is practically the same for all types of hardware.

The available inputs and outputs are each displayed in a window.

All inputs are scanned repeatedly and the status is displayed. You can set outputs by using the dialog box

5.2.5 The Communication Menu

The **Communication** menu contains functions for organising the communication between the host computer and the vision system.

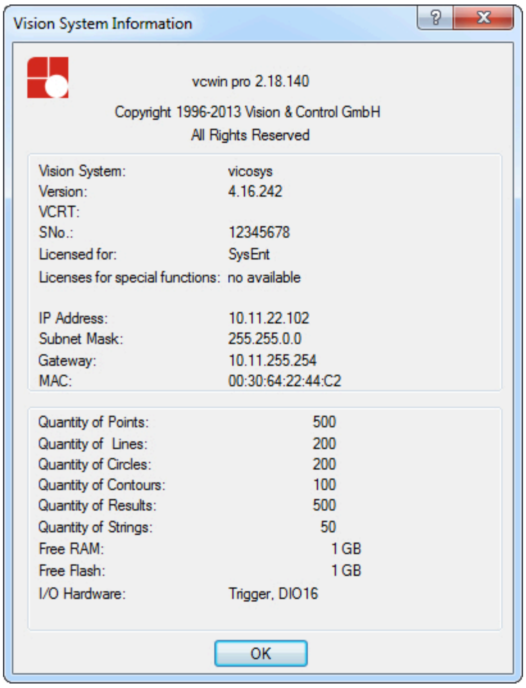
In the drop-down menu you will find:

- Functions to connect to the vision system, for program transmission and run of the vision system
- Settings for the interface to be used
- Offline Settings
- Information about the vision system

5.2.5.1 Connect

ADVICE

The connection must be established before teaching in test programs or transferring a test program from the computer to the vision system.



With **Communication > Connect** you connect the interface of the host computer with the vision system.

The connection to the BV system is established, the run mode is stopped and the available system resources are queried.

When connecting, the operating software recognizes the available geometry elements, results, contour buffers, counters, cameras, image memory pages, the I/O hardware of the connected vision system.

After connection, the vision system is in command mode.
The following information appears in the status bar:

- video mode (memory image, live image)
- image memory page
- camera number.

5.2.5.2 Disconnect

Stop the communication between host computer and system using **Communication > Disconnect**. This command executes the test program repeatedly until it is reinitialized with **Communication > Connect**.

If the test program is not configured to send back measuring results to the host system in run mode, the RS-232 cable or the Ethernet cable between the host computer and the vision system can be disconnected as the vision system processes all data automatically.

You recognize run mode by the "Not connected" text in the status bar.

Editing a Test Program

In order to change a test program that is running cyclically in a loop, initialize communication with the vision system, modify the program and then load the modified program back into the vision system via **Communication > Transfer**.

Test Program Settings Without a vision system

You cannot edit and test interactive commands which require communication with the vision system during teach-in (e.g. all the Locate commands). To edit commands interactively, you switch from run mode to command mode using **Communication > Connect**.

5.2.5.3 Transfer

ADVICE

Communication > Transfer is only possible if the program contains commands.

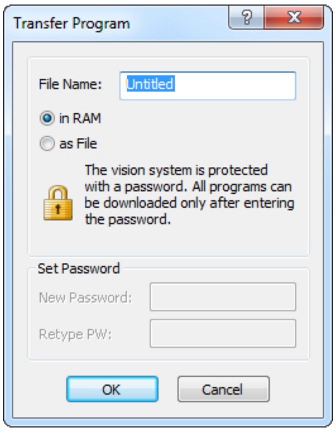


Image 16: Dialog Transfer: password protected system

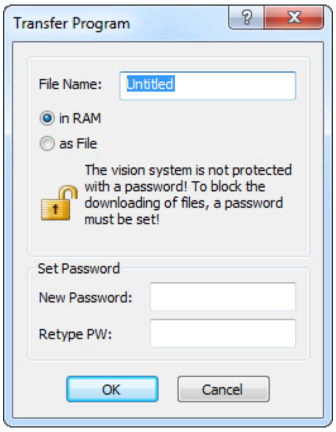


Image 17: Dialog Transfer: non password protected system

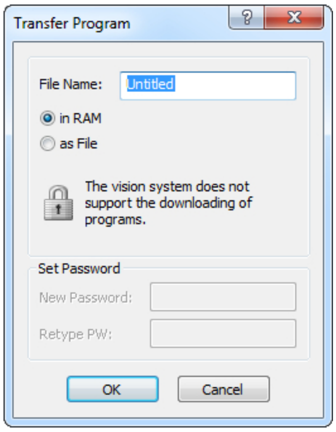


Image 18: Dialog Transfer: not downloadable system

If communication is taking place (command mode), send the test program loaded in the editor from the PC to a vision system using **Communication > Transfer**. The test program can be loaded either into the volatile memory or as a file to the vision system's non-volatile memory flash.

Input Parameters

Parameter	Description
File Name	The name under which the file should be saved in the vision system. A maximum of 8 characters are allowed with pictor M.
in RAM	The test program is loaded into the volantile memory of the vision system. With this option, you are asked whether this program should be started. The connection to the vision system is terminated when confirmed.
as File	The test program is loaded as a file in the flash of the vision system and simultaneously set as the initial program. You can execute the test program loaded with as File with **Communication > Disconnect**.
Set Password	With this option you can protect the vision system with a password to prevent the download of your test program. This function is active only in vision systems that can be protected by a password or do not have a password protection yet.

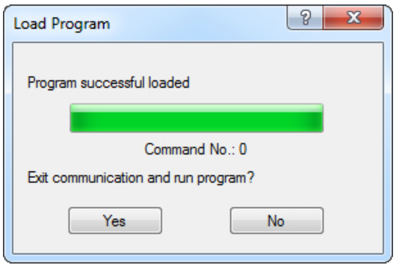


Image 19: Dialog Transfer: in RAM mode

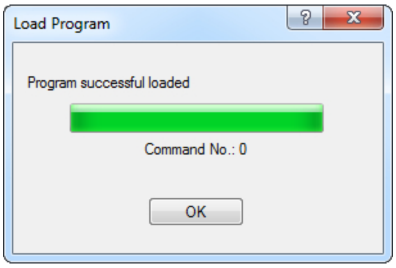
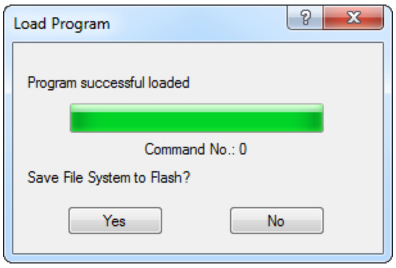


Image 20: Dialog Transfer: as File mode

ADVICE

For devices of the type pictor MxxE

After the program is succesfully loaded the query "Save File System to Flash" appears.

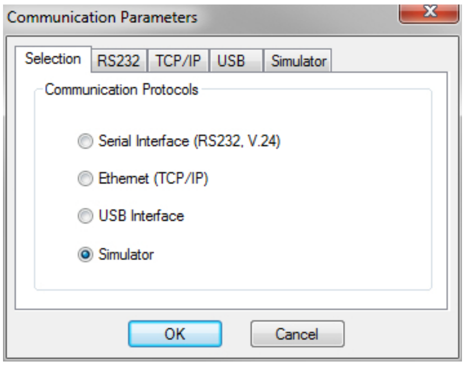


Confirm the query with [Yes] for permanent storage.
By using [No] the program is deleted in case of power interruption of the device.

5.2.5.4 Interface

ADVICE

If you are not an experienced administrator, please request the **instructions for setting up image processing systems via LAN** from the manufacturers.



Configure the communication interface between the host system and the vision system using **Communication > Interface**:

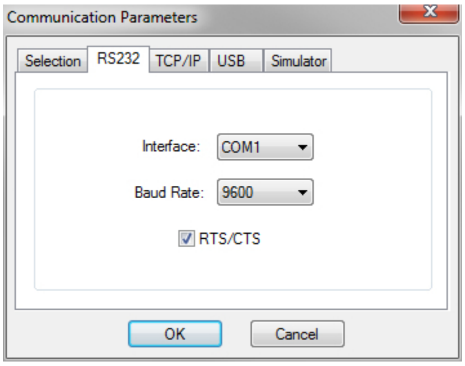
- RS232 (serial V.24)
- Ethernet (TCP/IP)
- USB
- Simulator

The parameters set in this menu are later used with **Communication > Connect** for setting-up the connection and must conform to the hardware connection set-up.

If the vision system does not respond, this could be due to a faulty interface configuration.

For Serial Connection to the vision system

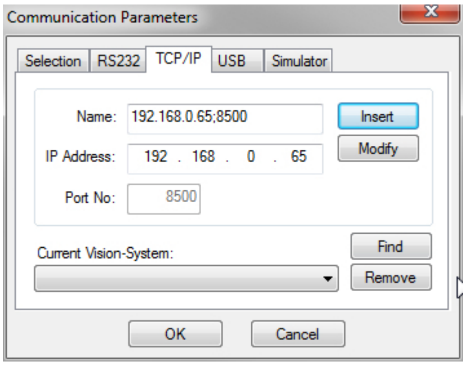
The baud rate of the serial interface is to set to 9600 after running the system. (except for pictor M24 =115200.) You can alter baud rate after the connection as required. (for example, if the program uses the command **Send Image**, to 115200.) Up to 16 serial interfaces can be supported.



For Connecting the vision system to the Ethernet

ADVICE

You can recognize active components that are connected to the Ethernet in the address list by the following appended text after the vision systems name: **- in the Net**. If a firewall is used, open UDP port 8500 on firewall. Otherwise the firewall blocks Ethernet connection.



To parameterize an Ethernet connection

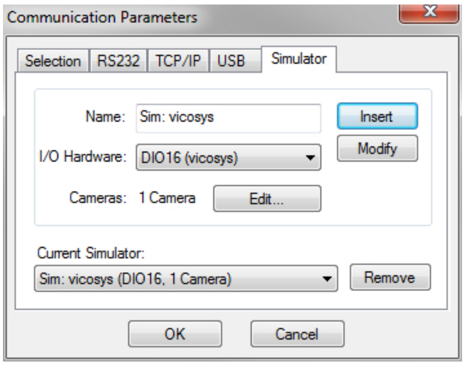
1 Connect both the host and vision system to the Ethernet. Log on to the host as Administrator.
2 For the host LAN connection, define an IP address from the sub-network of the vision system to be connected.
The IP address of the vision system can be obtained from the vision system manufacturer.
Example: For connecting a pictor M that still has the delivery settings (IP address 192.168.0.65), select a spare address for the host from the range of 192.168.0.1 to 192.168.0.254.
3 Start vcwin.
4 Select the option TCP/IP in the **Protocol Selection** tab, then switch to the **TCP/IP** tab.
5 Give a name for the connection.
6 Enter the current vision system IP address.
In order to insert the type designation and serial numbers of vision systems in the network into the address list automatically, click on the [Find] button. If possible, fetch vcwin the IP address from the DHCP server.
7 Leave the Port nr. setting unchanged at 8500. (**Only for administrators**: For remote maintenance, you can change the port in the registry via HKEY_CURRENT_USER \Software\Vision&Control\vcwin\Communication\Ethernet\Interface Adjust.)

In order to change the pictor IP address, use the **Change Pictor IP Address** command.
>> *"Change the IP Address of the Vision System", page 88*

USB Interface

ADVICE
The USB interface is not available for the current vision products yet. The implementation in vcwin is intended for future applications.

Connection to a simulator



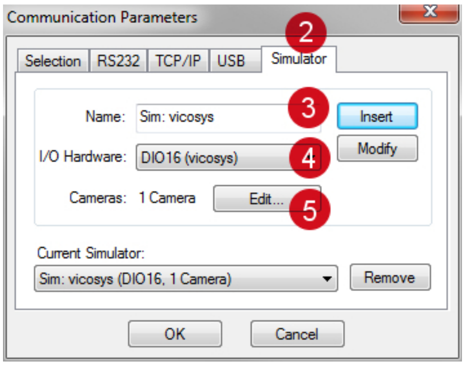
Simulators let you execute almost all commands and functions without connecting a vision system to the host computer. That allows you to test commands and check parameters and variables without needing a connected device. The simulator corresponds to a virtual vision system which imitates the technical properties.

There is a variety of cameras available (AVT-models, pictors, and customs). You can simulate up to 16 cameras with a simulator (matching the vicosys camera system). You are able to define each camera individually when using more than one camera for the simulator.

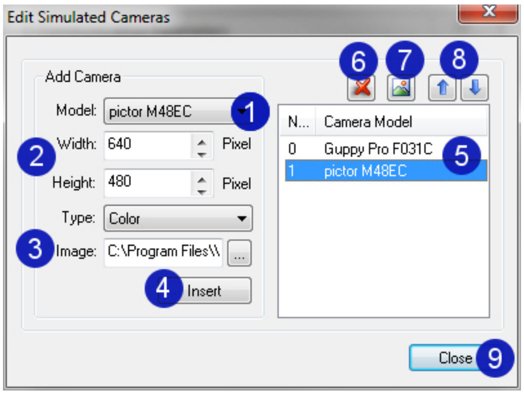
Other properties and features:

- Simulate different hardware types (PS8, DIO4/6, DIO4, DIO8, DIO16)
- Load single frames or image rows by image acquisition
- Supports command set of vicosys
- No RUN mode

To configure a simulator



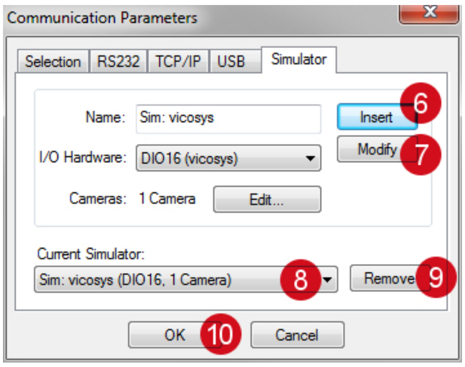
1. Open the Communication Parameters dialog via **Communication > Interface**.
2. Click on the Simulator tab (2).
3. Type in a name for the simulator (3).
4. Select the harware type (4).
5. Click on the **Edit** button (5) to add cameras to the simulator:



1. Select a camera model or use the "userdefined" option to create a camera with individual properties (1).

   ⇨ If you selected a camera model, the resolution and type of the camera are displayed.

2. If you selected "userdefined", enter the desired resolution here and select a camera type.
3. Here you can select an image (3) which will be the default picture for the image acquisition. In conjunction with the Save / Load image command and the "Use running index for file name" option (*see "Save / Load Image", page 340*) you also can use already acquired images. Choose images or image rows with the same resolution as the camera, if possible.
4. Click on the Insert button to add the camera to the simulator.
   - When you select a camera in the list, you can
     - delete the camera from the simulator (6),
     - modify the default picture (7),
     - change the order (8).
   - When you added the cameras to the simulator, close the dialog (9) and go back to the Communication Parameters dialog.



   - Click on the **Insert** button (6) to use the current simulator as the active simulator.
   - When you created more than one simulator, you can select the the active simulator via Current Simulator (8)
   - Use the **Modify** button (7) to save recent changes of the cameras or the hardware type to the current simulator.
   - To delete a simulator, click on the **Delete** button.
   - When you finished the configuration, click on the **OK** button.

     ⇨ The simulator name is shown in the status bar.

To connect to a simulator, set the interface to Simulator via Communication > Interface and click on the **Connect** button.

5.2.5.5 Offline Settings

ADVICE
This function can be used with all vision systems that can be connected via vcwin .

With **Communication > Offline Settings**, you can also edit programs when there is no vision system actually connected.

Edit programs without a vision system

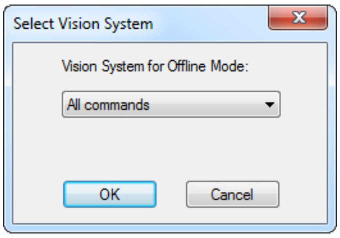
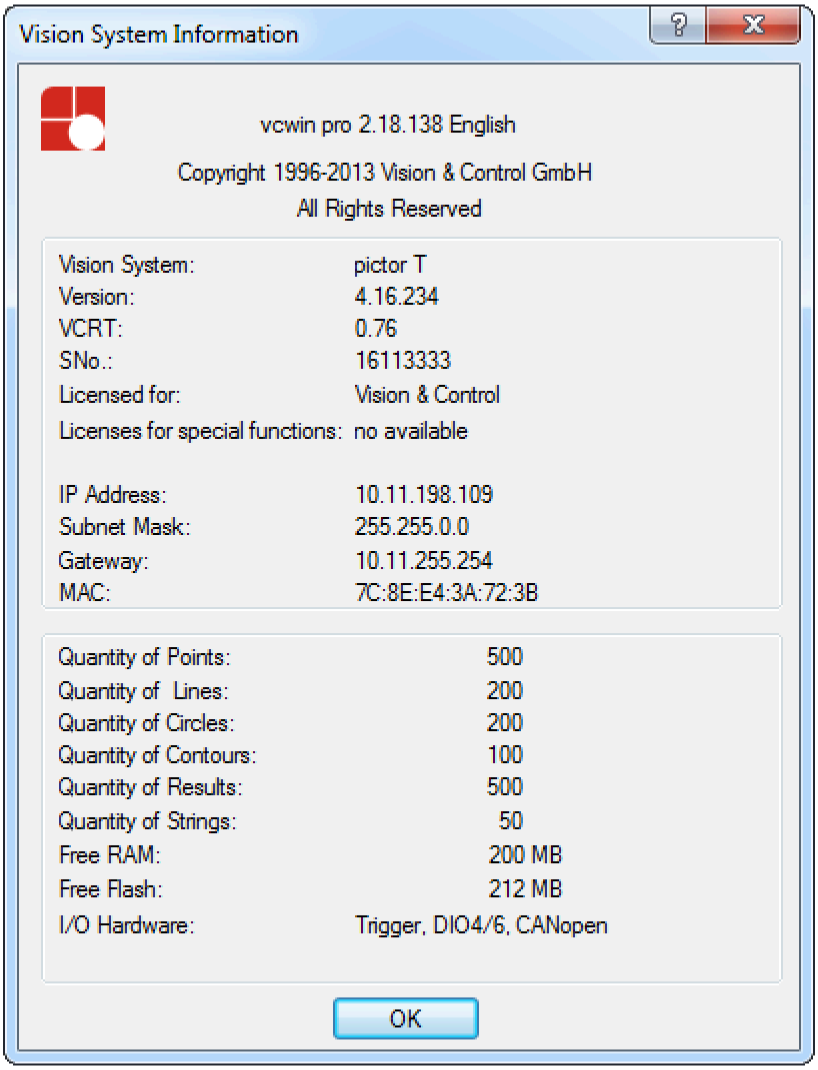


Image 21: Dialogue Select Vision System

1. Connect the vision system in vcwin. vcwin saves the vision system type internally. Note: If the same vision system is connected again, the internally saved vision system type will be erased.
2. Select **Communication > Offline Settings**, then select the vision system from the above dialog box.
3. Create the programs desired.

5.2.5.6 Vision System Information

Using **Communication > Vision System Information** to display all fundamental information about your vision system. Please note you must be connected to a vision system or a simulator.



5.2.6 Menu System Settings

The menu **System Settings** contains functions for system setting and for the communication of the vision system.

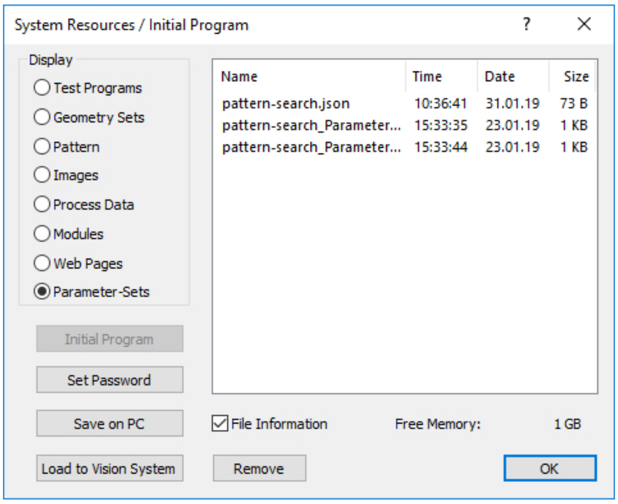
In the drop-down menu you will find:

- Options for managing test programs, geometrie sets, pattern, images and modules on the vision system
- Functions to make and restore data backups
- Options for changing the IP address of the vision system
- Options for managing date and time
- Settings for Fieldbuses, Web Server and DLC Server
- Option to restart the vision system
- Options for calibrating the cameras of the vision system, white balance and shading correction
- Option for managing licences for special functions
- Function to save the file system

5.2.6.1 System Resources / Initial Program

Overview

The Function **System Resources / Initial Program** is located in the **System Preferences** menu.



The dialog lists the files stored in the vision system, such as Test Programs, Geometry Sets, Pattern, Images, Process Data and Modules.

Usage

- Upload files to the vision system.
- Load files from the vision system to a PC.
- Deleting files from the vision system.
- Set a test program as the initial program.
- List of image files on the external disk (*see "External Storage Device", page 350*).

Group Box Display

Test Programs

Displays test programs saved on the vision system.
The Storing is performed with [Load to Vision System] or using **Communication > Transfer > Load as File** (*see "Transfer", page 73*).

Geometry Sets

Displays geometry sets saved on the vision system.
The Storing is performed with [Load to Vision System] or using **Command Selection > Control > Geometry Sets > Save** (*see "Geometry Sets", page 383*).

Pattern

Displays patterns saved on the vision system.
The Storing is performed with [Load to Vision System] or using **Command Selection >Locate > Save Pattern** (*see "Save Pattern", page 235*).

Images

Displays images saved on the vision system.
The Storing is performed with [Load to Vision System] or using **Command Selection > Control > Save / Load Image** (*see "Save / Load Image", page 340*).

Process Data

Displays process data saved on the vision system.
The Storing is performed with **Command Selection > Control > Save Process Data** (*see "Save Process Data", page 370*).

Module

Displays modules saved on the vision system.
Modules are created using the vicosys SDK.

Web pages

Displays web pages saved by the user on the vision system.
The web pages can be accessed via e.g.
http://<IP address of the vision system>/user-pages/<name of the webpage>

Parameter-Sets

Displays Parameter-Set saved by the user on the vision system.

Initial Program

With the button [Initial Program] you can set a selected program as initial program.

This button is only active with Display: **Test Programs**.

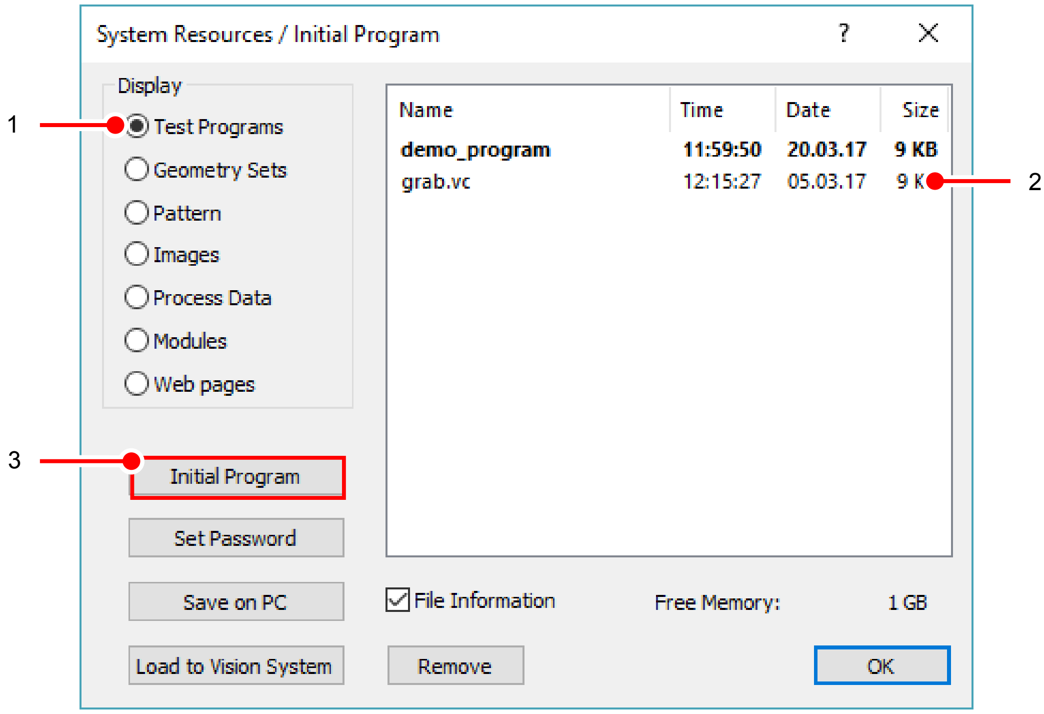


Image 22: Dialog System Resources / Initial Program > Set Initial Program

1. Activate **Test Programs**.
2. Mark the desired program.
3. Click the [Initial Program] button.
    - The current initial program is shown in bold.

ADVICE
The program listing can display a maximum of 1500 files in the system resources dialog.

Load in Image Page

With the button [Load in Image Page] you can load a selected image in an image page.

This button is only active with Display: **Images**.

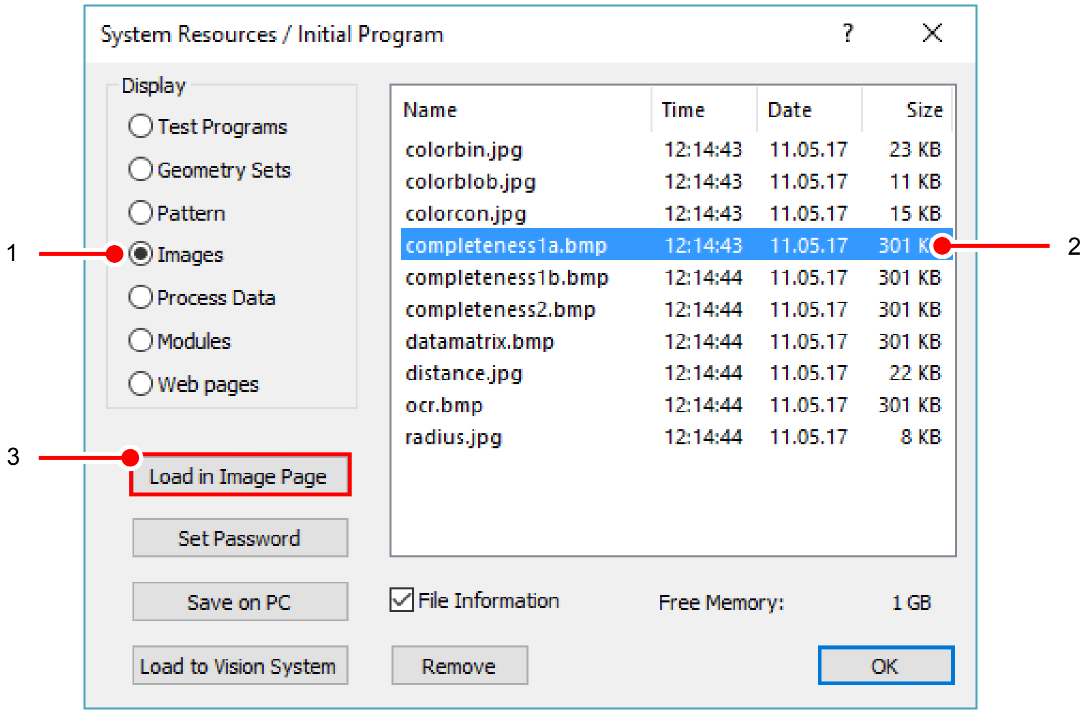


Image 23: Dialog System Resources / Initial Program > Load in Image Page

1. Activate **Images**.
2. Mark the desired image.
3. Click the [Load in Image Page] button.
    ◦ The image is load in the image page. You can see it in the Monitor Window.

Set Password

Test Programs, Geometry Sets, Patterns and Modules can be secured with a password to prevent unauthorized access.

When using the vcwin user management, only the "Programmer" is able to set or change the password.

The password will be awarded globally on the respective vision system, with which you are connected.

If the **Current Password** is grayed out, the vision system is not protected with a password.

The password will be reset by entering the old and a blank password.

The password prompt only occurs once per active connection from the operating software to the vision system.

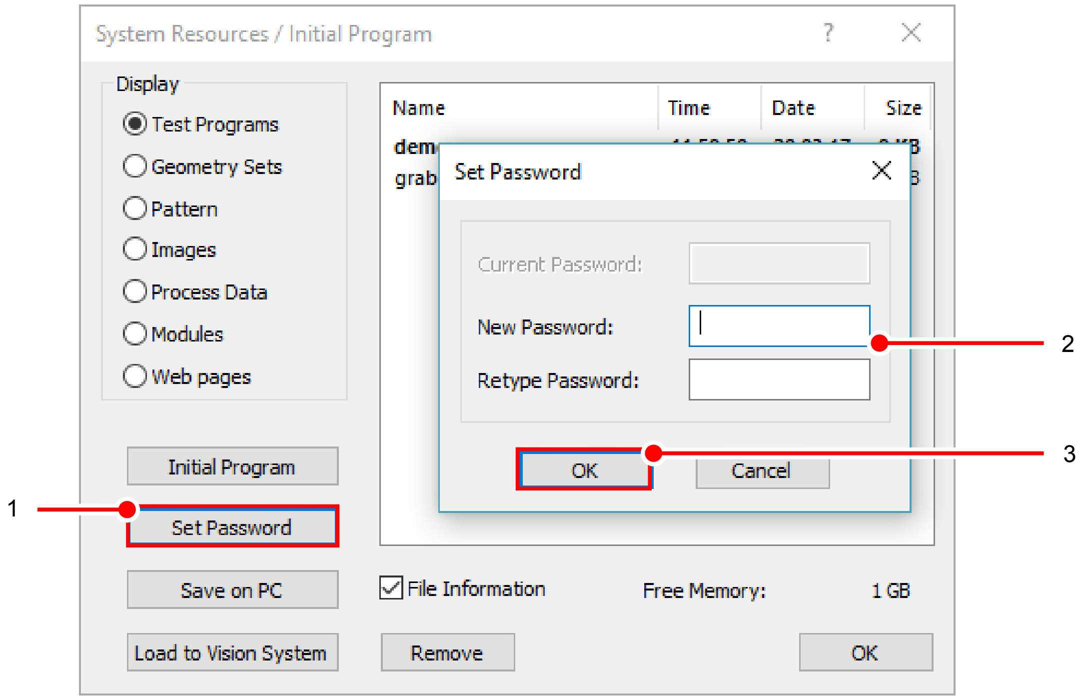


Image 24: Dialog System Resources / Initial Program > Set Password

1. Click the [Set Password] button.

   The dialog "Set Password" opens.

2. Enter a new password here. Retype the password in the next entry field.
3. Confirm with [OK].

Save on PC

With [Save on PC] you can save Programs, Geometry Sets, Patterns, Images, Process Data, Modules, Web pages and Parameter-Sets on a PC.

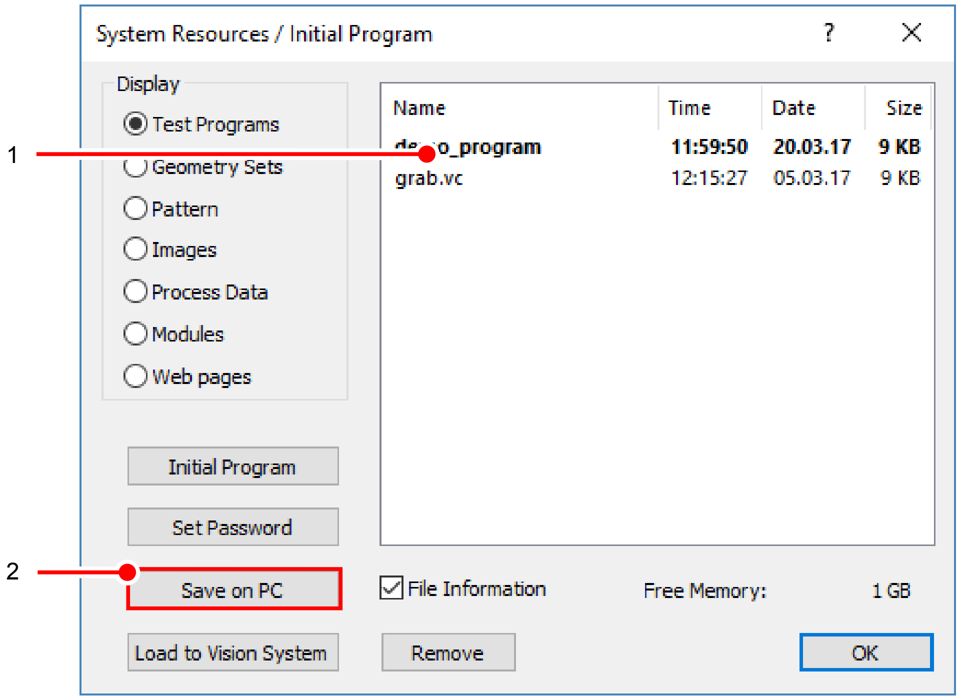


Image 25: Dialog System Resources / Initial Program > Save on PC

1. Mark the appropriate file (select multiple files by CTRL or SHIFT).
2. Click [Save on PC].

   If the vision system is password protected, a dialog will appear for entering the password (except for images and process data). Enter the password in the dialog. Choose the directory in which the corresponding files are to be saved.

ADVICE

Saves Images

For devices with firmware version 4.16.239 or older the selected JPG files are saved as compressed JPG 's. The JPG quality is set to 75% by default. You can adjust the level of compression in the section Application Settings >>Advanced<<.

Load to Vision System

With [Load to Vision System] you can load Programs, Geometry Sets, Patterns, Images, Process Data, Modules, Web pages and Parameter-Sets on a PC.

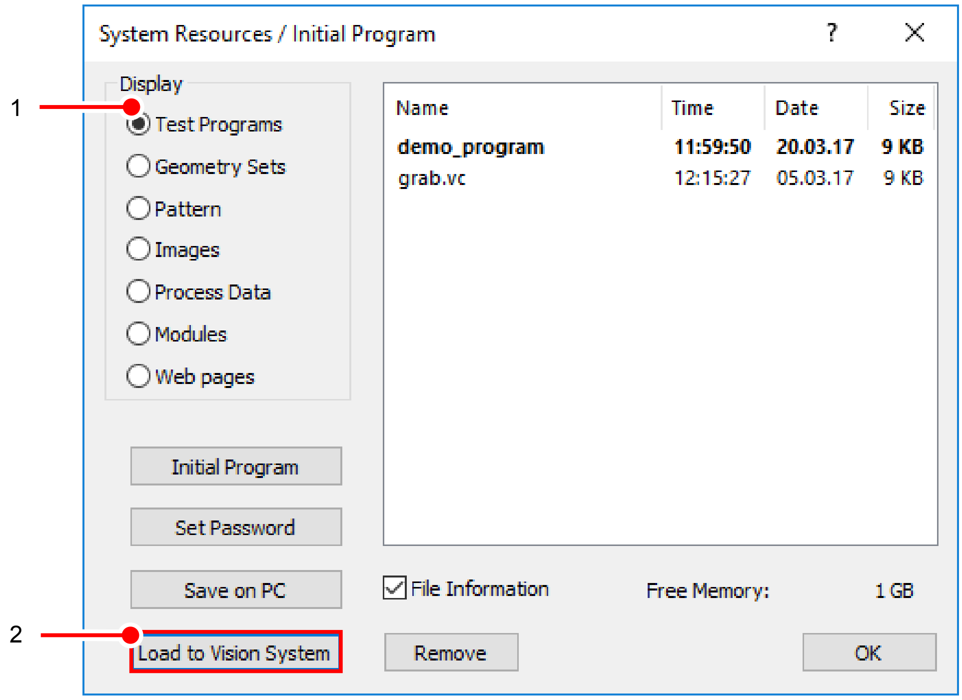


Image 26: Dialog System Resources / Initial Program > Load to Vision System

1. Activate under **Display** the appropriate file type.
2. Click [Load to Vision System].

   Navigate to the folder where the file, that you want to load to the vision system, is located. Select the appropriate file (select multiple files by pressing CTRL or SHIFT) and click the [Open] button.

Notes on file names

To ensure that your files are loaded correctly, please note the following guidelines for naming convention:

- Use only letters, numbers as well as the underscore, plus or minus.
- Do not use spaces.
- Do not use special characters such as !, §, $, %, &, #, /, or similar.
- Do not use umlauts like ä, ö, ü.

5.2.6.2 Make Data Backup

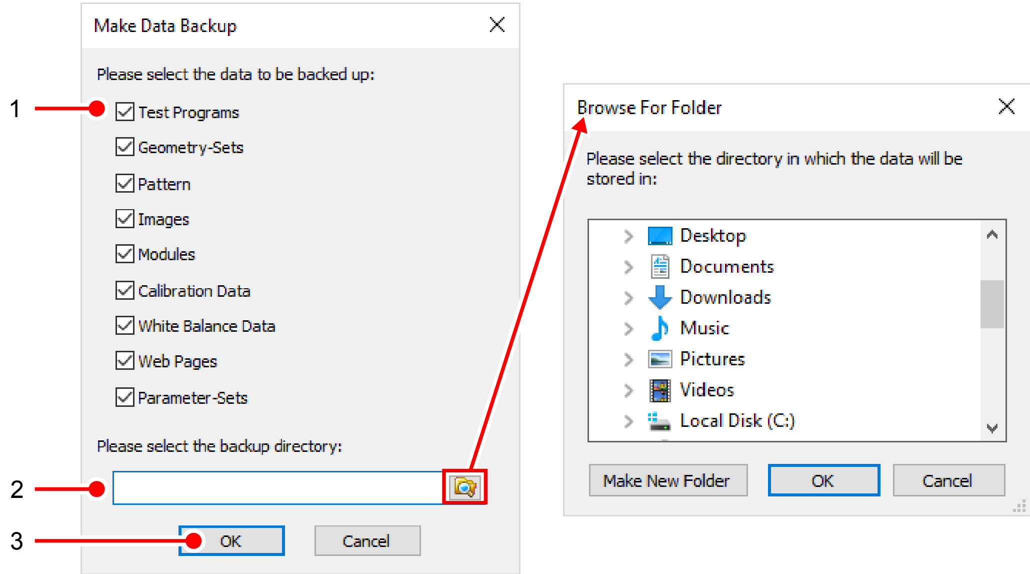
With **System Preferences > Make Data Backup** you can perform a full or partial backup of the data of your vision system.

The command includes the backup functions of the dialogs:

- **System Preferences > System Resources / Initial Program**
- **System Preferences > Calibrate Camera**
- **System Preferences > White Balance**

Brief Guide

Open the dialog with **System Preferences > Make Data Backup**.



1. Select here which data you want to backup.
2. Select here the backup folder.

   Use the ICON [Browse For Folder] to select an existing folder or create a new folder.

   The backup is created only in empty folders. It is not possible to overwrite the folders content.

   Confirm the folder selection with [OK].
3. Start the backup process with [OK].

Subdirectories are created in the selected directory and the data are stored there accordingly.

When finished, a window will appear with a summary of the backup. Confirm with [OK] to complete the process.

5.2.6.3 Restore Data Backup

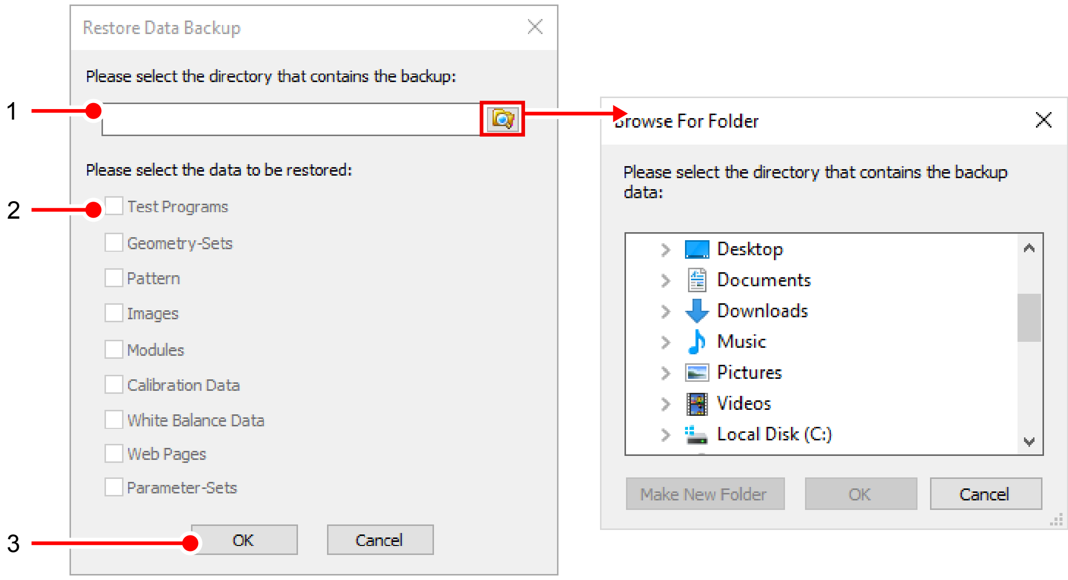
With **System Preferences > Restore Data Backup** you can perform a full or partial restoring of the data of your vision system.

The command includes the import functions of the dialogs:

- **System Preferences > System Resources / Initial Program**
- **System Preferences > Calibrate Camera**
- **System Preferences > White Balance**

Brief Guide

Open the dialog with **System Preferences > Restore Data Backup.**



1. Select the folder containing the backup data.

   Use the ICON [Browse For Folder] to select an existing folder or create a new folder.

   Confirm the folder selection with [OK].
2. Once you have selected the folder, the available data are displayed. Select which data you want to restore.
3. Start the restore procedure with [OK].

At the start of the restoring, the data are transferred to your vision system and stored there. If files already exist on the vision system, you will be asked if the files should be overwritten.

After completion of the process a window will appear with a summary of the restore process. Confirm with [OK] to complete the process.

5.2.6.4 Change the IP Address of the Vision System

With **System Preferences > Change IP Address of the Vision System** you configure the IP address of the vision system.

Brief Guide

1. Open the dialogue with **System Preferences > Change the IP Address of the Vision System.**
2. Confirm the following warning ("Do you really want to proceed?") with [Yes].

   The dialogue for changing the IP address is opened.

Parameter	Description
Current IP Address	Specifies the current parameters of the vision system.
Current Mode	Specifies the current mode.
New Mode	Specifies the mode to be used.
New static IP Address	Specifies the new static IP address (if the new mode requires a static address).

3. Under *New Mode*, select the address mode to be used. Depending on the vision system, the following variants are available:

Static IP Address

- The vision system uses only the static IP address specified under *New static IP Address*.

DHCP and static IP Address

- The vision system tries to obtain an IP address from a DHCP server. If this fails, the static IP address, specified under *New static IP Address* is used.

DHCP and Link-Local (only pictor N)

- The vision system tries to obtain an IP address from a DHCP server. If this fails, the vision system assigns itself an IP address from the "link local" address space.

4. If required, enter the new parameters under **New IP Address** (ask your administrator if necessary.):
   - IP address
   - Subnet mask
   - Gateway
5. Click the [OK] button.
6. Confirm the message ("The new IP address will be used when the vision system is restarted.").
7. Restart the vision system.
   - Terminate the communication of the operating software.
   - Disconnect the vision system from the power supply.
   - Connect the vision system to the power supply.

ADVICE

The IP address in use is displayed in the boot screen when the vision system is starting.

5.2.6.5 Date and Time of Vision System

Use **System Preferences > Date and Time of Vision System** to configure the date and the time of the vision system.

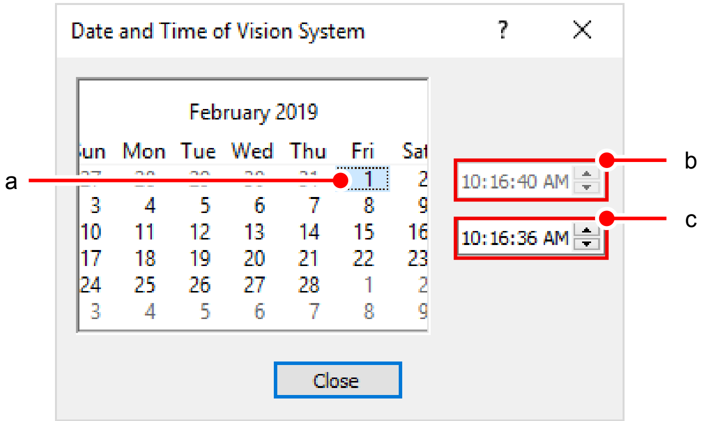
ADVICE

According to the model the vision systems also have a battery or a buffer-capacitor. In case of power loss, the date and time are backed up for a certain period (at least a few days, depending on device model). In case of a long interruption of the power supply, date and time must be reset.

Set date and time

1. Set up the connection to the vision system.
2. Choose the subject **Date and Time of Vision System** in the **System Preferences** menu.

   The date set on the vision system is displayed in the calendar (a) and the set time in the field (b).



3. In the calendar, select the desired date (a) and in the field (c) the new desired time. These are immediately transferred to the vision system.
4. Close the dialog by clicking the [Close] button.

Evaluate date and time

To evaluate the date and time of the vision system in programs, use the command **Save System Time** *see "Save System Time", page 385*.

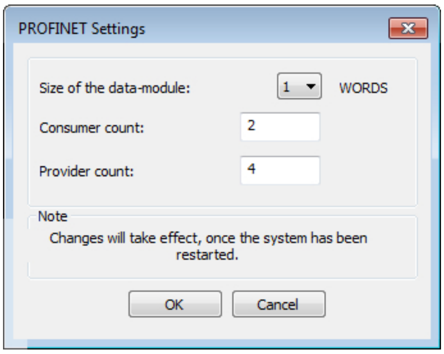
5.2.6.6 Fieldbus Settings

Here you can make changes to the fieldbus settings. Depending on the supported interface, the dialog for PROFINET, sercos III or CANopen is displayed.

ADVICE

The settings will be activated after restart the vision system.

PROFINET

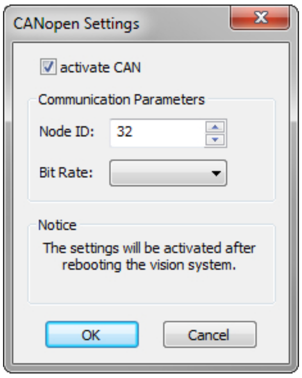


With **System Preferences > Fieldbus Settings** you can configure the settings of the fieldbus PROFINET.

You can make the following settings in the dialog for PROFINET:

Parameter	Description
Size of the Data Modules	Select here the size of the data modules in WORDs.
Number of Consumers	Enter the number of consumers (input modules) here.
Number of Providers	Enter the number of providers (output modules) here.

CANopen



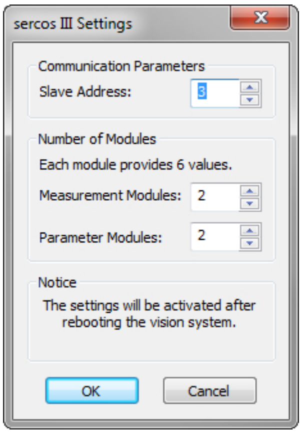
With **System Preferences > Fieldbus Settings** you can configure the settings of the fieldbus CAN.

Activate the check box if you want to use the fieldbus CAN. Deactivate the check box if you do not want to use the CAN fieldbus.

You can make the following settings in the dialog for CAN:

Parameter	Description
Node-ID	Set the node address (Note-ID) here to uniquely identify the device in the CAN network.
Bit Rate	Specify the transfer rate for the device here.

sercos III



With **System Preferences > Fieldbus Settings** you can configure the settings of the fieldbus sercos III.

You can make the following settings in the dialog for sercos III:

Parameter	Description
Slave Address	Set the slave address here to uniquely identify the device in the sercos III network.
Measurement Modules	Define the number of measurement modules (outputs) here. Each measurement module provides 6 outputs.
Parameter Modules	Define the number of parameter modules (inputs) here. Each parameter module provides 6 inputs.

5.2.6.7 Web Server Settings

You can make changes to the Web server settings here.

ADVICE
The settings will be activated after rebooting the vision system.

ADVICE
An activated web server may affect the execution time of programs and the number of available image memory pages.

With **System Preferences > Web Server Settings** you can configure the settings of the web server. Activate the checkbox if you want to use the web server. Deactivate the checkbox if you do not want to use the web server.

Server Settings

Parameter	Description
Port	Port on which the web server is accessible. Recommended: 80 (default) Not allowed: 20, 21, 22, 23, 8500

Memory Allocation

In this section you can allocate memory space for the image buffer.

ADVICE
Memory allocation can reduce the number of image memory pages in the vision system.

Parameter	Description
Image memory of the Vision System	Size of the image memory available on the vision system
Allocated memory for Web Server	Amount of memory allocated to the web server as buffer.

5.2.6.8 DLC-Server settings

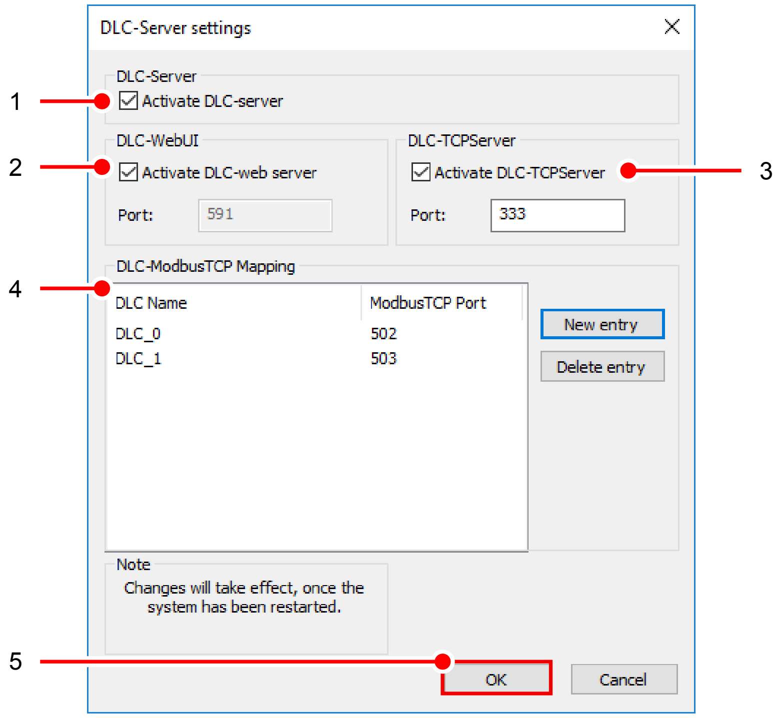
ADVICE
Changes to the settings are only applied after rebooting the vision system.

With menu **System Preferences > DLC-Server settings** you can change the settings of the DLC-Server.

The following options allow you to access a DLC-Server:

- WebUI
- DLC-API via TCP-Server
- ModbusTCP

Settings



1. Activate the checkbox if you want to use the DLC-Server. Deactivate the checkbox if you do not want to use the DLC-Server.
2. Activate the check box if you want to use the DLC-WebUI.

   The port for the WebUI is permanently set to 591 by the vision system (HTTP Alternate).
3. Activate the checkbox to start the DLC-TCPServer.

Parameter	Description
Port	Port on which the TCPServer can be reached. recommended: 333 (default) not allowed: 20, 21, 22, 23, 80, 591, 8500

4. Configure a DLC-ModbusTCP mapping to access a DLC via ModbusTCP.

   Click the [New Entry] button to add a new Mapping entry. To remove an entry, select the appropriate line and click [Delete Entry].

Parameter	Description
DLC Name	Name of the DLC (configurable via WebUI)
ModbusTCP-Port	ModbusTCP Port assigned to the DLC. not allowed: 20, 21, 22, 23, 80, 591, 8500, (TCP-Server Port)

5. Confirm with [OK] to complete the configuration.

5.2.6.9 Restart the vision system

With **System Preferences > Restart the Vision System** you trigger the restart of the vision system. Restarting is necessary in some cases (e.g. changed system preferences).

5.2.6.10 Calibrate Camera

ADVICE

Calibrate the vision system if you use it for measuring.

ADVICE

Re-calibration

You should re-calibrate the camera in the following cases:

- in case of mechanical, optical or electrical changes in the test set
- after changing camera
- changing the unit of measuring
- after maintenance and repair work

A re-calibration is recommended at regular intervals for preventive maintenance.

With **System Preferences > Calibrate Camera** you can calibrate vision systems or the cameras of the vicosys interactively without a test program. Absolute calibration is used for determining the calibration factors.

The calibration factor – the scale between image and world coordinates – is determined through a calibration command. Calibration is always active for the currently selected vision system or camera.

Brief Guide

Setting Scene

1. Select the camera that is to be calibrated.
2. Activate the checkbox **Live Image**. The Image is displayed on the control monitor or in the monitor window.
3. Position calibration phantom in the center of the image. Ideally it should cover 2/3 of the image field.
4. Enhance image sharpness and lighting.

Grab Image

5. Deactivate the checkbox **Live Image**.
6. Push the button [Grab Image].

Perform calibration

7. Teach-in the calibration window
   - To switch to editing in the video image, double click on the right mouse button.
   - Set the size and position of the window, *see "Teaching in Test, Detection and Search windows", page 26*.
8. Specify calibration phantom
   - Chose the calibration object (from light to dark/ from dark to light).
   - Chose the calibration shape of the calibration phantom (rectangle or circle).
   - Enter the exact (already known) dimensions of the calibration phantom.
9. Detection Algorithm
   - **Gray Value with Subpixel**
   - **Gradient**
10. Specify calibration direction

- Choose whether to calibrate in the x- or y- direction, or in both directions.

11. Start Calibration
    - Click the [Calibrate] button.
    - Make a visual check on the control monitor or in the monitor window that the calibration phantom is detected precisely.

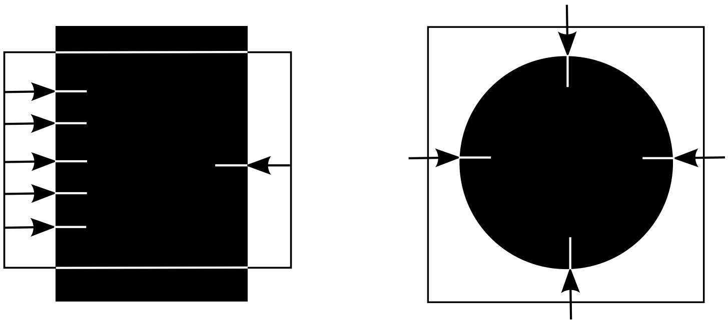


Image 27: Calibration with rectangle object (left side) and circle object (right side)

- Perform the calibration for all cameras.

Finish Calibration

12. To save the calibration data permanently on the vision system click the button [Save Settings].

Input Parameters Grab Image

Grab the image with which the calibration is to be made using the [Grab Image] button.

Parameter	Description
Camera	Number of the vision system's camera to be calibrated.
Live Image	Option for displaying the live image.

Input Parameters Calibration Object

Parameter	Description
Dark/Light	States whether the calibration body is brighter or darker than the surroundings. Dark: edge transition for calibrating from light to dark Light: edge transition for calibrating from dark to light
Rectangle/ Circle	States the shape of the calibration phantom.
Width/Height	Reference dimensions in µm of a rectangular calibration phantom.
Diameter	Reference dimension in µm of a round calibration phantom.

Input Parameters Detection Algorithm

Parameter	Description
Grey Value with Subpixel	A particularly precise algorithm that should be applied at nearly ideal specimens and test pieces. Parameter: Threshold, Width, Noise Filter, Diffusion
Gradient	A very stable and robust method that can be used in non-optimum conditions. Parameter: Threshold, Width, Noise Filter, Diffusion

Input Parameters Calibrate

Parameter	Description
X Direction	Only calibrate in the horizontal direction.
Y Direction	Only calibrate in the vertical direction.
Both directions	Calibration in the horizontal and vertical directions.
Calibrate	With the button [Calibrate] the procedure is started and determines the calibration factors.

ADVICE

Because of aberrations, it is always advisable to select both directions for calibration.

Export Calibration Data

With the button [Save on PC] you save the calibration data of all cameras to a PC.

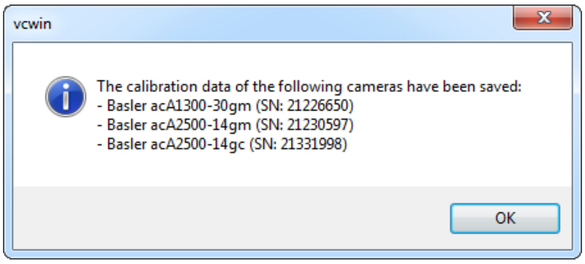


Image 28: Message: calibration data stored

The calibration data are stored in a file with the extension *.vcc. The calibration data are correlated based on the data of the manufacturer, model name and serial number.

Import Calibration Data

With the button [Load to Vision System] you upload all stored calibration data on the initialised vision system.

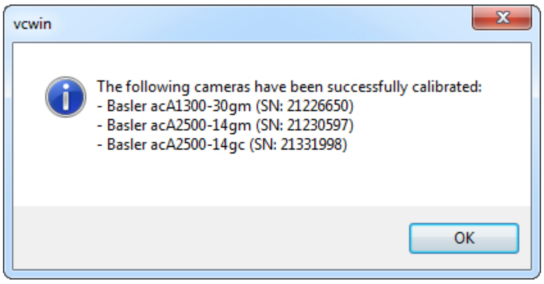


Image 29: Message: calibration data loaded

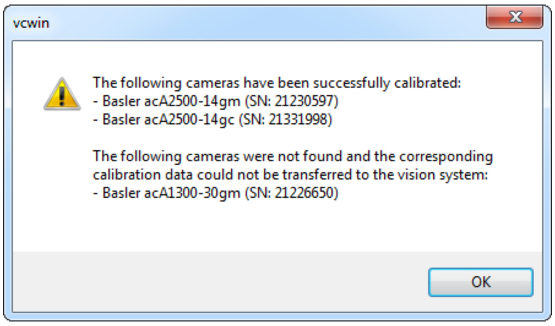


Image 30: Message: calibration data loaded/ not loaded

Save Settings

With the button [Save Settings] you save all the calibration data permanently on your vision system. Equivalently, you can choose **Save Settings** in the **System Preferences** menu.

ADVICE

The function Save Settings stores in addition to the calibration data also the demo mode and video mode.

5.2.6.11 Display Calibration Data

Overview

The dialog **Display Calibration Data** is located in the **System Preferences** menu.

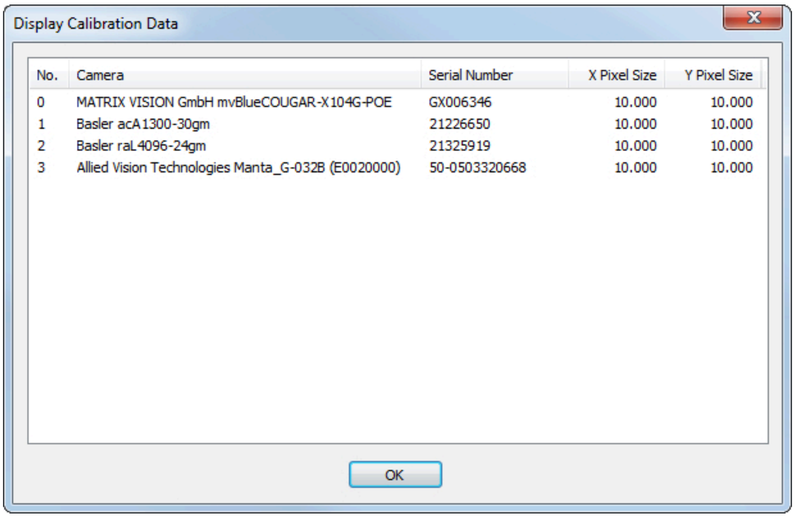


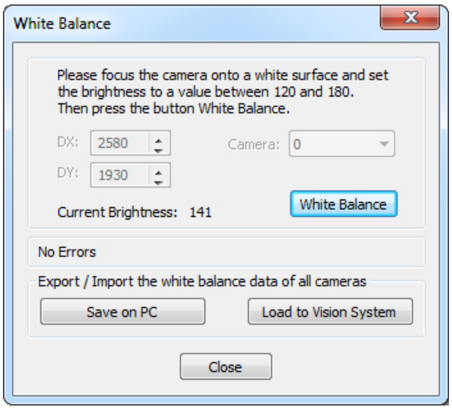
Image 31: Dialog Display Calibration Data

The dialog reads the current calibration data of the vision system and displays them for each camera.

5.2.6.12 White Balance

ADVICE

The white balance is a compensation method for true color reproduction. It should be done absolutely by color cameras before use.



Use **System Preferences > White Balance** to carry out the white balancing process. In this case, an adjustment of the gain of individual RGB color channels is performed. A correctly calibrated color camera yields identical values for R, G, and B when viewing a gray surface.

Perform White Balance

1. Select the camera from the drop-down list.
2. Direct the camera onto a white surface.
3. Adjust the current brightness of the image with the lens iris of the camera between 120 and 180.

4. Click on the button [White Balance] when in the lower part of the dialog *No Errors* is displayed.
5. The white balance will be carried out. If the message *White Balance OK* appears in the lower part of the dialog, the white balance is completed.
6. The white balance data are saved immediately.

Parameter White Balance

Parameter	Description
DX:/DY:	Specifies the size of the test area. The test area is always centered in the image. Depending on the vision system, the size of the area can be changed.
Camera:	Number of the camera that is provided for white balance.
Current Bright-ness:	Indicating the current brightness of the image. For white balance, the brightness is ideally in the range of 120 to 180.
White Balance	With the button [white balance] the process is started and the factors are determined.

Export White Balance Data

With the button [Save on PC] you save the white balance data of all cameras to your PC.

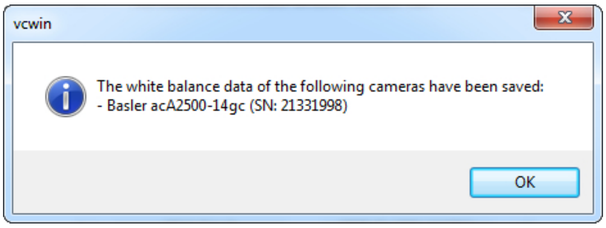


Image 32: Message: white balance data stored

The white balance data are stored in a file with the extension *.vcw. The white balance data are allocated based on the data of the manufacturer, model name and serial number.

Import White Balance Data

With the button [Load to Vision System] you upload all stored white balance data on the initialized vision system.

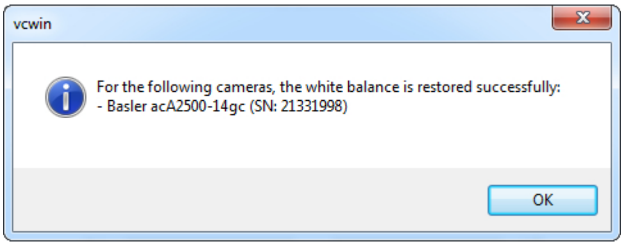
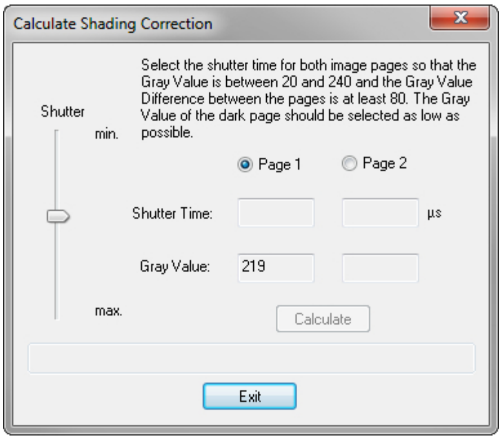


Image 33: Message: white balance data loaded

The white balance data will be associate to the vision system or the cameras of the vision system based on the data by manufacturer, model name and serial number.

5.2.6.13 Shading Correction



ADVICE

This command is specifically for line pictor M41EL.

Shading correction is a tool with which you can correct systematic brightness errors in the test image. These errors can occur due to vignetting or inhomogeneous lighting. Shading correction uses two differently lit images to determine the greyscale value differences which occur as a result of these brightness errors and writes them to a shading matrix. This matrix is then applied to all images captured with the vision system and offsets the brightness information against the correction values to ensure homogeneously illuminated test images.

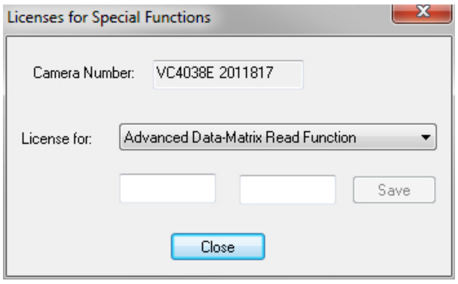
ADVICE

Shading correction should be carried out immediately after installing the camera and lighting equipment.

Quick-start guide

1. Connect to the vision system.
2. Switch the video mode to Live Image in the Video Control Panel.
3. Open the monitor window and switch to full screen to best evaluate the displayed image. Check to ensure that the image is sufficiently homogeneously illuminated.
4. Click **Exit** to exit the live image. The video mode is automatically switched to Memory Image.
5. Open the command **System Preferences > Shading Correction.**
6. Set the shutter for the two image sides using the controller on the left-hand side. As specified in the dialog, ensure that the resulting greyscale values are both between 20 and 240 and that the greyscale value difference between side 1 and side 2 is at least 80.
   ⇨ As soon as you change the shutter setting, an image is captured automatically and the average greyscale value displayed.
7. Once you have set the appropriate shutter values, click the button [Calculate].
8. Then click the button [Exit] to save the settings.
   ⇨ The calculated shading matrix is saved in the FLASH of the device and then applied for correction with each image capture.

5.2.6.14 Licenses for Special Functions



This command allows you to enter and activate your licenses for the functions "Edge Based Object Search" and "Advanced Data Matrix Read Function".

5.2.6.15 Save Settings

Save the current system modes (demo mode, video mode, shutter speed) and the calibration factors to the flash using **System Preferences > Save Settings**.

For vision systems of the type pictor MxxE data is not saved on the flash but in the RAM disk. You should ultimately call **System Preferences > Save File System in Flash** to transfer the data in the flash to save them permanently.

5.2.6.16 Save File System in Flash

ADVICE
This function is only required for vision systems of the type pictor MxxE.

With **System Preferences > Save File System in Flash** you save the contents of the RAM disk to the nonvolatile memory / Flash of the vision system.

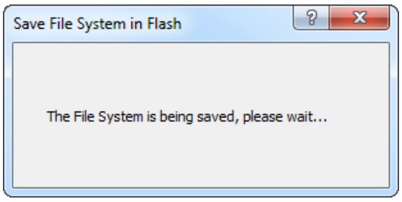


Image 34: Dialog: Save File System in Flash (1)

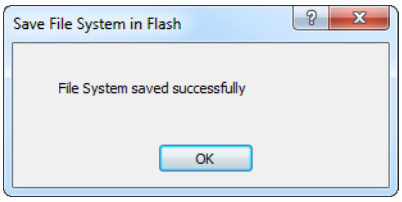


Image 35: Dialog: Save File System in Flash (2)

5.2.7 The Options Menu

With the **Options** menu you can change the appearance of the user interface.There is also the user management.

In the drop-down menu you will find:

- Settings to display commands
- Adjustments to the surface such as language, font and colour, settings for timeouts, breaks and display time and the quality of images with the JPEG option
- Settings for user management

5.2.7.1 Full Display

With this option, you can switch between the full display of the commands containing the complete program text and a short display which shows the command number, the command name and a short version of the parameter list.

```
No.    Label  Command
*****         Last Change at vcwin 2.33.0
00000         Standard Image Acquisition with camera 0
00001         Locate Point 0; Max. Gradient Algorithm; Search Arrow:
              [625, 772/554, 762]
00002         Locate Point 1; Max. Gradient Algorithm; Search Arrow:
              [1150, 205/1250, 305]
00003         Distance between Point 0 and Point 1 (directly);
              Result 1
```

example program in option Short Display

```
No.    Label  Command
*****         Last Change at vcwin 2.33.0
00000         Standard Image Acquisition with camera 0
00001         Determine Point 0 (World); Max. Gradient [10, 1, 2];
              Light > Dark
              Search Arrow: [625, 772/554, 762]
00002         Determine Point 1 (World); Max. Gradient [10, 1, 2];
              Light > Dark
              Search Arrow: [1150, 205/1250, 305]
00003         Distance between Point 0 and Point 1 (directly)
              Result: 1; Nominal 0 (+10/-10)
```

example program in option Full Display

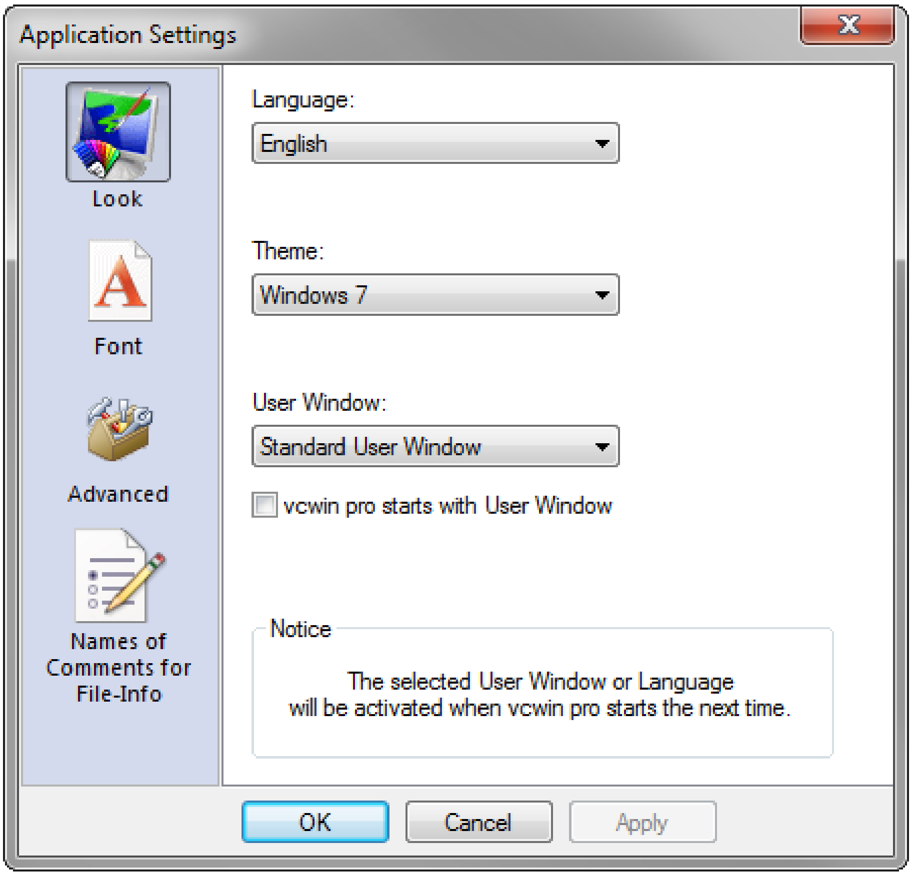
5.2.7.2 Display Highlight Colors

This disables or enables the presentation of the commands background color in the command window (*see "Highlight with Color", page 55*). This option is active by default.

This setting is a setting of the operating software and is not stored within the program.

5.2.7.3 Application Settings

Application Settings



Page Look

Language settings:

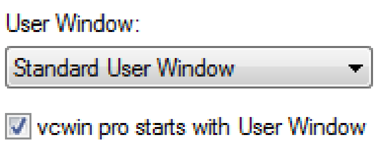
Here you can define the language of the user interface. You have to restart the operating software to activate this change.

Theme settings:

Here you can define the design of the user interface. The design is applied immediately after clicking the "OK" or "Apply" button.

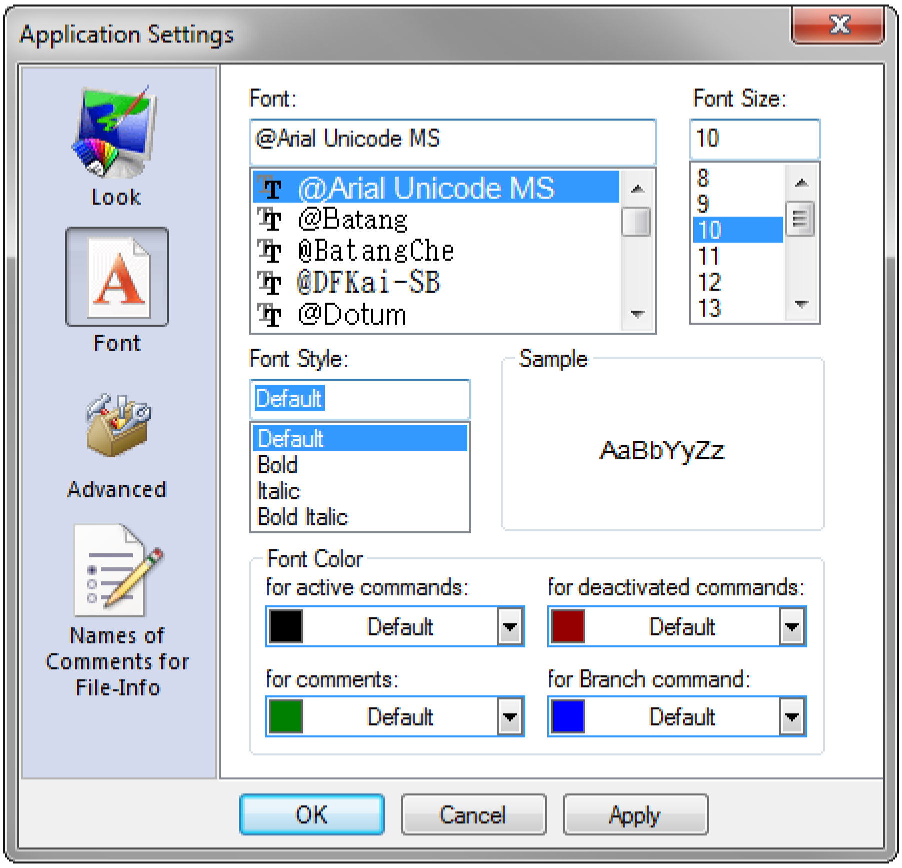
User Window settings:

This is used to select which User Window appears in the operating software. You have to restart the operating software to activate this change.



Activate the checkbox "vcwin starts with User Window" if you want to start the operating software with the choosen User Window. Deactivate the checkbox if the operating software should be started with the programming window.

Page Font



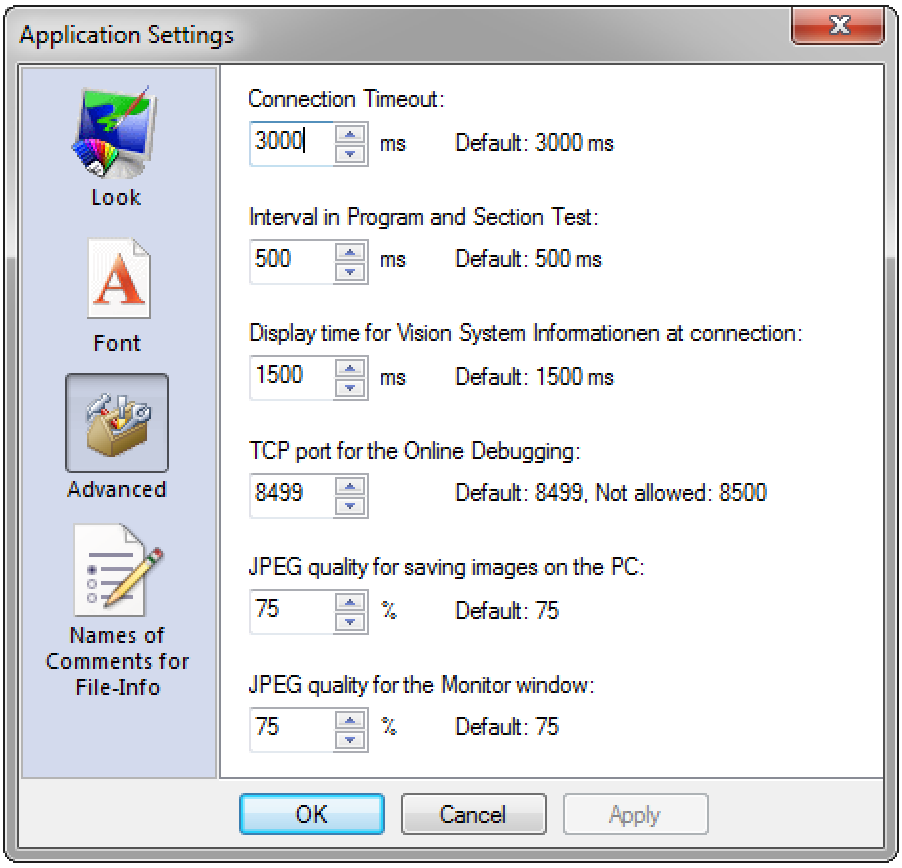
Font settings:

Here you can define the font, the font size and the font style for the text display of the check programms.

Font colour settings:

Here you can define colours for active commands, deactivated commands, comments and Branch commands.

Page Advanced



Connection Timeout:

Here you can set the time, after witch the vision system needs to answer at the latest. For slower connections (like VPN) this time should be raised.

Interval in Program and Section Test:

Changes the length of the interval between the commands in a program- or section test.

Decrease the value to used to speed up the programm- or section test.

Display Time for Vision System Information at Connection:

Changes the display time of the dialog "Vision System Information". Those Informations can also be viewed under **Communication > Vision System Information**.

TCP port for Online Debugging:

Here you can change the port used for debugging (Port 8499 is set as standard for debbuging), if you want to use this port in a different manner (e.g. datatransfer).

JPEG quality for saving images on the PC:

Here you can set the quality of a JPEG image that is to be saved on the PC.

This setting affects:

- Menu: Utilities > Receive Image from Vision System, *see "Receive Image from Vision System", page 70*
- Menu: System Preferences > System Ressources/Initial Program > Images > Save on PC, *see "System Resources / Initial Program", page 81*

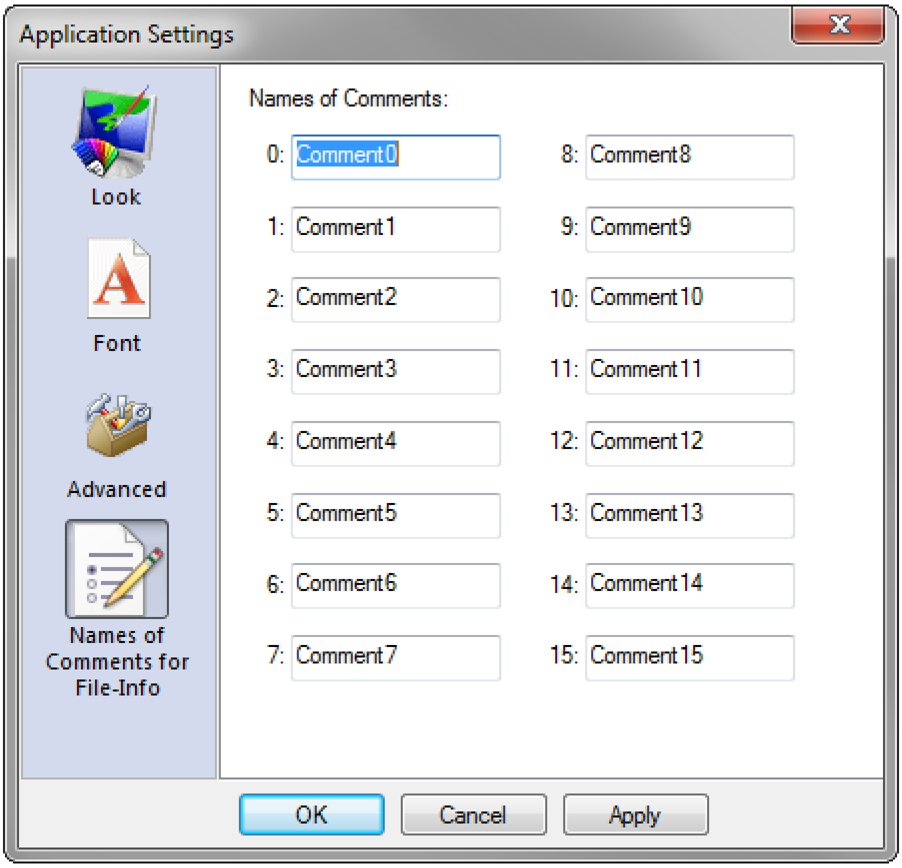
JPEG quality for the Monitor Window:

Here you can set the quality of the JPEG images that are transferred from the vision system to the Monitor Window (even at Live Image). This setting affects also the saving of the images directly from the Monitor Window.

This setting affects:

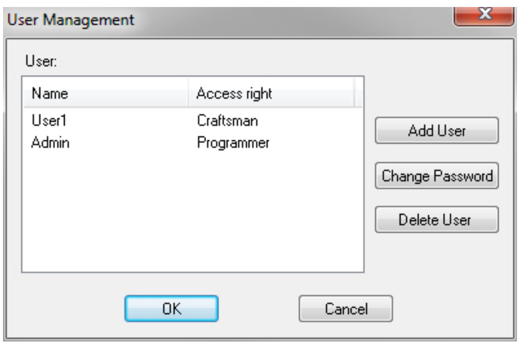
- Monitor Window / Save Image, *see "Docking Window Monitor Window", page 118*
- Monitor Window / Live Image, *see "Docking Window Monitor Window", page 118*
- Display of Monitor Window, *see "Docking Window Monitor Window", page 118*

Page Names of Comments



Here you can change the names of the commands, that are used under **File > Info** "Table".

5.2.7.4 User Management



Here, you can assign the users and their user rights as **Craftsman**, **Foreman**, or **Programmer**.

User group	Rights in vcwin
Programmer	Access to all functions without restrictions.
Foreman	• In the Programming Window, the right to change commands enabled for foreman in **Edit > Command Selection**. • Change settings in the User Window
Craftsman	Operate programs in the User Window.

To create a new user

1. Click on **Add User**.
2. Enter the user name.
3. Select the user right.
4. Enter the password in the **Password** field, then enter the password again in the **Confirm Password** field.

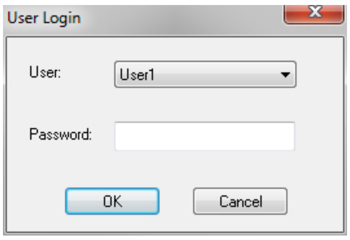
To delete a user

1. Mark the user.
2. Click on **Delete User**.

ADVICE

If you upgrade your operating software from version 2.10 or less to a higher version (2.11 or above), you have to create again all users you have created via **Options > User Management** in the older version.

5.2.7.5 User Login



Here, you can switch the user during the programming, e.g. to test the program using Foreman or Craftsman rights.

To log in

1. Select the user.
2. Enter the user password.

5.2.8 The Help Menu

The **Help** menu contains the menu item to invoke the help and the program information.

In the drop-down menu you will find:

- the online-help (these instructions of use)
- Informations about the user interface and the processing system

5.2.8.1 Help Content

vcwin Online Help

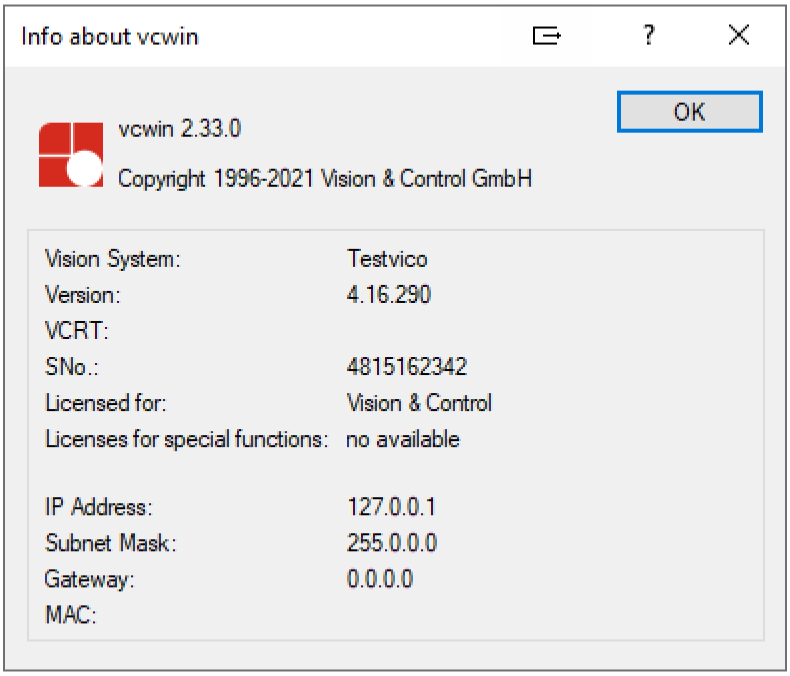
Access the help with **vcwin > Help > Help Content**.

To start a new query on a particular subject use the tabs [Contents] or [Search] located in the help windows menu bar.

The help is context sensitive. This means that you can call help about a dialog box directly by pressing F1.

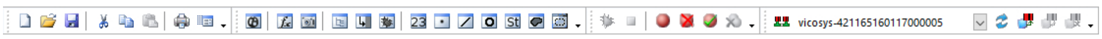
Help texts can be printed, copied to the clipboard or marked with a bookmark.

5.2.8.2 Info about vcwin



Info about vcwin displays information about the program, such as version number, manufacturer and license etc.

5.3 Toolbars



The toolbars are located in the upper left corner of the programming window, below the menu bar.

The toolbars consists of buttons, symbolized by icons (thumbnails), enabling quick access to menu items and individual functions. They include a selection of frequently used commands.

In the default setting the toolbars contains of following toolbars: **Default**, **Docking Windows**, **Debugging** and **Communication**. In the default view, the predefined toolbars **Commands:**

Image, **Commands: Locate**, **Commands: Definition**, **Commands: Control**, **Commands: Evaluation** and **Commands: Check** are not displayed.

Wie Sie die Symbolleisten ihren Bedürfnissen anpassen können erfahren Sie im Abschnitt *"Customize Toolbars", page 23*

In the section below the toolbars are listed, and each of the buttons are explained.

5.3.1 Toolbar - Default

The **Default** toolbar, provides the following functions from the menu **File** as buttons:

Icon	Designation	Description
	New	Creates a new program.
	Open	Opens an existing program.
	Save	Saves the current program.
	Cut	Cuts the selection and transfers it to the clipboard.
	Copy	Copies the selection to the clipboard.
	Past	Inserts the contents of the clipboard.
	Print	Prints the current program.
	About	Displays program information, version number and copyright.

5.3.2 Toolbar - Docking Windows

The **Docking Windows** toolbar, provides all docking windows of the menu **View** as buttons.

Icon	Designation	Description
	Monitor Window	Shows/Hides the Monitor Window.
	Command Selection	Shows/Hides the Command Selection.
	Video Control Panel	Shows/Hides the Video Control Panel.
	Project Tree	Shows/Hides the Project Tree.
	Labels Tree	Shows/Hides the Labels Tree.
	Debug History	Shows/Hides the Debug History.
	Result List	Shows/Hides the Result List.
	Point List	Shows/Hides the Point List.

Icon	Designation	Description
	Line List	Shows/Hides the Line List.
	Circles List	Shows/Hides the Circles List.
	Strings List	Shows/Hides the Strings List
	Contours List	Shows/Hides the Contours List.
	Test windows	Shows/Hides the Test windows.
	PROFINET Status	Shows/Hides the PROFINET Status.
	Parameter-Set-Declaration	Shows/Hides the Parameter-Set-Declaration.
	Parameter-Set-List	Shows/Hides the Parameter-Set-List.
	Parameter-Set-Editor	Shows/Hides the Parameter-Set-Editor.

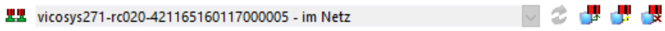
5.3.3 Toolbar - Debugging

With the **Debugging** toolbar, you can insert breakpoints in your program in order to control the flow of the Online Debugging better. Information about Online Debugging and the use of breakpoints, see *"Online Debugging ", page 42*.

The **Debugging** toolbar, provides the following functions from the menu **Utilities** as buttons:

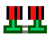
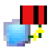
Icon	Designation	Description
	Start/Continue Debugging	Starts or resumes the online debugging.
	Stop Debugging	Stops the Online Debugging.
	Set/Delete Unconditioned Breakpoint	Sets or deletes an unconditioned breakpoint.
	Set/Delete Conditional Breakpoint (error)	Sets or deletes a breakpoint that is triggered if the previous command failed.
	Set/Delete Conditional Breakpoint (good)	Sets or deletes a breakpoint that is triggered if the previous command was free of errors.
	Delete All Breakpoints	Deletes all breakpoints.

5.3.4 Toolbar - Communication



The **Communication** toolbar gives you quick access to functions to set the communication between the host computer and the vision system. Information about connecting and other points of communication can be found at *"The Communication Menu", page 72*.

The **Communication** toolbar, provides the following functions from the menu **Communication** as buttons:

Icon	Designation	Description
	Interface	Opens the dialog for configuring the Communication Parameters.
	Selection of the vision system	Opens a list of vision systems.
	Update List	Updates the list of vision system connected via Ethernet.
	Connect	Connects to the vision system.
	Transfer	Transfers the program to the vision system.
	Disconnect	Disconnects from vision system. Starts program in vision system.

5.3.5 Toolbar Commands Image

The toolbar **Commands: Image** gives you quick access to the most commonly used image commands. If you click on the button, the dialog box of the relevant command opens. Information about the commands and their usage, please see the command reference under *"Image Commands", page 136*.

ADVICE

In the default view the toolbar **Commands: Image** is not displayed.

To show toolbars:

Click the menu **View > Toolbar**, and then click the toolbar **Commands: Image**. A check mark next to the toolbar's name indicates that the toolbar is displayed.

The toolbar **Commands: Image** provides the following functions from the command selection **Image** as buttons:

Icon	Designation	Description
	Grab Image	Captures an image.
	Video Mode	Selection of live image and memory image.
	Display	Configure Acquisition, Work, Demo and Display Page.
	Configure Shutter	Configures the exposure time.
	Synchronous Flash	To trigger signal synchronized image capturing including lighting control.
	Delete Image	Deletes the image or the overlay.

5.3.6 Toolbar Commands Locate

The toolbar **Commands: Locate** gives you quick access to the most commonly used locate commands. If you click on the button, the dialog box of the relevant command opens. Information about the commands and their usage, please see the command reference under *"Locate Commands", page 176*.

ADVICE

In the default view the toolbar **Commands: Locate** is not displayed.

To show toolbars:

Click the menu **View > Toolbar**, and then click the toolbar **Commands: Locate**. A check mark next to the toolbar's name indicates that the toolbar is displayed.

The toolbar **Commands: Locate** provides the following functions from the command selection **Locate** as buttons:

Icon	Designation	Description
	Locate Point	Locates a point in the image.
	Locate Line	Locates a line in the image.
	Locate Circle	Locates a circle in the image.
	Locate Contour	Locates a contour in the image.
	Blob Analysis	Finds related areas (blobs) in the image.
	Search Pattern	Found a previously taught in pattern in the image.

5.3.7 Toolbar Commands Definition

The toolbar **Commands: Definition** gives you quick access to the most commonly used definition commands. If you click on the button, the dialog box of the relevant command opens. Information about the commands and their usage, please see the command reference under *"Evaluation Commands", page 273*.

ADVICE

In the default view the toolbar **Commands: Definition** is not displayed.

To show toolbars:

Click the menu **View > Toolbar**, and then click the toolbar **Commands: Definition**. A check mark next to the toolbar's name indicates that the toolbar is displayed.

The toolbar **Commands: Definition** provides the following functions from the command selection **Evaluation** as buttons:

Icon	Designation	Description
	Define Point	Defines a point from other geometry variables or fixed values.
	Define Line	Defines a line from other geometry variables.
	Define Circle	Defines a circle from other geometry variables.
	Center of Gravity of Contour	Determines the center of gravity and the area of a contour.
	Best Fit Line	Calculates a straight line from points or contours and determines their straightness.
	Best Fit Circle	Calculates a circle from contours or points and determines their roundness.

5.3.8 Toolbar Commands Control

The toolbar **Commands: Control** gives you quick access to the most commonly used control commands. If you click on the button, the dialog box of the relevant command opens. Information about the commands and their usage, please see the command reference under *"Control Commands", page 326*.

ADVICE

In the default view the toolbar **Commands: Control** is not displayed.

To show toolbars:

Click the menu **View > Toolbar**, and then click the toolbar **Commands: Control**. A check mark next to the toolbar's name indicates that the toolbar is displayed.

The toolbar **Commands: Control** provides the following functions from the command selection **Control** as buttons:

Icon	Designation	Description
	Datatransfer	Sends and receives geometry variables.
	Line I/O	Queries single digital inputs or sets single digital outputs.
	Counter	Sets or edits a counter.
	Branch	Sets labels and condition blocks and jumps with and without condition to labels and subroutines.
	Port I/O	Queries multiple digital inputs or sets multiple digital outputs.
	Wait	Stops the inspection process for the specified time.

5.3.9 Toolbar Commands Evaluation

The toolbar **Commands: Evaluation** gives you quick access to the most commonly used evaluation command. If you click on the button, the dialog box of the relevant command opens. Information about the commands and their usage, please see the command reference under *"Evaluation Commands", page 273*.

ADVICE

In the default view the toolbar **Commands: Evaluation** is not displayed.

To show toolbars:

Click the menu **View > Toolbar**, and then click the toolbar **Commands: Evaluation.** A check mark next to the toolbar's name indicates that the toolbar is displayed.

The toolbar **Commands: Evaluation** provides the following functions from the command selection **Evaluation** as buttons:

Icon	Designation	Description
	Distance	Calculates the distance between points, lines and circles.
	Combine Results	Sets results or combines them to other mathematical results.
	Angle	Calculates angles between lines.
	Superimpose	Superimposes geometry variables and text on the image.

5.3.10 Toolbar Commands Check

The toolbar **Commands: Check** gives you quick access to the most commonly used check commands. If you click on the button, the dialog box of the relevant command opens. Information about the commands and their usage, please see the command reference under *"Locate Commands", page 176*.

ADVICE

In the default view the toolbar **Commands: Check** is not displayed.

To show toolbars:

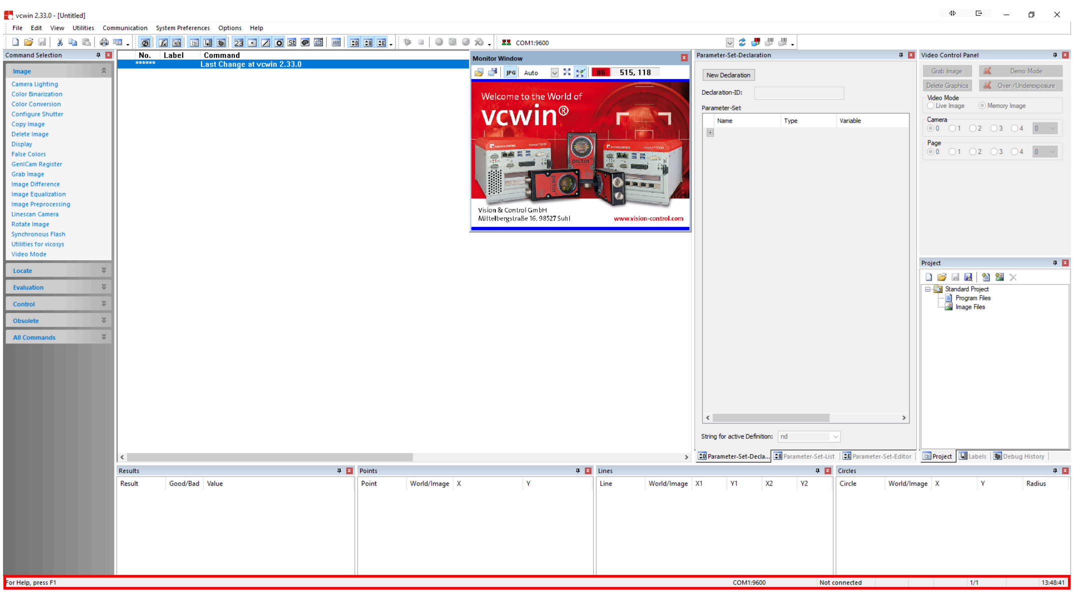
Click the menu **View > Toolbar**, and then click the toolbar **Commands: Check**. A check mark next to the toolbar's name indicates that the toolbar is displayed.

The toolbar **Commands: Check** provides the following functions from the command selection **Locate** as buttons:

Icon	Designation	Description
	Test Gray Value	Determines gray values in an image region.
	Count Pixels	Determines the number of pixels of a specific gray value range in an image region.
	Test Brightness Percentage	Determined the percentage of bright pixels in an image region.
	Inspect Surface	Determines disruptions in an image region.
	Focus	Determines the focus value as a measure of image sharpness.
	Brightness Offset	Compensates variations in the brightness of the image.

5.4 Status Bar

The status bar is located at the bottom of the programming window.



In this section are displayed cyclically read-out program and system information.

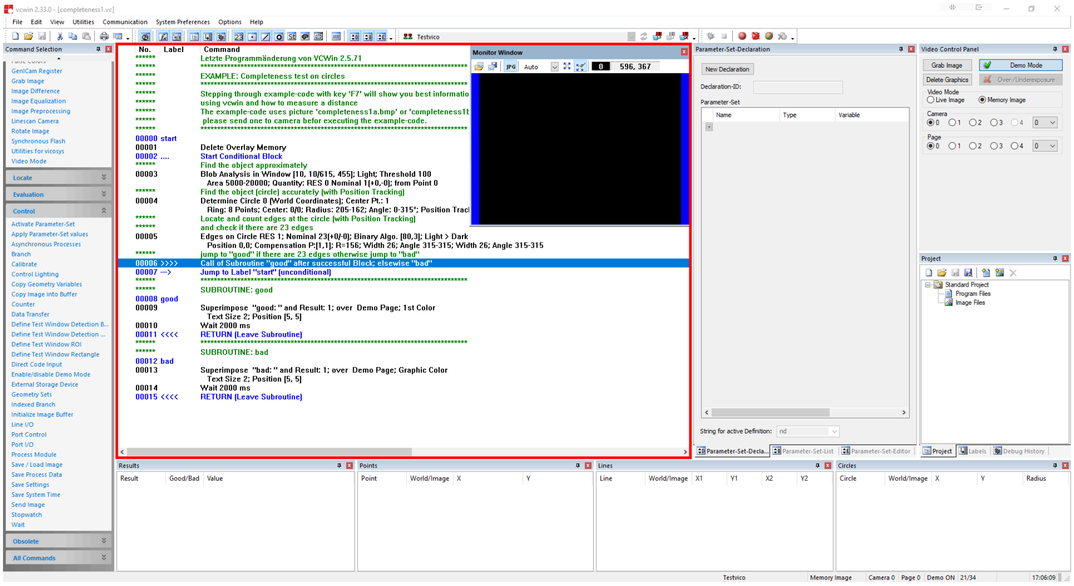
1 Number and result of the last executed command, or a quick help on menu items, symbols and commands.
2 Displays the connection type (COM, TCP) between vision system and host computer or respectively simulator and host computer. And if possible the connection parameters and the vision system type or the name of the simulator.
3 Displays the active Video mode: live image or memory image. If there is no connection to a vision system or simulator, it shows "Not connected".
4 In command mode, the camera number appears when the vision system is initialised.
5 In command mode, the image page number appears when the vision system is initialised.
6 Displays whether the Demomode (graphic overlays in monitor window) is currently on (ON) or off (OFF).
7 Total number of lines in the program as well as the line number of the selected command.
8 Time protocol: displays the duration of the last command executed. Use the time protocol to optimise the timing in test programs. For Test Section and Test Steps, the operating software displays the time required for each step, in turn.
9 The system time of the host.

ADVICE

Time protocol: The time shown for the display of graphics during a test (step mode F7) serves only as a rough guide. The expected time in run mode is somewhat lower.

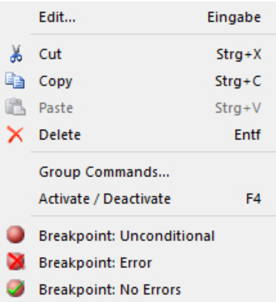
5.5 Command Display

The command display is located centrally in the programming window.



The command display contains the editable commands of the test program in text representation. It contains important information about parameters and variables.

A context menu is opened by right-click in the command window. It contains commands and functions from the main menu.



Insert Commands

Commands can be entered via the docking window **Command Selection** by clicking on the desired command.

Alternatively, you can open a command by simply typing the name with your keyboard. To do so, click in the Command Display and start typing the name of a command. Now a select field pops up and suggests commands based on your input. To choose and open your desired command, click on the name and press [ENTER]. To cancel this operation, press [ESC].

In both cases, the command dialog opens. After completing the configuration, the command is then insert after the last active command (marked in blue).

Edit Commands

Commands can be edited via calling the command dialog. To do this, double-click the desired command in the command display.

Command appearance

Commands are displayed as text in the command list of the command display.

You can change the appearance of the text between short display (abbreviated) or full display via **Options > Full Display**.

Full Display

```
No.    Label  Command
00003         Determine Line 3 (World Coordinates);
              10 Single Detections to 156 Pixels; Search Arrow:
              [557, 154/472, 154]; Max. Gradient [5, 1, 1]; Dark >
              Light
```

Short Display

```
No.    Label  Command
00003         Determine Line 3 (World Coordinates);
```

Font Color

To improve the clarity of the program, different colors are assigned to the program elements.

Wie Sie die Schriftart und Schriftfarbe individuelle anpassen können, erfahren Sie unter: *"Application Settings", page 102*.

Active commands

These commands are shown in black, with serial number, command type and parameters.

```
No.    Label  Command
00001         Standard Image Acquisition with camera 4
00002         Determine Line 3 (World Coordinates);
              10 Single Detections to 156 Pixels; Search Arrow:
              [557, 154/472, 154]; Max. Gradient [5, 1, 1]; Dark >
              Light
```

Deactivated commands

These commands are shown in red without serial number, with command type and parameters. Automatically it is switched to short display.

```
No.    Label  Command
-----         Standard Image Acquisition with camera 4
-----         Determine Line 3 (World Coordinates);
```

Commands can be activated or deactivated via **Menu Edit > Activate / Deactivate** or via the key [**F4**].

Comments

Comments are shown in green without serial number.

```
No.    Label  Command
*****         This is a comment.
*****         This is a comment,
*****         that spans multiple lines and
*****
*****         contains blank lines.
```

Comments can paste from the **Edit** menu > **Insert Comment** or via the key [**F2**].

Branch command

These commands are shown in blue, with serial number, label name and parameter.

```
No.    Label  Command
00000  Start
00001  --->   Jump to Label "Start" (unconditional)
```

Commands for program control (Branch) are:

- Branch
- Indexed Branch
- Port Control

5.6 Docking Windows

An important part of the user interface are docking windows. The docking windows contain commands, tools or indicators for specific tasks and functions.

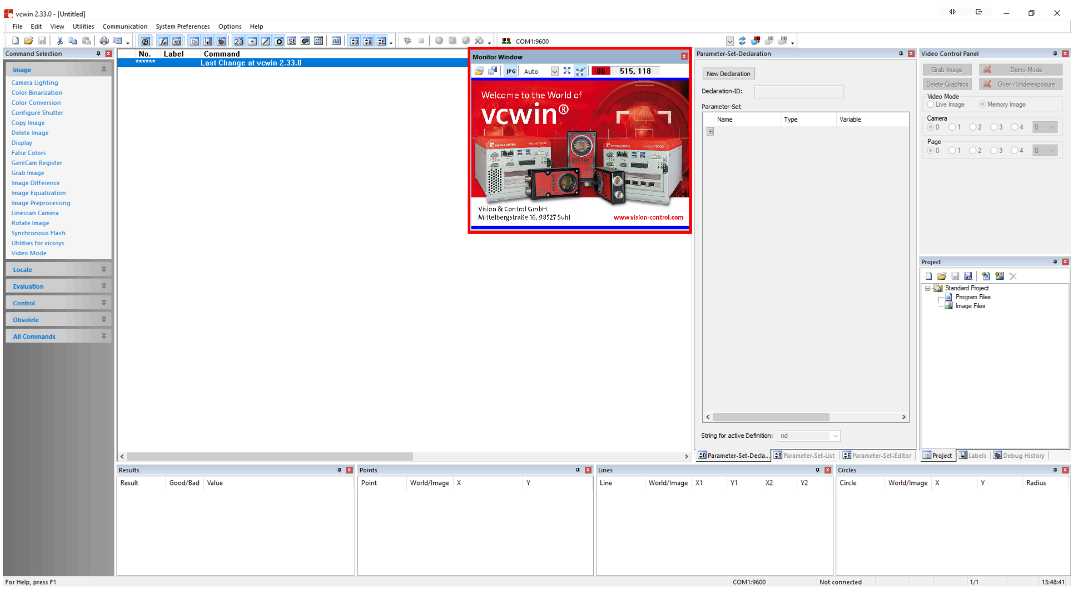
The default setting includes a selection of the most commonly used docking windows.

- **Monitor Window** (floated)
- **Command Selection**
- **Video Control Panel**
- **Project**, **Labels**, **Debug History** (grouped)
- Geometry Lists: **Results**, **Points**, **Lines**, **Circles**
- Geometry Lists: **Strings**, **Contours**, **Test Windows** (hidden)
- **PROFINET Status** (hidden, floated)
- **Parameter-Set-Declaration**, **Parameter-Set-List**, **Parameter-Set-Editor** (grouped)

How you customise the user interface by using docking windows to your needs, please refer to the section *see "Customize interface", page 22*.

5.6.1 Docking Window Monitor Window

In default view, the docking window **Monitor Window** is floating and located in the upper right-hand area of the command display.



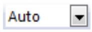
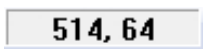
Use the monitor window to evaluate live- or memory images, to teach in patterns or search windows or check parameters and variables, if no control monitor is connected.

To show or hide docking window Monitor Window

Click in the **View** menu > **Monitor Window** or click on the corresponding icon in the toolbar Docking Window. A frame around the icon indicates that the docking window is displayed.

Image 36: ***show or hide Monitor Window***

By using the toolbar of the monitor window, you can run the following commands:

Icon	Designation	Description
	Load Image	Opens an image from a file.
	Save Image	Saves the current image in a file.
	Fast Image Loading	Activates/Deactivates the lossy image transfer (JPEG compressed) from the Vision System.
	Zoom	Sets the magnification factor. Standard: "Auto".
	Fullscreen	Activates/Deactivates the fullscreen mode.
	Automatic Fullscreen Mode	Activates/Deactivates the automatic fullscreen mode for teach-in mode and live image mode.
	Display grey value	Displays the grey value for the current mouse cursor position.
	Display coordi-nates	Displays the coordinates for the current mouse cursor position.

ADVICE

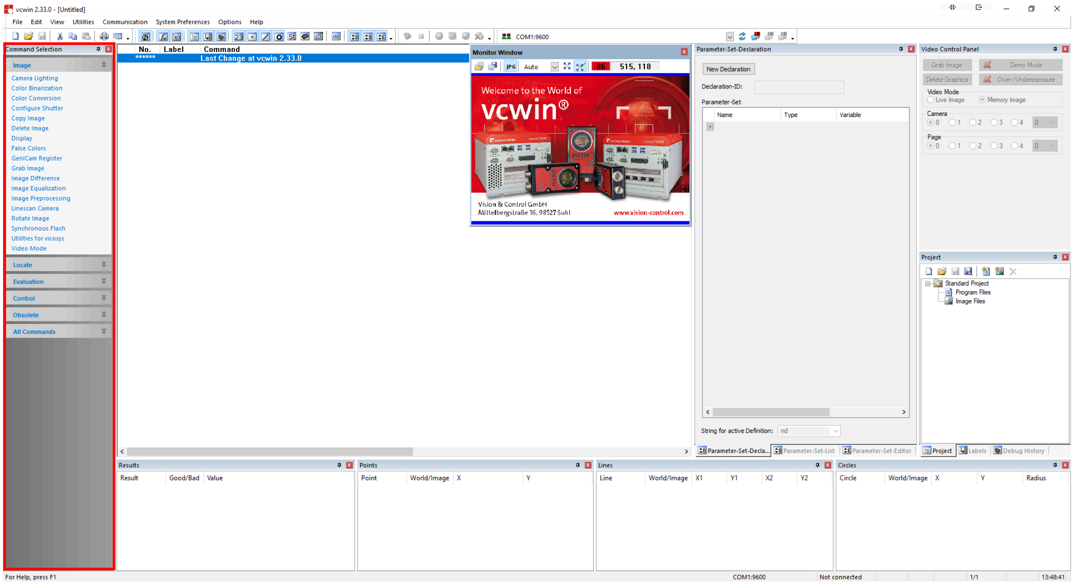
Use of lossy image compression!

If activated (JPG), the picture is fetched from the vision system as compressed and lossy JPG. This enables a faster image transmission at the cost of a lower image quality. It is recommended to not activate this option due to the alteration of grey values that comes with the compression. Also for colour images, some colour values are not consistent with the original. Because of the faster transmission speed, the JPG mode is suited for remote maintenance via VPN.

The JPG quality is set to 75% by default. You can adjust the level of compression in the section Application Settings >>Advanced<<, *see "Application Settings", page 102*.

5.6.2 Docking Window Command Selection

In the default view, the docking window **Command Selection** is located in the left-hand area of the programming window.



The docking window command selection contains all usable image processing commands. They are separated into the categories **Image, Locate, Evaluation, Control,** and **Obsolete** (deprecated features). The category **All Commands** includes all commands in alphabetical order. To open an image processing command from the command selection, click on the appropriate categoryname and click on the required command. The command is opened in an dialog box.

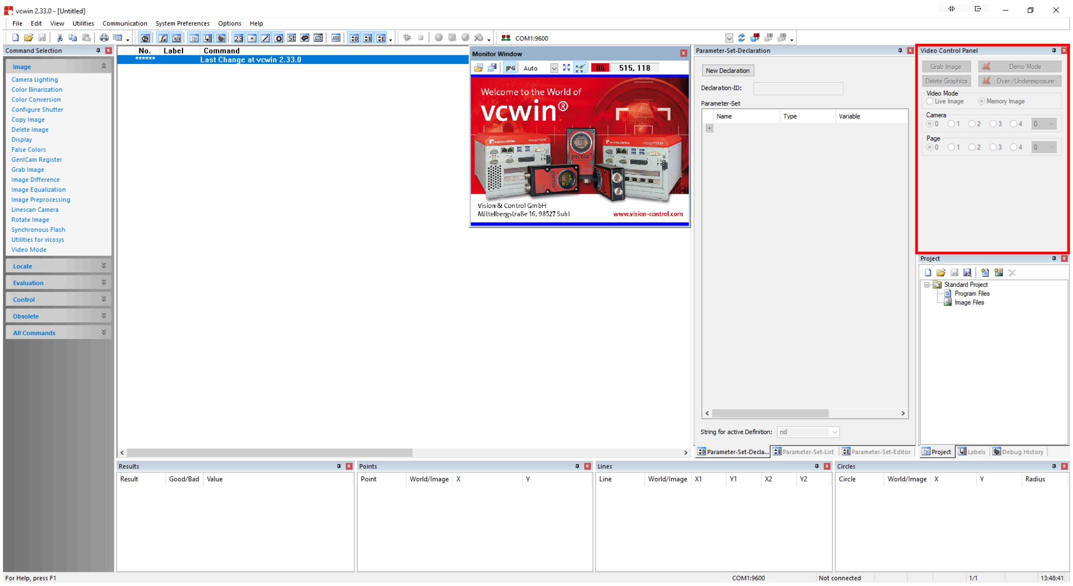
After being connected to a vision system, commands that are not supported by the vision system, are grayed out in the command selection.

To show or hide docking window Command Selection

Click in the View menu > **Command Selection** or click on the corresponding icon in the toolbar **Docking Window**. A frame around the icon indicates that the docking window is displayed.

5.6.3 Docking Window Video Control Panel

In default view, the docking window **Video Control Panel** is located in the upper right-hand area of the programming window.



Use the **Video Control Panel** for grab an image in command mode. It does not write the command into the test program that is being edited. You can set the camera directly in live image mode before the image is grabbed.

The image output take place via the build-in **Monitor Window** (*see "Docking Window Monitor Window"*) or an external control monitor.

To show or hide docking window **Video Control Panel**

Click in the **View** menu > **Video Control Panel** or click on the corresponding icon in the toolbar Docking Window. A frame around the icon indicates that the docking window is displayed.

Image Acquisition

The current image from the vision system is sent to image memory of the selected image memory page. The number of cameras and image pages available depends on the vision system concerned.

Parameters	Description
Camera	Defines the camera for the image acquisition (via radio buttons or combo box).
Page	Defines the image page for the image acquisition (via radio buttons or combo box).
Grab Image	You grab an image from the set camera onto the defined image page with the [Grab Image] button.
Overexpose / Underexposure	Option for displaying overexposed and underexposed areas in an acquired image. Displays gray values from 0 to 10 in red and gray values from 245 to 255 in blue. The display is only on the control monitor.

Overlay Graphic

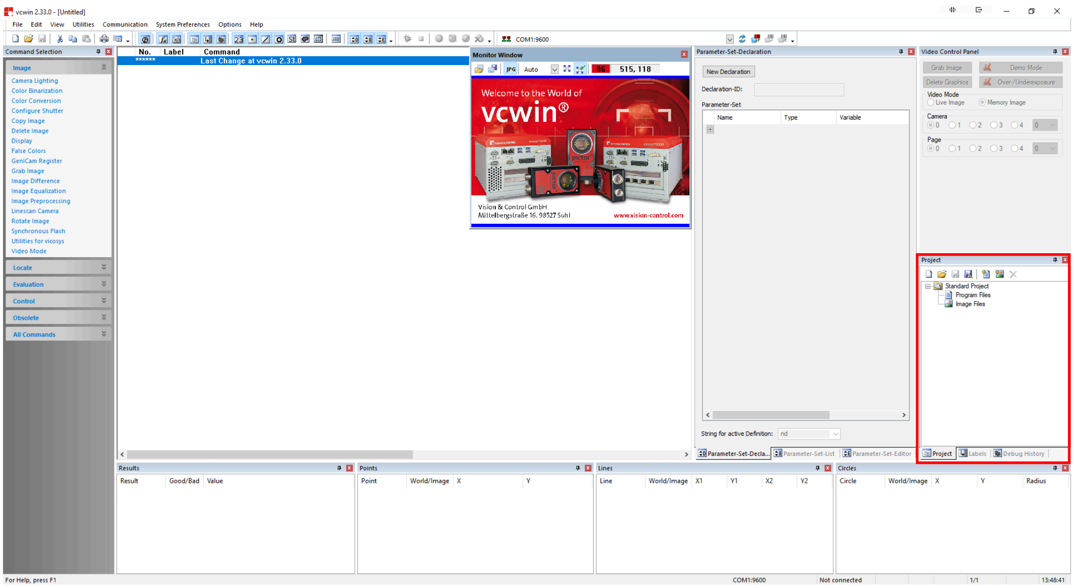
Parameters	Description
Demo Mode	This mode enables/disables the display of overlay graphics (e.g., from detection beams, contours or points) on the control monitor or in the monitor window. At the same time, the **Demo Mode ON/OFF** text is displayed in the status bar.
Delete Graphics	With the button [**Delete Graphics**] you delete the overlay graphics of the current image page on the vision system.

Video Mode

Parameters	Description
Live Image	In Live Image mode the video image from the current camera appears on the video monitor or Monitor View. At the same time the text **Live Image** appears in the status bar.
Memory Image	In memory image mode the current memory image page appears on the video monitor or Monitor View. At the same time the text**Memory Image** appears in the status bar.

5.6.4 Docking Window Control Area

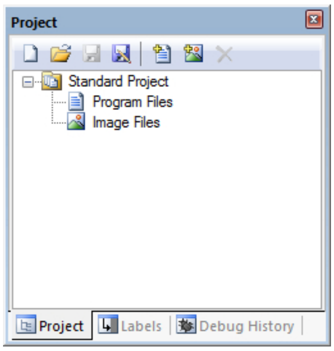
In default view, the docking window **Control Area** (Project, Labels, Debug History) is located in the lower right-hand area of the programming window.



The docking windows **Project**, **Labels** and **Debug History** are in grouped arrangement.

Docking Window Project

In the docking window **Project** you will find the project-tree. You can create and organize your Machine Vision projects here. You can assign and organize program and image files for individual projects within this window.



You can find following functions in the toolbar:

- New Project
- Open Project
- Save Project
- Save Project as
- Add Program File to the project
- Add Image File to the project
- Remove File (Program or Image) from the project

Every project has its own image and program files. Please note that you should first save a new project before you can save the relevant data to the hard disk.

In order to add image or program files, please use the object menu. Programs can be opened from the Project Window by double-clicking the left mouse button. Images can be loaded directly to a image memory page with double-clicking the left mouse button.

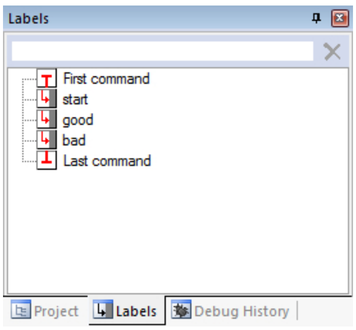
To show or hide docking window Project

Click in the **View** menu > **Project Tree** or click on the corresponding icon in the toolbar Docking Window. A frame around the icon indicates that the docking window is displayed.

Docking Window Labels

In the docking window **Labels** you will find the Labels tree.

Im Markenbaum können Sie die mit dem Steuerungsbefehl Programmkontrolle (*see "Branch", page 367*) gesetzten Marken einsehen.



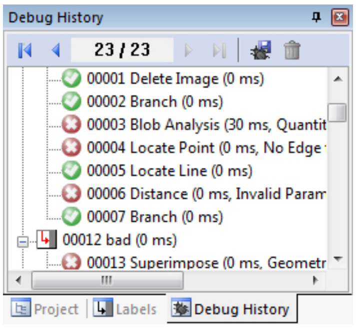
For reasons of clarity, there is a search filter located above. If you enter a search term here, only labels including the search term will be displayed. Use the "X" behind the search filter to reset the filter and display all labels.

To show or hide docking window Labels

Click in the **View** menu > **Labels Tree** or click on the corresponding icon in the toolbar Docking Window. A frame around the icon indicates that the docking window is displayed.

Docking Window Debug History

In the docking window **Debug History** you will find the results of the online debugging.



The debug history displays the results of processed commands from the current debugging process. Weitere Informationen finden Sie unter *"Online Debugging ", page 42*.

To show or hide docking window Debug History

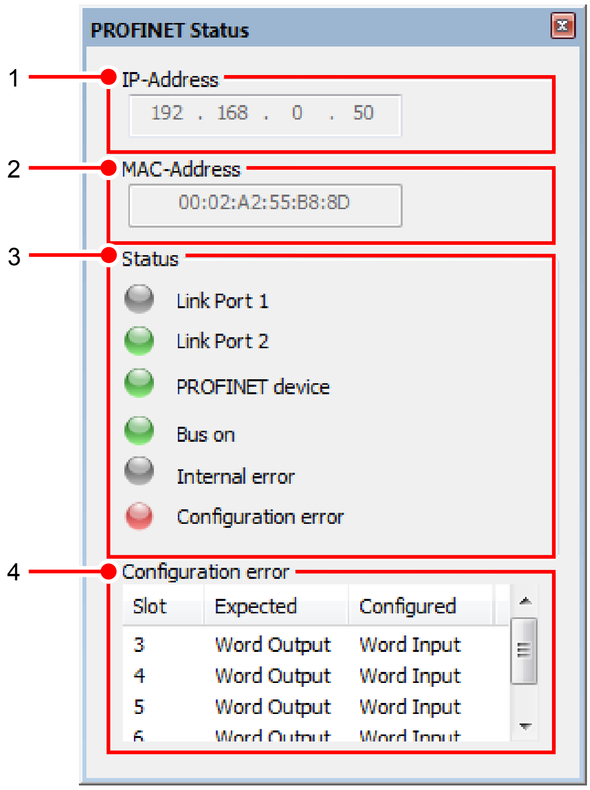
Click in the **View** menu > **Debug History** or click on the corresponding icon in the toolbar Docking Window. A frame around the icon indicates that the docking window is displayed.

5.6.5 Docking Window PROFINET Status

The PROFINET Status docking window is hidden in the standard view.

The docking window PROFINET Status displays the current PROFINET status. The content of the window is deactivated (grayed out) if a device without a PROFINET card is connected.

Dialog



1. IP-Address: IP address of the PROFINET card
2. MAC-Address: MAC address of the PROFINET card
3. Status:

Parameter	Status	Description
Link Port 1	Green	Link active on port 1 - physical connection established
	Gray	Link inactive on port 1 - no physical connection
Link Port 2	Green	Link active on port 3 - physical connection established
	Gray	Link inactive on port 2 - no physical connection
PROFINET Device active	Green	Stack configured and started
	Gray	Stack inactive
Bus on	Green	Cyclic data transfer is running
	Gray	No data transfer
Internal error	Gray	No errors
	Red	An error has occurred on the PROFINET card
Configuration error	Gray	No errors
	Red	A configuration error has occurred - errors are listed in the Configuration error area

4. Configuration error

   Tabulation of the current configuration errors:

Parameter	Description
Slot	Slot at which the error occurred
Expected	Expected module ID
Configured	Configured module ID

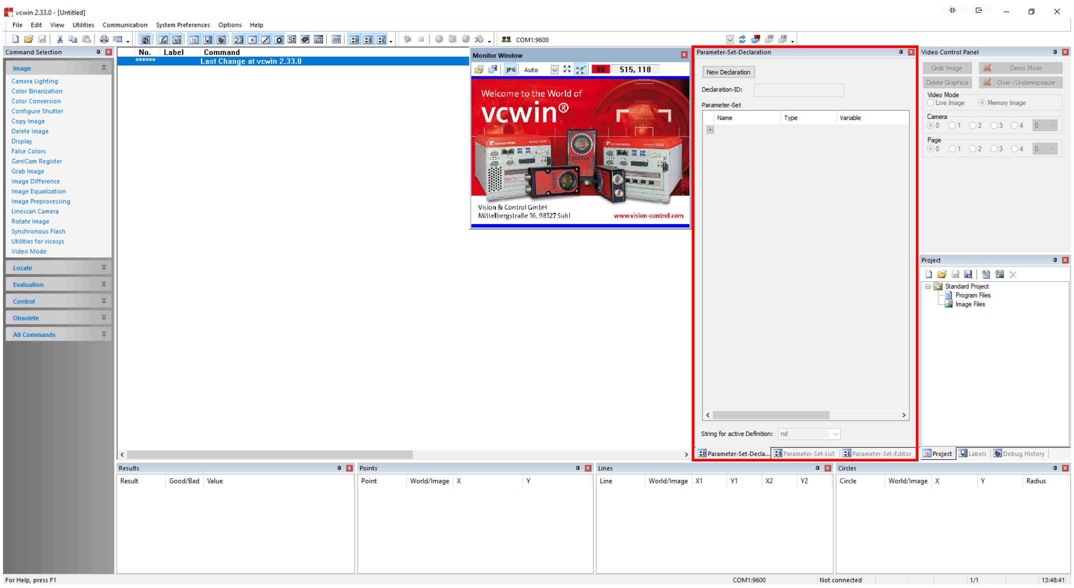
Show or hide the docking window

Click on the menu **View > PROFINET Status** or click on the corresponding icon in the **Docking Windows** toolbar.

A frame around the icon indicates that the docking window is displayed.

5.6.6 Docking Window Parameter-Set-Declaration

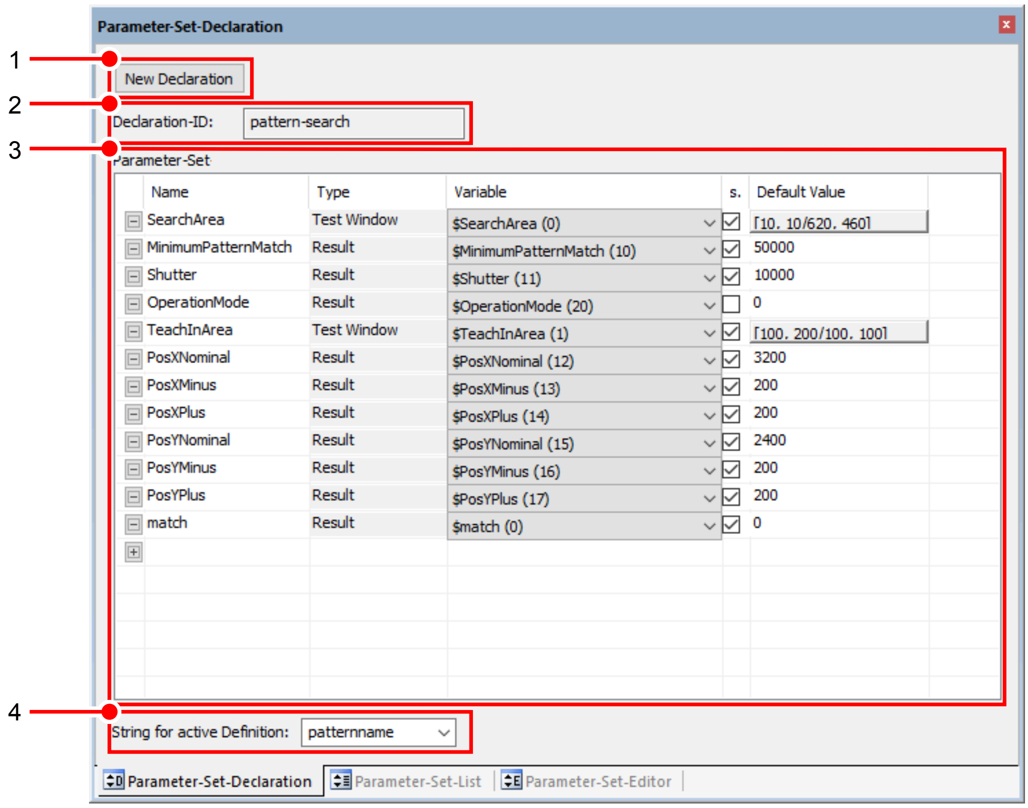
The docking window of the Parameter-Set-Declaration is located in the standard view in the right-hand area of the programming window.



The Parameter-Set-Declaration defines which geometry variables you want to control via Parameter-Sets. Depending on the defined Parameter-Set, different values can be assigned to the geometry variables.

The Parameter-Set-Declaration is saved directly in the test program. The Parameter-Sets are stored on the vision system.

Dialog



1. New Declaration

   With the [New Declaration] button, you create a new Parameter-Set-Declaration.
   If a Parameter-Set-Declaration already exists in the test program, it will be overwritten.

   Enter the name of the new Parameter-Set-Declaration in the following dialog. Use only letters, numbers, and spaces for the name.

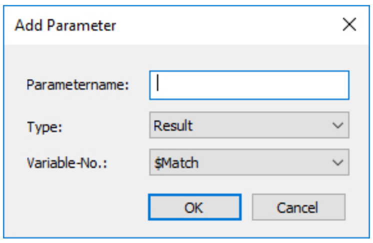
2. Declarations-ID
   The name of the current Parameter-Set-Declaration is displayed here. The field is empty if no Parameter-Set-Declaration has yet been assigned to the test program yet.

3. Parameter-Set

   Here all parameters of the Parameter-Set-Declaration are defined.

   To add a parameter:

   Click ⊞. The Dialog *Add Parameter* opens.



Parameter	Description
Parametername	Enter the name of the parameter here. The designation must be conform to the conditions of the vcwin geometry variables. (ISO / IEC 8859-15, first letter, no special characters)
Type	Select the type of geometry variable (Result, String, Test window, Point).
Variable-No.	All available geometry variable numbers of the selected geometry variable type are displayed in this area. If a name has already been assigned to a geometry variable, it will also be displayed and proposed as a parameter name if selected. The assigned geometry variable gets the dollar sign ($) as a prefix.

[OK] inserts the created parameter with the storage flag and the default values of the geometry variables into the parameter list.

The storage flag indicates whether the value of a parameter should be stored in the Parameter-Set-Definition. A parameter, where this flag is not set, is assigned the default value when changing the Parameter-Set-Definition, regardless of which value has been assigned to the parameter in the meantime.

The default values can be changed in the parameter list by clicking on the corresponding entry.

4. String for active Definition:

   Define the name of the string in which the name of the active Parameter-Set is to be stored.

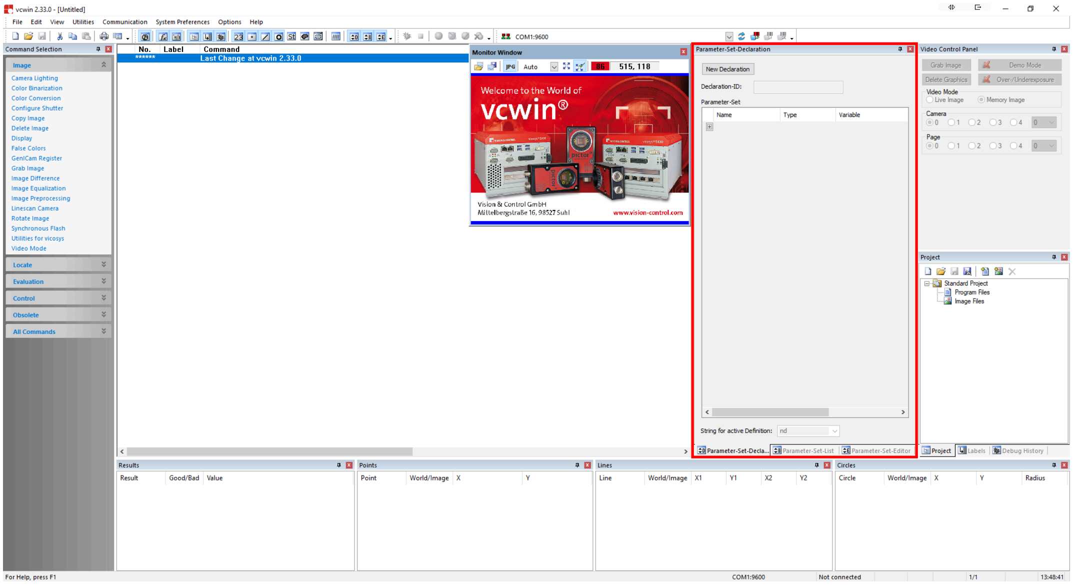
Show or hide the docking window

Click on the menu **View > Parameter-Set-Declaration** or click on the corresponding icon in the **Docking Windows** toolbar.

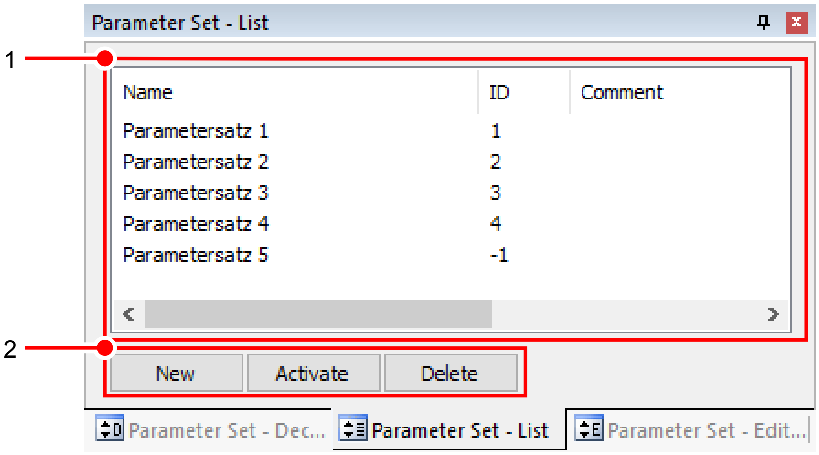
A frame around the icon indicates that the docking window is displayed.

5.6.7 Docking Window Parameter-Set-List

The docking window of the Parameter-Set-List is located in the standard view in the right-hand area of the programming window.



Dialog



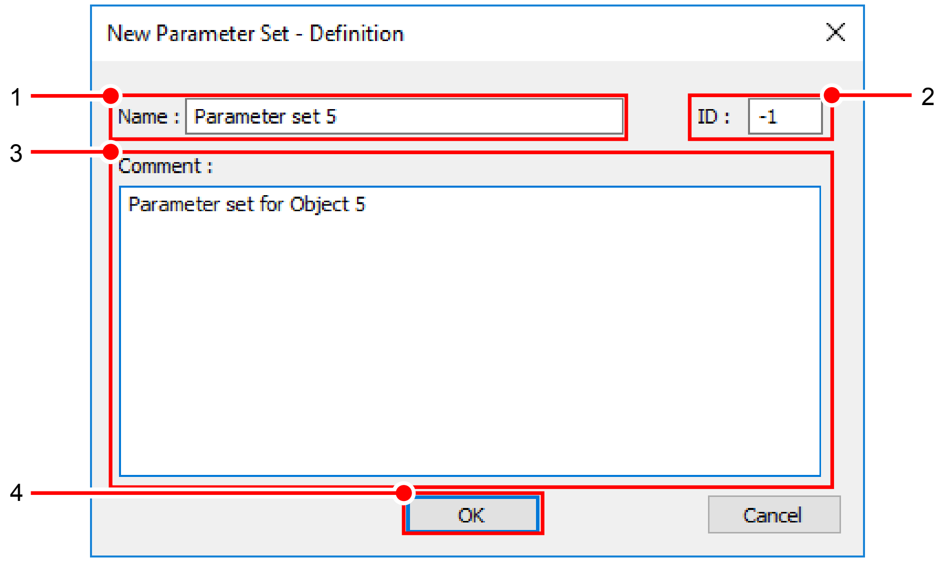
1. List overview

   List of all Parameter-Set-Definitions available on the vision system.
   Empty if no Parameter-Set-Definitions exists.

2. Functions for creating, activating and deleting specific Parameter-Set-Definitions.

New

With [New] you create a new Parameter-Set-Definition.



1. Enter the name of the new Parameter-Set-Definition. Use only letters, numbers, and spaces for the name.
2. Specify the PLC-ID for an external changeover of the Parameter-Set. "-1" means that no PLC-ID is assigned.
3. Here you can add comments to the Parameter-Set-Definition.
4. Insert the Parameter-Set-Definition in the Parameter-Set-List

Activate

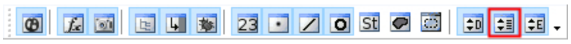
With [Activate] you activate the selected Parameter-Set-Definition. When calling **Control > Apply Parameter-Set Values**, this Parameter-Set-Definition is used.

Delete

With [Delete] you remove the selected Parameter-Set-Definition.

Show or hide the docking window

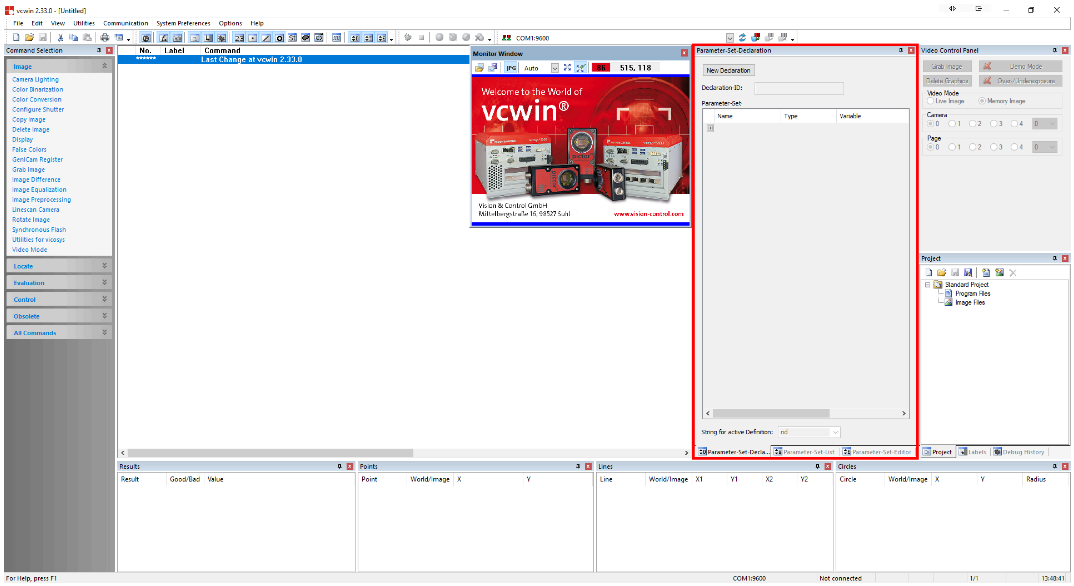
Click on the menu **View > Parameter-Set-List** or click on the corresponding icon in the **Docking Windows** toolbar.



A frame around the icon indicates that the docking window is displayed.

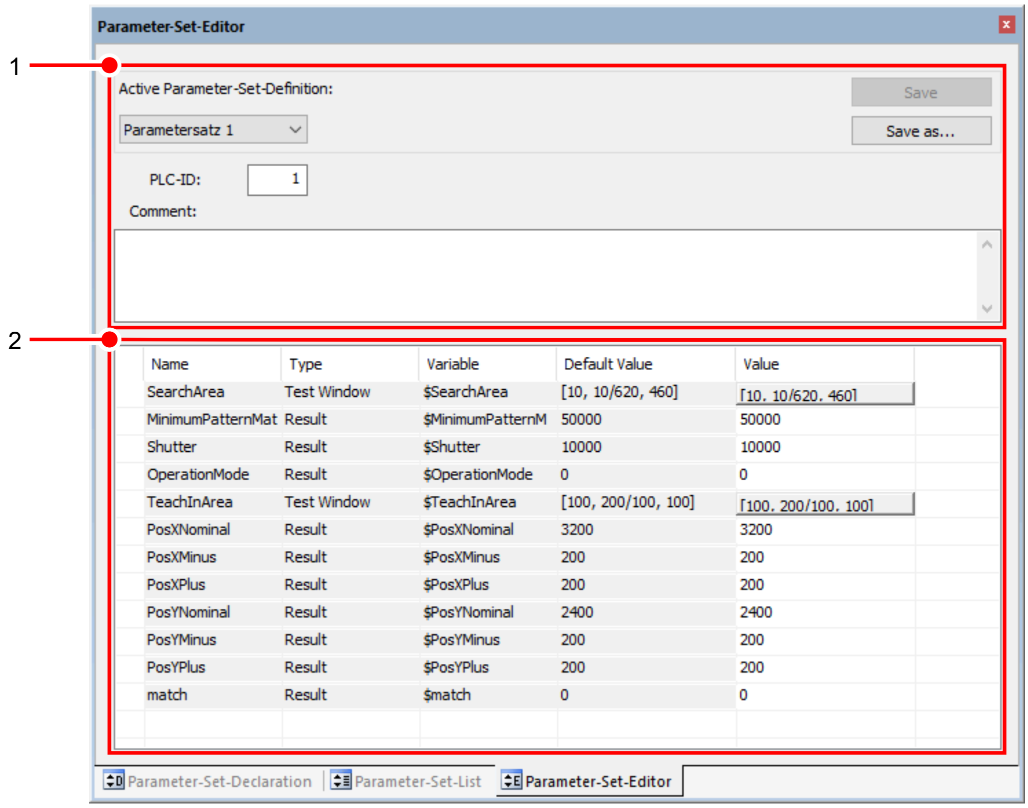
5.6.8 Docking Window Parameter-Set-Editor

The docking window of the Parameter-Set-Editor is located in the standard view in the right-hand area of the programming window.



With the Parameter-Set-Editor you can edit the respective Parameter-Sets.

Dialog



1. Select the Parameter-Set

   Under *Active Parameter-Set-Definition*, select from the drop-down list the Parameter-Set to be edited.
   The selected Parameter-Set is activated and the parameters and values are loaded into the editor.

You can reassign the PLC-ID or change the comment of the Parameter-Set.

The [Save] button is activated as soon as a value of the Parameter-Set has been changed. With [Save] you save the changes made in the RAM of the vision system to the Parameter-Set.

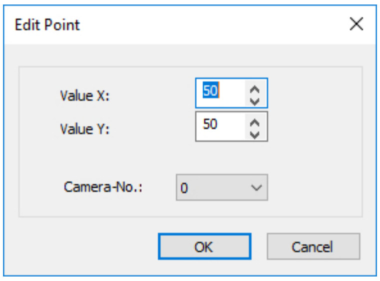
With [Save as] you create a new Parameter-Set from the current values of the active Parameter-Set.

2. Edit parameters

   This area displays the parameters defined in the Parameter-Set.

   You can edit the value of the parameters in the *Value* column.

   - Strings and results can be written directly in the text box.
   - Points and Test windows are edited via the respective edit dialog. Click on the values to be edited. Enter the new values in the edit dialog and confirm them with [OK].



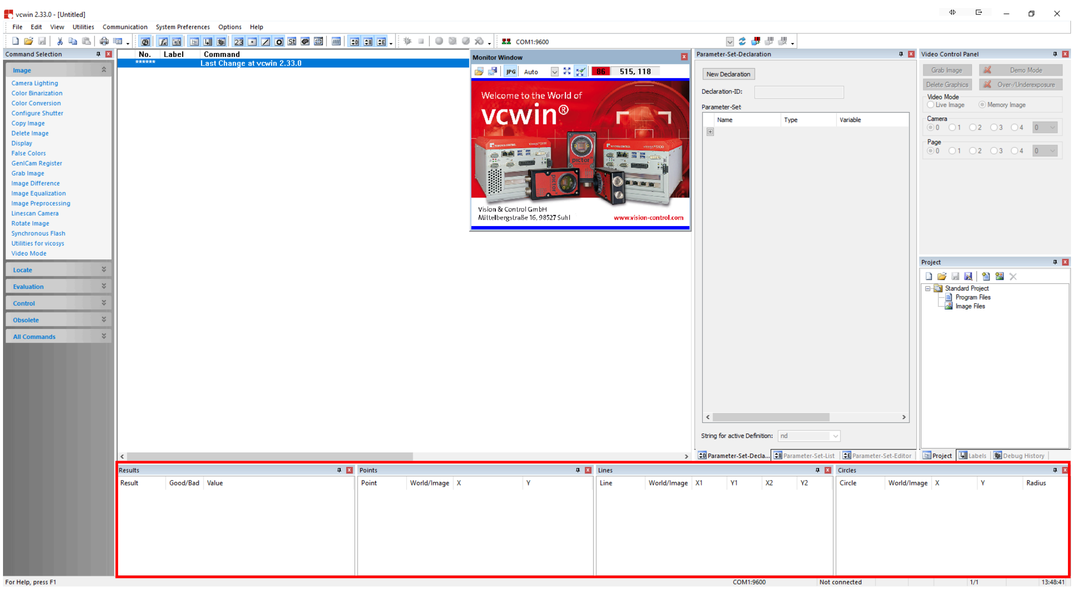
Show or hide the docking window

Click on the menu **View > Parameter-Set-Editor** or click on the corresponding icon in the **Docking Windows** toolbar.

A frame around the icon indicates that the docking window is displayed.

5.6.9 Docking Window Geometry Lists

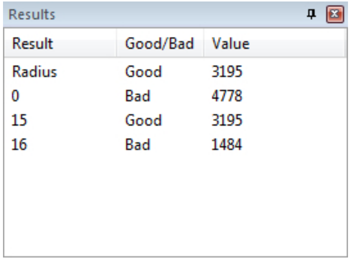
In the default view, the docking windows Geometry Lists are located in the lower area of the programming window.



The geometry lists contain the geometry variables that are generated by executing programs.

Docking Window Results

In this docking window, you can see all currently defined results of the program.



Information:

- number or name of the result
- evaluation of the result
- value of the result

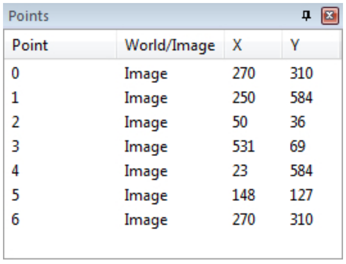
To show or hide docking window Results

Click in the **View** menu > **Result** or click on the corresponding icon in the toolbar **Docking Window**.

A frame around the icon indicates that the docking window is displayed.

Docking Window Points

In this docking window, you can see all currently defined points of the program.



Information:

- number or name of the point
- type of coordinate
- x coordinate
- y coordinate

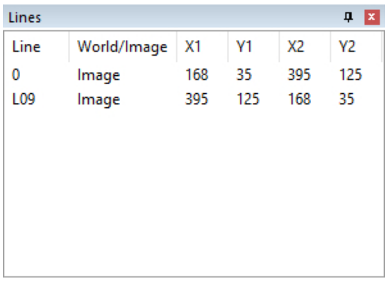
To show or hide docking window Points

Click in the **View** menu > **Points** or click on the corresponding icon in the toolbar **Docking Window**.

A frame around the icon indicates that the docking window is displayed.

Docking Window Lines

In this docking window, you can see all currently defined lines of the program.



Information:

- number or name of the line
- type of coordinate
- x coordinate point 1
- y coordinate point 1
- x coordinate point 2
- y coordinate point 2

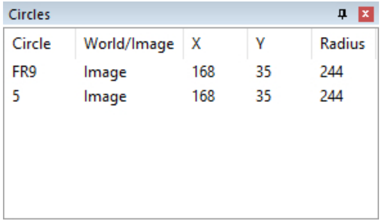
To show or hide docking window Lines

Click in the **View** menu > **Lines** or click on the corresponding icon in the toolbar **Docking Window**.

A frame around the icon indicates that the docking window is displayed.

Docking Window Circles

In this docking window, you can see all currently defined circles of the program.



Information:

- number or name of the circle
- type of coordinate
- x coordinate of the centre of circle
- y coordinate of the centre of circle
- radius

To show or hide docking window Circles

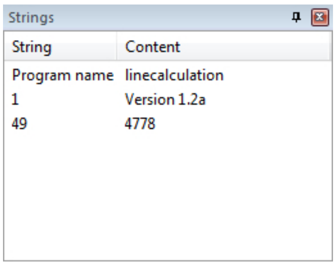
Click in the **View** menu > **Circles** or click on the corresponding icon in the toolbar **Docking Window**.

A frame around the icon indicates that the docking window is displayed.

Docking Window Strings

ADVICE
The geometry list **Strings** is not displayed in the default view

In this docking window, you can see all currently defined strings of the program.



Information:

- number or name of the string
- content of the string

To show or hide docking window Strings

Click in the **View** menu > **Strings** or click on the corresponding icon in the toolbar **Docking Window**.

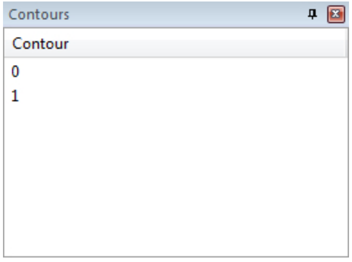
A frame around the icon indicates that the docking window is displayed.

Docking Window Contours

ADVICE

The geometry list **Contours** is not displayed in the default view.

In this docking window, you can see all currently defined contours of the program.



Information:

- number or name of the contour

To show or hide docking window Contours

Click in the **View** menu > **Contours** or click on the corresponding icon in the toolbar **Docking Window**.

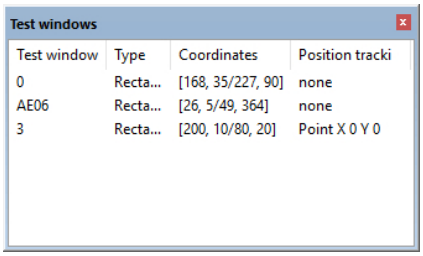
A frame around the icon indicates that the docking window is displayed.

Docking Window Test windows

ADVICE

The geometry list **Test windows** is not displayed in the default view.

In this docking window, you can see all currently defined test windows of the program.

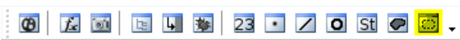


Information:

- number or name of the Test window
- type
- coordinates (X, Y/dX, dY)
- position tracking

To show or hide docking window Test windows

Click in the **View** menu > **Test windows** or click on the corresponding icon in the toolbar **Docking Window**.



A frame around the icon indicates that the docking window is displayed.

6 PART 2 - COMMAND REFERENCE

6.1 Image Commands

6.1.1 Rotate Image

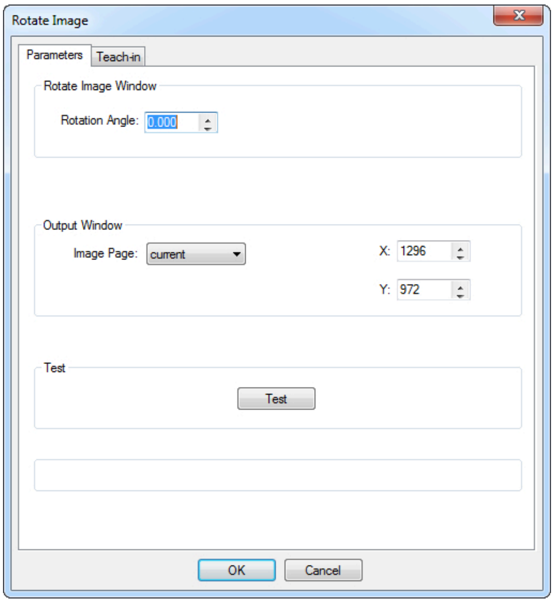


Image 37: Dialogue Rotate Image, tab control Parameters

With **Image > Rotate Image** you can choose an angle and incorporate this in the command for rotating a circular section of an image. The command can be used with position tracking for its x/y positions and rotation. This enables image contents to be rotated automatically by a determined angle.

Brief guide

- Define the following parameters in the Teach-in tab:
  - The size and the position of the image section to be rotated
  - Position tracking e.g. the position of a found pattern and the associated line of rotation as a phi line
  - Mirror mode, if the object to be rotated was found mirrored in the image
- Define more conditions in the parameters tab:
  - A fixed angle can be given for the rotation angle by which the image section is to be rotated
  - Under output window, enter the image storage page to which the rotated image section is to be copied
  - Specify the position in which the rotated image section is to be inserted in the image using x/y coordinates
- Test the command with the [Test] button and insert the command into the test program with the [OK] button
- Verify the result by displaying the specified image storage page via the “Video Control Panel” dialog

Teach-in the Detection Window

The Teach-in window can be only trained as circle.

Position and Appearance of the Search Window

After double clicking with the right mouse button on the **Search Window** tab, you can position and adapt the search window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The search window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input parameters

Parameter	Dialogue tab	Description
Mirror mode	Teach-in	This option specifies whether the image section being rotated is to be mirrored. This directly affects tracking and is therefore located in the teach-in dialog.
Rotation angle	Parameter	Offset angle by which the image is always rotated
Image page		The image page to which the rotated image section is to be copied
X/Y		x/y position in the image at which the rotated image section is to be inserted

Test

When testing, the selected section is rotated by the given rotation angle. The rotated image section is shown in the selected image page.

6.1.2 Copy Image

With **Image > Copy Image** you add a command to copy an image memory page into another image memory page. The source page and the output page can either be defined by result, or be entered directly. The overlay of the source page can also be copied.

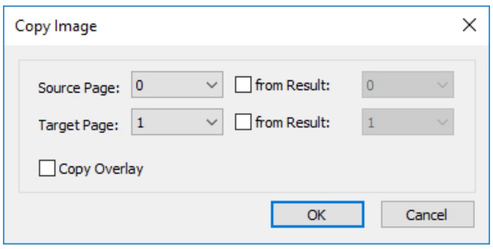


Image 38: Dialogue: Copy Image

Brief instructions

1. Enter the number of the source page or select the result that contains the number of the source page.
2. Enter the number of the output page or select the result that contains the number of the output page.
3. If you want to copy the overlay, activate the *Copy Overlay* check box.
4. Use the [OK] button to insert the command into the test program.

6.1.3 Delete Image

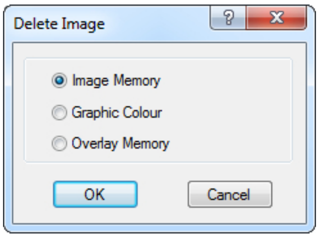


Image 39: Dialogue Delete Image

Insert a deletion command into the test program using **Image > Delete Image**. This way you clear the image memory, graphic page or overlay memory depending on the selected option.

Options

Options	Description
Image Memory	Clears the content of the work page.
Graphic Color	Clears the demo page superimpositions.
Overlay Memory	The overlay memory is a separate memory for graphics (statisticly displayed assistance lines, counters, numbers, texts), which corresponds to the image memory. No image information is overwritten when the overlay memory is used. The overlay memory information is superimposed on the image memory information. The demo page superimpositions are likewise cleared. **Note**: Because the overlay memory is a separate memory, there are differences between the different types of vision systems.

6.1.4 Grab Image

Write a command in the test program to acquire an image and transfer it to image memory using **Image > Grab Image**.

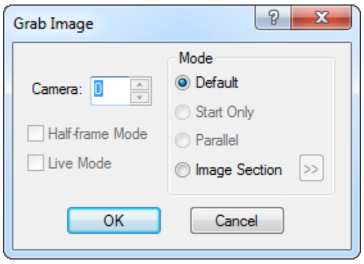


Image 40: Dialogue Grab Image

Options

Camera

Select one of the cameras installed. vcwin automatically recognizes the number of cameras available during connection. The camera number for vicosys can be checked with the **Sort Cameras** command in the utilities for vicosys.

Mode

Parameters for Mode option	Description
Default	Acquiring an image takes place in the current image memory page. The system waits until acquisition is complete before carrying on with the next commands.
Start Only	The image aquisition "start only" waits until a previous image aquisition is terminated, if this still runs. That is called he synchronizes itself with the processing time. Image acquisition starts. An image is written into the current image memory page. At the same time, the system carries on with the next commands.
Parallel	The parallel image aquisition waits until a previous image aquisition is terminated, if this still runs. That is called he synchronizes itself with the processing time.The first image is acquired in an image memory page, which then acts as an image acquisition page. The previous image memory page acts as the image acquisition page for the next image acquisition. Each image acquired is processed on the same image memory page as it was acquired in — this page changes its function from the image acquisition page to the processing page.
Image Section	Allows substantially accelerated image acquisition within a small limited region (ROI).

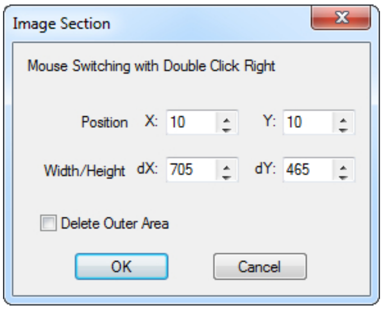


Image 41: Dialogue Grab Image > Image Section

To select an image section

ADVICE

Use multiple adjacent image sections, each with the appropriate gain, in order to capture inhomogeneously illuminated objects.

1. Activate the Image Section option, then click on the double arrow.
2. Set the area for image acquisition via either the options in the rectangle parameters or by hand.

Input Parameters

Image Section Input Parameters	Description
Position	Position of the rectangle
Width / Height	width/ height of the rectangle
Delete the Outer Area	Option **active**: The area outside the rectangle selected is deleted. Option **inactive**: Image pages outside the rectangle selected remain unchanged.
Double-click Right	Double-clicking the right-hand mouse button toggles from the **Image Section** dialog to the control monitor. The position and size of the rectangle for image acquisition in the video image can then be changed as required. Double-clicking with the right-hand mouse button again toggles back to the **Image Section** dialog box.

Tuning Parameters

The **Half-frame Mode** and **Live Mode** options enable program time to be saved. These functions are not supported by all vision systems .

ADVICE

The camera must always be calibrated in the mode in which the image was acquired (half-frame mode or live image mode). The calibration should be made immediately after the image acquisition.

Tuning Parameters for Half-frame Mode and Live Mode	Description
Half-frame Mode	In order to save time, only every second image line is acquired. The acquired image has only half the resolution. This option is not released for progressive-scan cameras (pictor M1208 … M1418) – for which a full image acquisition is always performed.
Live Mode	Only for pictor M type cameras. If the option is activated, the camera remains in live image mode after the image acquisition.

6.1.5 Image Difference

With **Image > Image Difference** you write a command to the test program, which subtracts the gray value information of a reference image from that of a comparison image within an overall image or an image detail.

Use this command to detect changes between two images. The reference page is always subtracted from the comparison page.

Only pictures of the same type can be compared. It is not possible to compare a gray-scale image with a color image.

The reference page is always subtracted from the comparison page.

The absolute value of the resulting difference image can be stored in an output page.

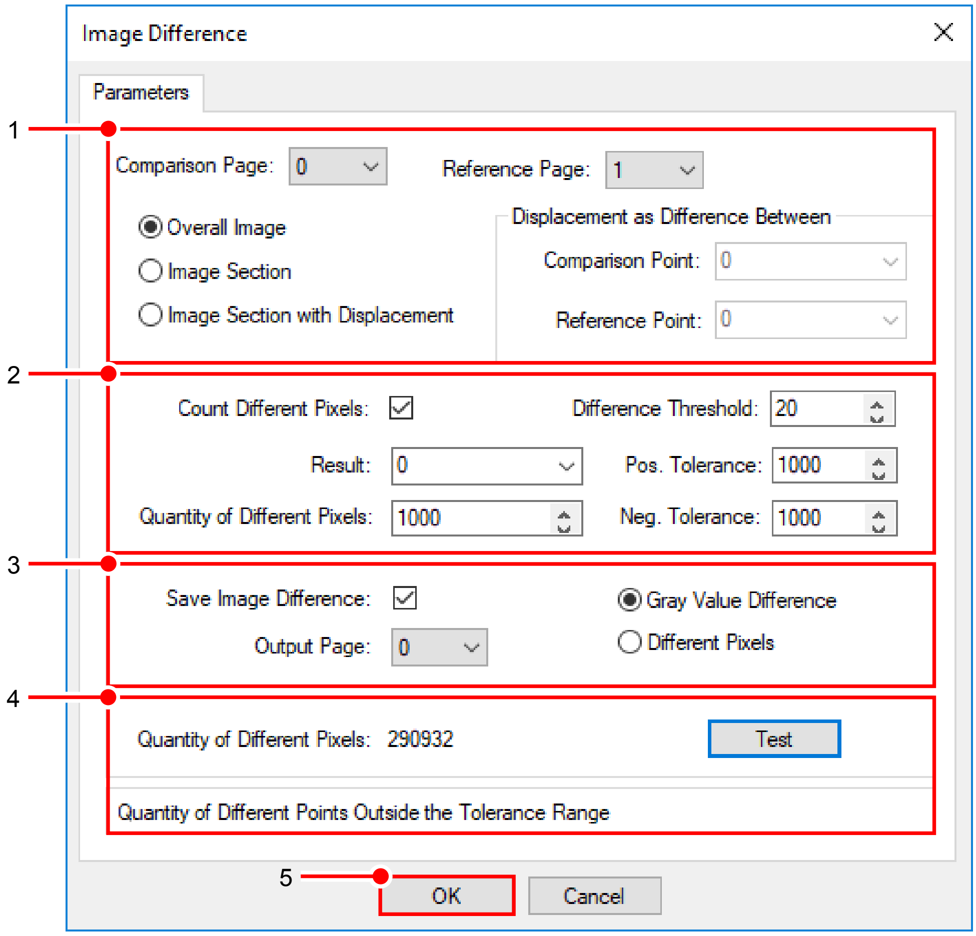
Parameters of the function

Setting parameters	Description
Comparison Page	Contains the comparison image
Reference Page	Contains the reference image
Overall Image	The image difference is determined in the overall image.
Image Section	The image difference is determined in the image section. The image section is defined in the *Teach-in* tab.
Image Section with Displacement	The image difference is determined in the image section. In addition, a displacement is made by the difference of the positions of the following two points. • Comparison Point (point on the comparison page) • Reference Point (point on the reference page)

Evaluation Parameter	Description
Count Different Pixels	If the option is active, the difference pixels are counted.
Difference Threshold	Gray value difference between comparison page and reference page. If this is exceeded, a pixel is evaluated as a difference pixel.
Result	Result number or identifier for storing the result. The number of differential pixels is stored.
Quantity of Different Pixels	Nominal value of difference pixels
pos / neg Tolerance	Allowed upper and lower deviation from the nominal value

Output and Saving Parameters	Description
Save Image Difference	If the option is active, the difference image is saved.
Gray Value Difference	The gray value difference is stored for each pixel.
Different Pixels	The difference pixels above the difference threshold are retained. Pixels below the difference threshold are stored in black.
Output Page	Output page in which the difference image is to be saved.

Brief Instruction



1. Define
   - Comparison Page and Reference Page
   - Method: Overall Image, Image Section or Image Section with Displacement
     Specify the desired Image Section when using the "Image Section" and "Image Section with Displacement" methods.
     Define the Comparison Point and the Reference Point when using the "Image section with displacement" method.
2. Set the specifications for the evaluation of the function (Difference Threshold, Quantity of Different Pixels and Tolerances). The result is considered good if the determined quantity of differential pixels is in the range of the Quantity of Different Pixels incl. Tolerance.

   Set the specifications for storing the result.
3. Set the specifications for saving the difference image with the associated settings.
4. Test the function

   After activating the [Test] button, the result as well as status or error messages are displayed.

   Modify the parameters set in point 1 and point 2 until the test result is correct.
5. Insert the command into the test program with the [OK] button.

6.1.6 Image Equalization

Overview

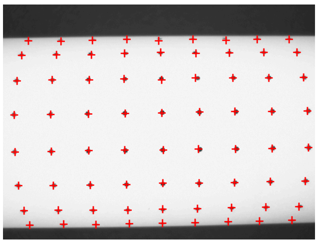


Image 42: grid pattern found

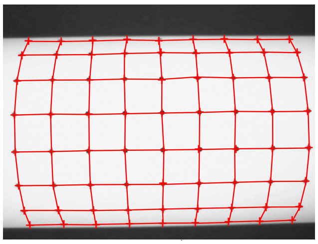


Image 43: points connected, correction grid determined

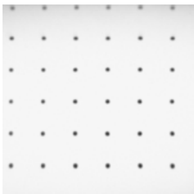


Image 44: equalized image

With **Image > Image Equalization** you add a command for correction of perspective distortions and for unwinding folded or circular bended surfaces (e.g. half spheres, pipes, etc.). Image distortions caused by lenses (pillow-shaped distortion, barrel-shaped distortion) can also be corrected with this command.

ADVICE

For highly accurate measurements, the use of telecentric lenses is recommended. Due to the recalculation and equalization measurement errors can occur because of shifted pixels.

For the image equalization, respectively a grid pattern is required as the template, so that the existing distortion of the lens or the object to be detected can be detected by the software. Based on this grid pattern, a correction grid is calculated which equalizes the image section in the image.

An example grid can be found in the appendix. *see "Dot Matrix", page 434*

Preparation

Depending on the application, you have several options.

- Place the grid pattern in the field of view of the camera at the position where later the test object is to be located. You achieve the correction of lens distortion.
- Sheath the subsequent test object (blank, box, tube etc.) with the grid pattern (see image in overview). Place the object at the position where later the test object is to be located. You achieve the unwinding of the surface.

Work Flow: Teach-in Distortion

1. Switch to the tab control "***Teach-in Distortion***".
2. Capture an image from the grid pattern by clicking the button [Grab Image].
3. Determine, by means of the geometry rectangle, which image content has to be read as the distortion.
   - the size and position of the area to be taught (ideally, the number of points should be the same on each line in the x-direction).
   - the gray value threshold for the grid pattern

   Information for teaching, see also: *"Teaching in Test, Detection and Search windows", page 26*.

4. Click on the Button [Teach Distortion]. If the pattern is not recorded correctly, a message in the lower left part of the dialog appears. Accordingly, adjust the parameters.

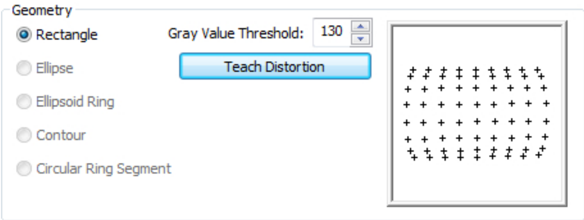


Image 45: Dialog Image Equalization, tab control Teach-in Distortion

Work Flow: Determine Correction Area

1. Switch to the tab control "***Teach-in***".
2. Set the position of the correction window, the size is already given through the window of "Teach-in Distortion". In this area the image will be equalized. If the test object is always in the same place, you do not need to change this option.
   For correcting lens distortion, this area should not be changed; otherwise the lens is equalized incorrect. (see also: *"Teaching in Test, Detection and Search windows", page 26*)
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *"Position Tracking of Objects", page 35*)

ADVICE

The position tracking rotates the grid to the angle change of the phi-line!
The moment where a phi-line is selected, the angle is saved. If the angle changes (e.g. by Locate Line), the line is rotated about the difference of the new angle and the stored angle.

Work Flow: Parameterizing and Testing

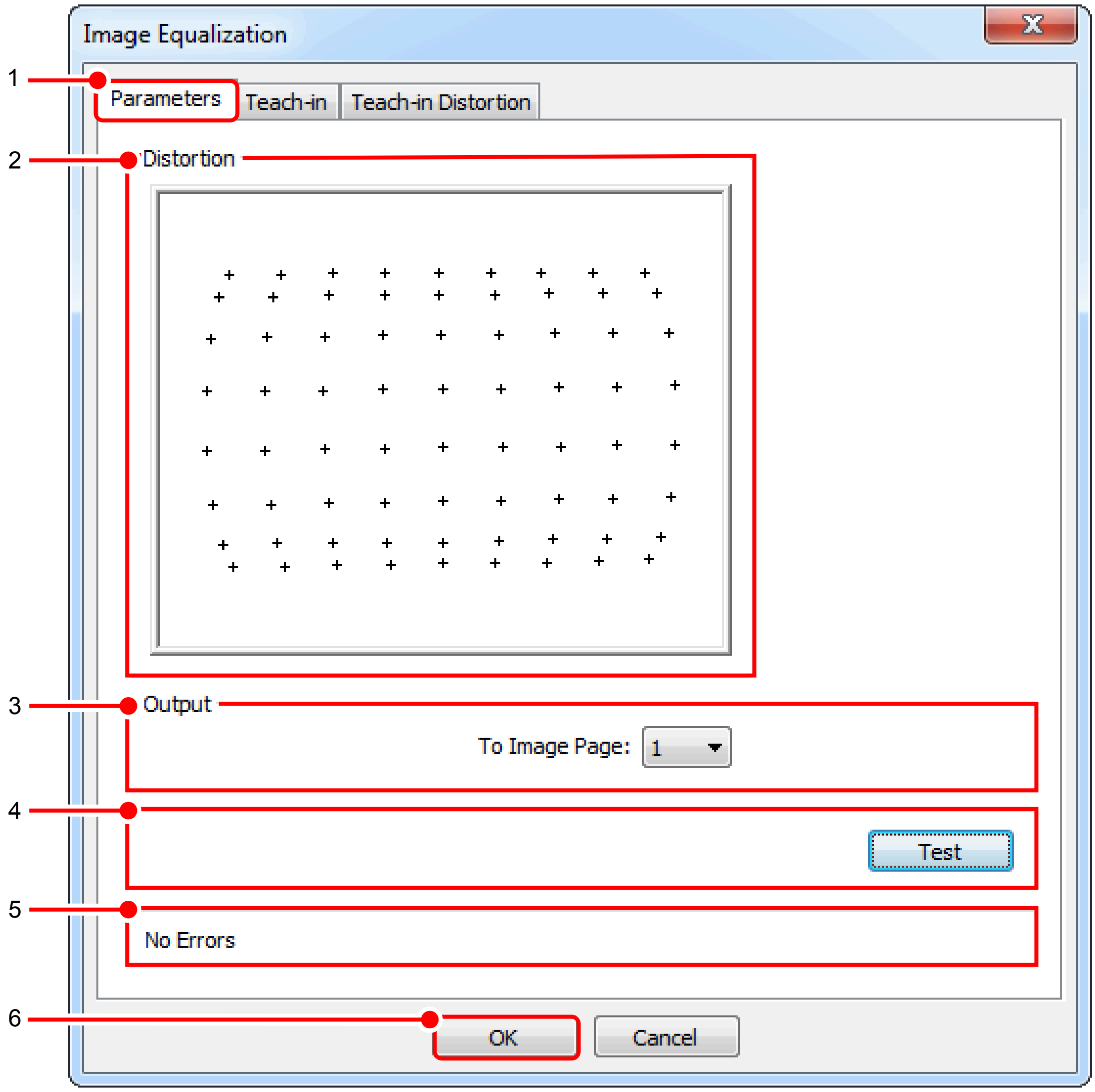


Image 46: Dialog Image Equalization, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Here the current distortion grid is displayed. This grid is used for equalization of all with this command equalized images.
3. Specify the image page, to which the equalized image should be copied to for further processing. The command copies the image that is determined by the grid (regardless of its location) to the center of the target page. In combination with position tracking images can be brought in to a stable position, in which later tests can be carried out without further tracking.

ADVICE

The setting of -1 means that the respective current work page will be used. This can be specified with the command Display (see also: *"Display"*).

For optimum performance, the target page should be different from the source page. Otherwise, the content of the source page is also copied from the image processing system. This has a greater expenditure of time as result.

4. Testing and Evaluate suitable parameters
   The command is executed with the parameters by activating the [Test] button. The selected area is equalized using the calculated correction grid and the equalized image is shown in the chosen image page.
5. Use the [OK] button to insert the command into the test program.

6.1.7 Image Preprocessing

Enable an image to be acquired and test the image, or a taught region of it, using various filters with the **Image > Image Preprocessing** command.

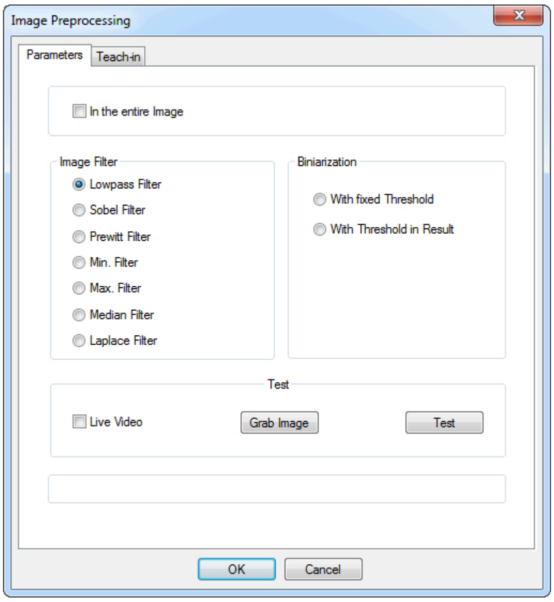


Image 47: Dialogue Image Preprocessing, tab control Parameters

Summary

1. Use the checkbox labeled **In the entire image** to determine whether the image pre-processing should apply to the entire image or only to a section of it.
2. When the checkbox is activated the needed geometry date can be defined on the Teach-in tab.
3. Select and parameterize a filter.
4. Select between live image and memory image:
   - Mark **Live Video**, if the image pre-processing should take place in live image mode.
   - Cancel **Live Video**, then click on **Grab Image**, if the image pre-processing should take place in memory image mode.
5. Click **Test**.

Filters

All linear filters employ a 3 x 3 matrix.

Filters	Use / Result of the filtering
Lowpass	- Noise suppression. - Filter out higher frequencies. - Multiple use increases the effect.
Median	- Suppression of so-called ‘salt & pepper’ noise. - The edges remain preserved.
Sobel	- Edge filter with integrated smoothing. - The exact location of the edge is at the maximum brightness. - The result image displays the contrast.
Prewitt	- Edge filter without integrated smoothing. - The exact location of the edge is at the maximum brightness. - The result image displays the contrast.
Laplace	- Every edge results in a double contour. - The edge location is not the maximum. - The direction of the edge is decided by the sign.
Min.	- The edge location is shifted towards the darker direction. - Dark areas become somewhat larger. - Results in sharper lines from sharper edges.
Max.	- The edge location is shifted towards the lighter direction. - Light areas become somewhat lighter. - Results in sharper lines from sharper edges.

6.1.8 Display

Specify the usage of the various image memory pages using **Image > Display**. The following types are possible:

- Acquisition page
- Work page
- Demo page
- Display page

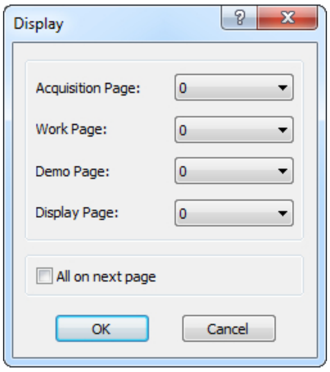


Image 48: Dialogue Display

Image Memory Pages

Image memory pages are memory areas in the vision system, which you can use for image acquisition, image display, image processing, for superimposing geometry variables etc.

Depending upon the vision system, up to 24 images can be stored in different image memory pages.

ADVICE

Refer to the hardware manual for the number of image memory pages available on the vision systems of the type pictor M.

Vision systems of the type pictor T have 24 image memory pages.

The number of image memory pages available on vision system vicosys depends on the resolution of the connected cameras.

You can reserve four image memory pages for special tasks:

Image Memory Page	Image memory page reservation possibilities
Acquisition page	This page is reserved for the image acquisition.
Work page	The vision system operates with its image processing algorithms on this page.
Demo page	Demos for illustrating the processing are stored during the image processing on this page. Superimposing takes place either in graphic colour or with the colours 1, 2 or 3. In both cases the image contents should not be overwritten. Information about turning the demo mode on or off you will find here: *see "Docking Window Video Control Panel", page 121*
Display page	The page is displayed on the monitor window or on the control monitor.

ADVICE

Instead of specifying a page, you can also select "unchanged" or "next page".

If you want e.g. to set only the acquisition page on the next page and do not change the other pages, you can set the acquisition page on "next page" and the other 3 pages on "unchanged".

Working with Image Memory Pages

- Separate the acquisition page from the work page to increase the processing speed of the test programs. Thus, in parallel processing mode (parallel image acquisition), you can start, for example, an image acquisition (grab image), and while the image acquisition is using on the **acquisition page**, the previously acquired image is processed on the **work page**.
- Separate between the work and demo page if, e.g., a scene is analysed under differing illuminations, but the results are only to be displayed in one image.

Use of Next Free Page	Description
All on next page	When using this option, each of the 4 pages are placed on the next page. **Use**: If there is a defective part in the testing process, the image of this part remains on the previous image page even if a new program sequence has already started to run.

6.1.9 False Colors

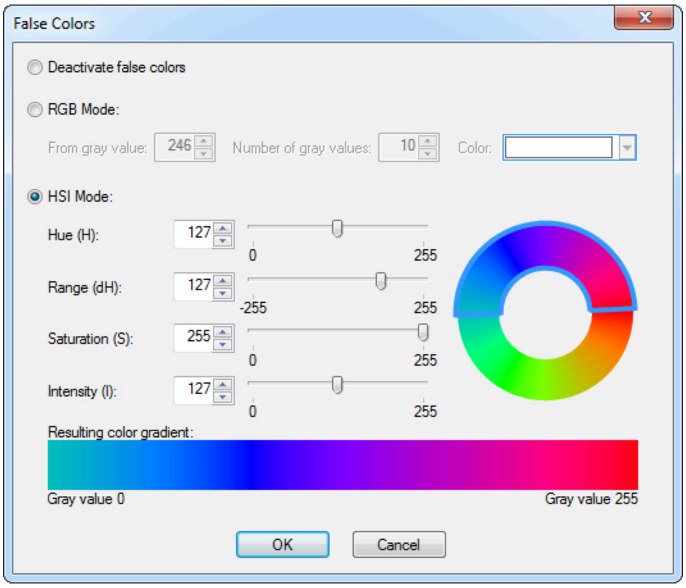


Image 49: Dialogue False Colors

With **Image > False Colors** you can assign specific colours to an image. This allows a better graphical visualisation with fine nuances.

The main use of false colours is particularly in the area of thermal imaging. However the algorithm can be applied to every grey value image.

Parameters for RGB-Mode

RGB-Mode	Description
From gray value	Beginning of the grey scale range. Example: From gray value "100" equates to the grey value "100".
Number of gray values	Number of Grey values starting with "From gray value". Example: Value of 50 starting with "From gray value" of 100 equates to a grey scale range from 100 to 149.
Color	Declaration of the colour that will be assigned to the grey scale range.

Parameters for HSI-Modus

The HSI-Colourspace is a specific colour model, which describes the colour with the help of hue, saturation and intensity.
The HSI-Mode is applied to all 256 grey values.

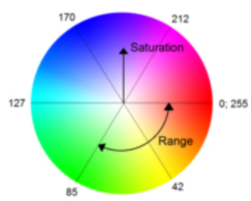


Image 50: HSI-Colourspace

HSI-Mode	Description
Hue (H)	Declaration of hue e.g.: 0 = red, 42 = yellow, 170 = blue, 255 = red This value is assigned to the grey value "0".
Range (dH)	Declaration of the colour range that is applied consistently and equally to all grey values. Starting with the hue you can set the colour range clock-wise or anti-clockwise. It can be either positive or negative. The value of hue + colour range is assigned to the grey value "255".
Saturation(S)	Saturation of the colour range, e.g.: 0 = neutral grey, 128 = slightly saturated colour, 255 = completely saturated pure colour. The saturation affects all 255 used colours consistently.
Intensity(I)	Declaration of the colour lightness (colourfullness), the total amount of light passing through a particular area. e.g.: Intesity of "0" = no light passing (black); Intensity of "255" = full lightness. The intensity affects all 255 used colours consistently.

Example:

Hue (H)= 50; Range (dH) = 100; Saturation (S) = 255; Intensity (I) = 127;

The values are: gray value "0" H = 50, S=255, I=127 and grey value "255" H = 149, S=255, I=127. The colours are applied consistently and equally to all grey values of the grey scale range.

The resulting colour gradient is shown in the control window below.

Image 51: Dialogue False Colors > Resulting color gradient

6.1.10 Color Binarization

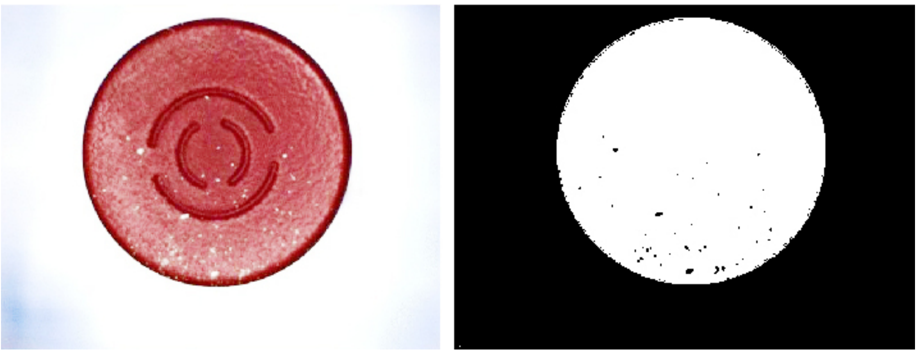


Image 52: Color Binarization before (left) and after (right)

With **Image > Color Binarization** you convert areas of a color image into a binary (Black/White) image. Use this command for colored objects, in order to process them with classical detection procedures. The areas that should be binarized must contain similar color properties in RGB or HSI format. The resulting black-and-white binary image contains only black pixels (gray value 0) and white pixels (gray value 255).

Work Flow
Define Area of Binarization

1. Switch to the tab control "***Teach-in***".
2. Please set here the geometry, size and position of your binarization window. In this area, your image is converted into a black-and-white image (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow
Parameterizing and Testing

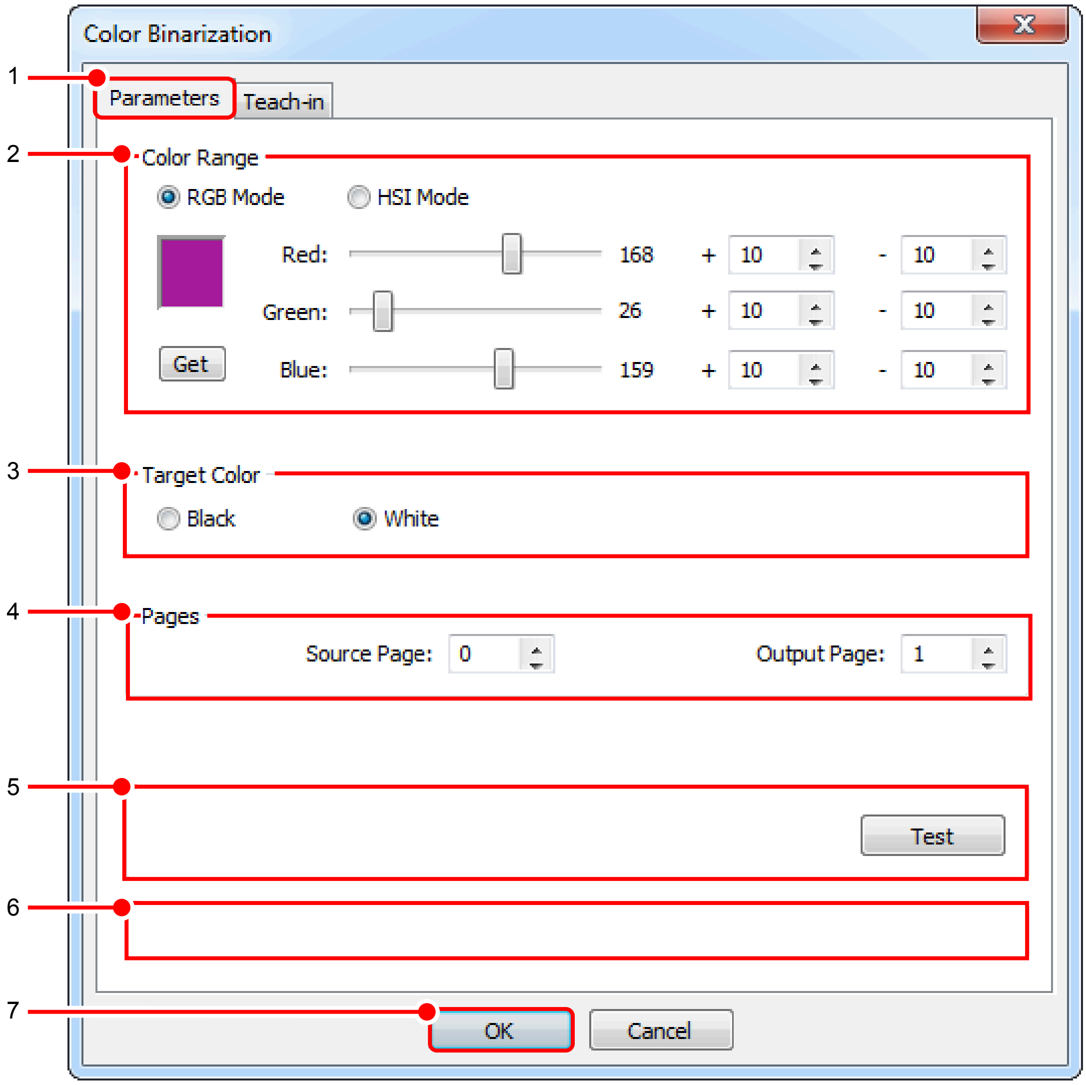


Image 53: Dialog Color Binarization, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Please set in the area "Color Range" the parameters for color range, nominal color and color deviation.

Parameter	Description
RGB Mode/ HSI Mode	Selection whether in the RGB and HSI color space should be checked. Use the HSI mode if you want to check the hue or the saturation irrespective of the brightness.
Slider Red, Green, Blue	If the selection "RGB Mode" is active. Values of the corresponding color channel.
Slider H, S, I	If the selection "HSI Mode" is active. Values for color (Hue), saturation and intensity.
Tolerance	Minimum and maximum tolerances of the respective color channel.
[Get]	The average color of the test window is taken as the default for the target color.

ADVICE

Using the button [Get]

For best results, go to the tab control "***Teach-in***", here reduce the binarization window and place it in a unique color range. Then, go to the tab control "***Parameters***" and click the button [Get] to get the values of the color the of the binarization window.

Then adjust the test window as usual.

3. In the area "Target Color", determine in which binary color the chosen target color is shown.

Parameters	Description
black	After binarization, objects with the target color will be displayed black (gray value 0), the rest is white.
white	After binarization, objects with the target color will be displayed white (gray value 255), the rest is black.

4. In the area "Pages" determine the source and the output page.

Parameters	Description
Source Page	The image page from which the image is taken for binarization.
Output Page	The image page on which the binarized image is written.

ADVICE

The setting **-1** means that the image page that is currently in use, will be overwritten. You can specify the image page with the command **Display** (see also: *"Display", page 147*).

For optimum performance, the target page should be different from the source page. Otherwise, in addition the content of the source page is swapped from the vision system. This operation requires more time.

5. Testing and setting the appropriate parameters
By activating the [Test] button the command with the set parameters is executed. On the source page the corresponding binarized areas are shown in red. On the output page the binarization is shown in the chosen colors. If necessary, change the settings in step 2 until the result meets your requirements.
6. Evaluation Area
Eventually error messages are shown here.
7. Use the [OK] button to insert the command into the test program.

6.1.11 Color Conversion

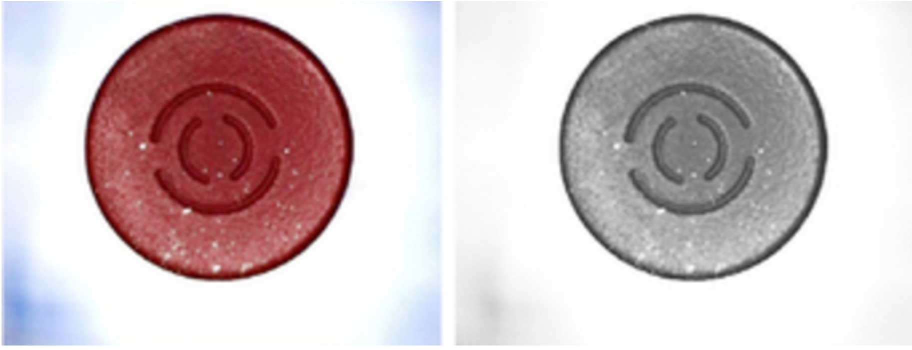


Image 54: Color Conversion before (left) and after (right)

With **Image > Color Conversion** you convert an area of a color image into a gray scale image. Use this command for colored objects, in order to process them with classical detection procedures. The areas that should be converted must contain similar color properties in RGB or HSI format. The resulting gray scale image contains only black pixels (gray value 0) and white pixels (gray value 255).

Work Flow
Define Area of Conversion

1. Switch to the tab control "***Teach-in***".
2. Please set here the geometry, size and position of your conversion window. In this area, your image is converted into a gray scale image (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow

Parameterizing and Testing

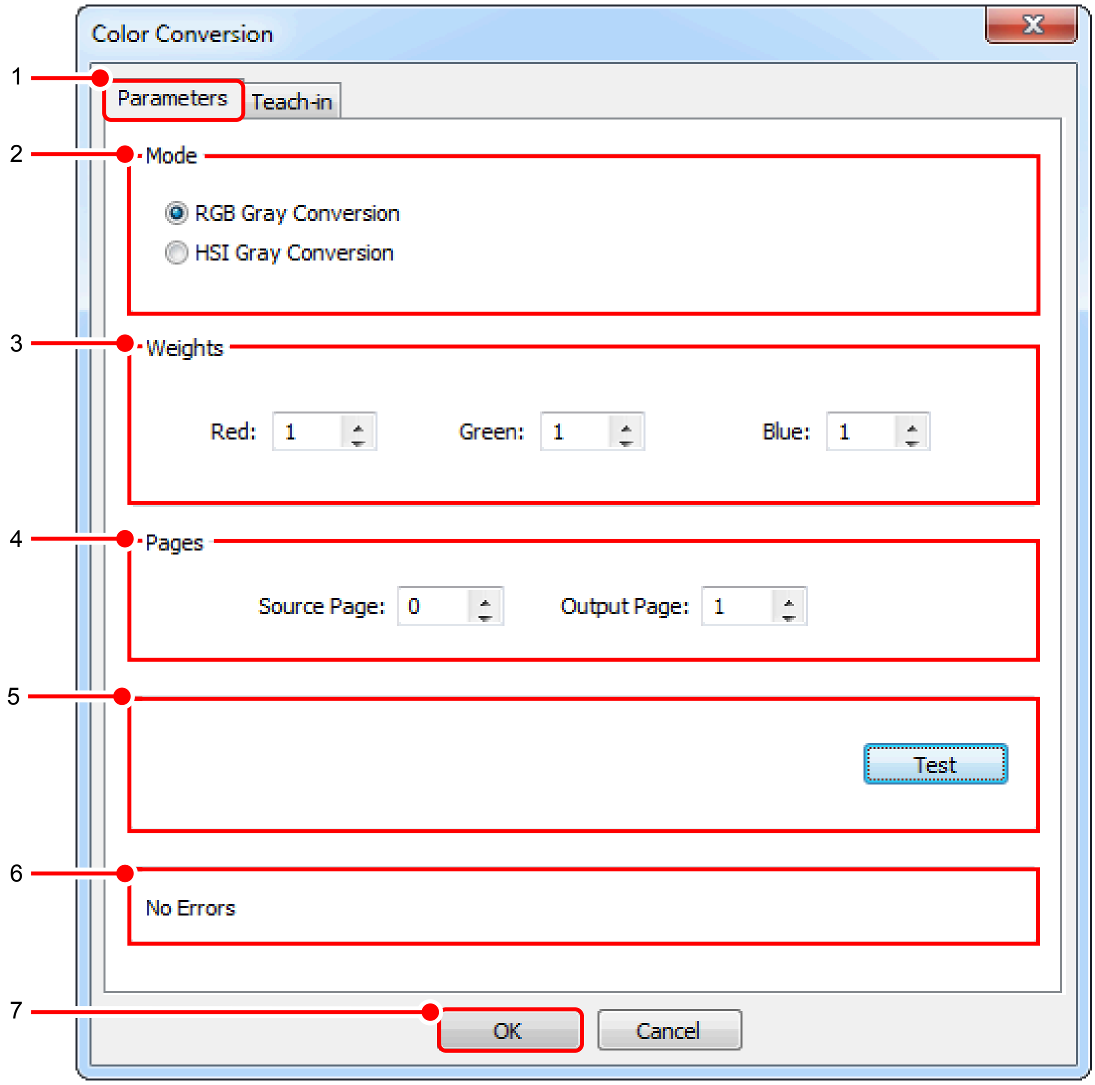


Image 55: Dialog Color Conversion, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Please set in the area "Mode" the parameters for color space to be used.

Parameter	Description
RGB- Gray Conversion	The channels red, green and blue are summed by their weight.
HSI- Gray Conversion	The channels hue, saturation and intensity are summed by their weight. Use the HSI mode if you want to check the hue or the saturation irre-spective of the brightness.

3. Set in the area "Weights" how strongly the individual channels are factored in the conversion.
   For all weights, the sum is formed. Each channel is emphasized so strongly, as his proportion to the sum.

Parameter	Description
Red, Green, Blue	If the selection "RGB Mode" is active. Weighting parameters for the channels red, green and blue.
H, S, I	If the selection "HSI Mode" is active. Weighting parameters for the channels of hue, saturation and intensity.

4. In the area "Pages" determine the source and the output page.

Parameter	Description
Source Page	The image page from which the image is taken for conversion.
Output Page	The image page on which the converted image is written.

ADVICE

The setting -1 means that the image page that is currently in use, will be overwritten. You can specify the image page with the command Display (see also: *"Display", page 147*).

For optimum performance, the target page should be different from the source page. Otherwise, in addition the content of the source page is swapped from the vision system. This operation requires more time.

5. Testing and setting the appropriate parameters
   By activating the [Test] button the command with the set parameters is executed. On the output page the converted area is shown in gray scales. If necessary, change the settings in step 3 until the result meets your requirements.
6. Evaluation Area
   Eventually error messages are shown here.
7. Use the [OK] button to insert the command into the test program.

6.1.12 GenICam-Register

With the command **Image > GenICam Register** you can get access dirctly to the GenICam register of your vicosys GigE Vision camera.

ADVICE

This command is only for GigE Vision cameras in combination with the vision system vicosys.

Documents to GenICam registers can be obtained from the respective camera manufacturer.

Frequently used settings were applied as pre-defined functions (Presettings).

Only one register of a camera is read or written per command.

Work Flow

Parametrize

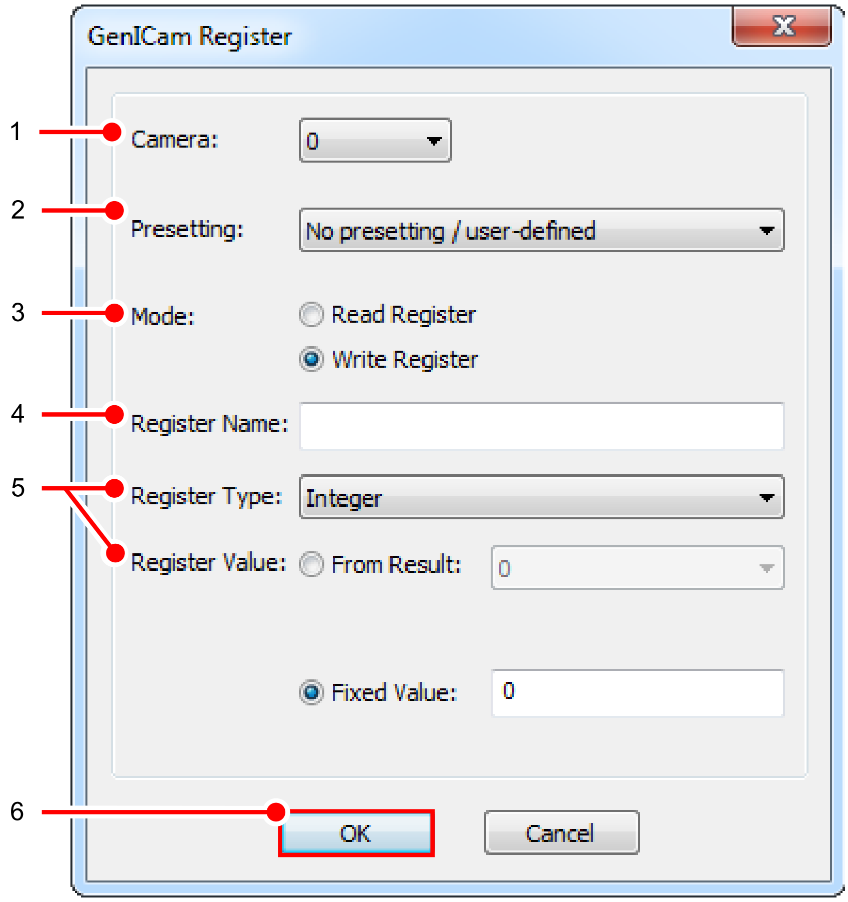


Image 56: Dialog: GenICam Register

1. All cameras connected to vicosys are listed here. Select from the drop-down list the GigE vision camera to be edit.
2. Set here whether you want to manually change a register or you want to use a pre-defined function (Presetting). If you choose a presetting, the following parameters are automatically selected/filled and grayed out.
3. Select here, if the register is to be read or written.
   This Option is only active if you choose "No presetting / user-defined" in step 2.
4. Enter here the name of the register.
   This Option is only active if you choose "No presetting / user-defined" in step 2.
5. Enter here the type and the value of the register.
   The value depends on the choosen register and the access (read/write).

   Read

   - Enter the type of the register (Integer, Float, Boolean, String, Enumeration).
   - Enter the result number or name under which the value should be stored.
   - When using Register type "Float" enter the Multiplier. This determines by witch value the result is multiplied. Example: Multiplier 10 means that the result is multiplied by 10.

   Write

   - Enter the type of the register (Integer, Float, Boolean, String, Enumeration, Command).
   - Select if the value should be taken from a result/string or from a fixed value.

- Specify the appropriate value of the register (e.g.: True / False in Boolean).
- If write a "Float" from Result is choosen, enter the Divisor. This determines by which value the result is divided. Example: Divisor 10 means that the result is divided by 10.

This Option is only active if you choose "No presetting / user-defined" in step 2.

6. Use the [OK] button to insert the command into the test program.

6.1.13 Camera Lighting

ADVICE
This command is specifically for line pictor T.

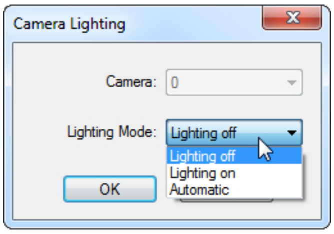


Image 57: Dialogue Camera Lighting

Use **Image > Camera Lighting** to control the internal lighting or the lighting connection of the pictor T devices.

Input Parameters

Parameters	Description
Camera	Here you can select the camera you want to control/set. Devices of the pictor T series have only one camara. For this reason the field is deactivated.
Lighting Mode	Here you can select the mode of the lighting.

Lighting Modes

Mode	Status
Lighting off	The lighting is always off.
Lighting on	The lighting is always on.
Automatic	The lighting is generally off and is only switched on for the period of the image recording (Commands: Crab Image, Synchronous Flash and Video Mode "Live Image"). This modus is recommended.

ADVICE
With a low lighting load the Automatic mode prevents a high thermal load and therefore extends the LEDs' service life and saves energy.

6.1.14 Configure Shutter

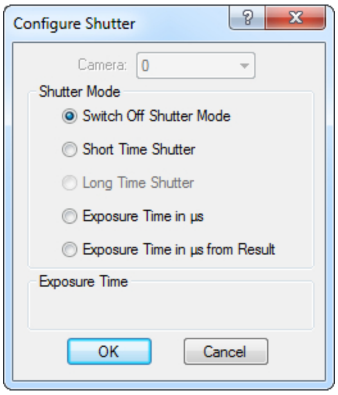


Image 58: Dialogue Configure Shutter

Add a command to set the shutter mode and the exposure time in the program using **Image > Configure Shutter**. The set values will remain valid until you actively change them.

Options

Options	Description
Switch Off Shutter Mode	You set the exposure time to 20 ms with this option.
Short Time Shutter	The exposure lasts for the duration of the image acquisition command. You should use short time shutter for grabbing moving test objects. **Exposure Time** … Reciprocal value of the exposure time in seconds (see the table below). For Machine Vision systems without progressive-scan cameras. The setting is made in the following steps (Intermediate values are rounded internally to the nearest value.)
Long Time Shutter	For vicosys the Long Time Shutter mode is not available. You should use long time shutter mode primarily for grabbing weakly illuminated test objects. The image acquisition takes place over several image acquisition cycles. **Note**: The duration of the image acquisition is determined by the exposure time entered. It is not necessary to enter the image acquisition command several times in the program. **Exposure Time** … Exposure time in quantity of half frames
Exposure Time in µs	The exposure is made with the set exposure time. **Exposure Time** … Exposure Time in µs
Exposure Time from Result	The exposure is made with the exposure time read from the result structure. **Result No.** … Result, which contains the exposure time.

Input Value	Exposure Time
10000	1/10040 s
4000	1/4394 s
2000	1/2068 s
1000	1/1005 s
500	1/495 s
250	1/250 s
125	1/125 s
100	1/120 s

Exposure time for Short Time Shutter without progressive-scan sensors

6.1.15 Synchronous Flash

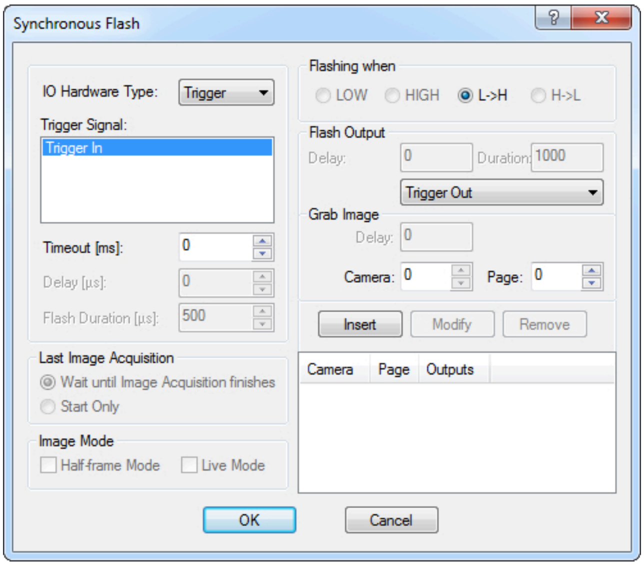


Image 59: Dialogue Synchronous Flash

Enter a trigger-controlled image acquisition command with **Image > Synchronous Flash**. This command contains the flash signal output and the synchronization between trigger, flash, and image acquisition.

If you use the digital I/Os for receiving the trigger impulse, for the flash impulse output as well as for the start of image acquisition, you can set the delay from trigger to image acquisition as well as from image acquisition to flash.

Parameters

- Hardware type
- Digital input
- External trigger signal for flash release
- Camera number, acquisition page and output of camera
- Image acquisition signal
- Flash signal

Time Sequence

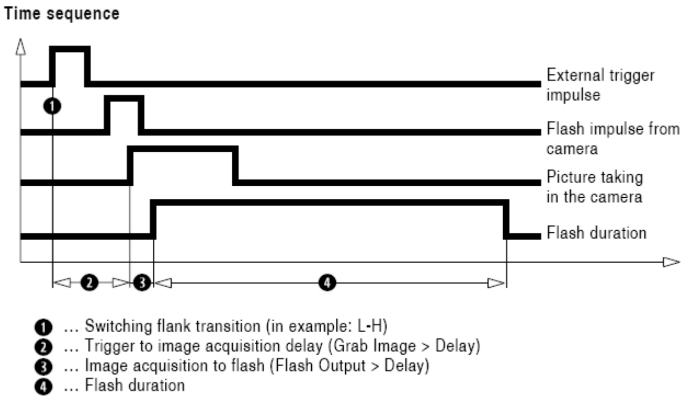


Image 60: Synchronous Flash Time Sequence

Advantages of Flash as Compared to Image Acquisition with Shutter and Continuous Lighting

- The large amount of light, together with control of the flash duration, means shorter image integration times and hence better acquisition of moving objects.
- The amount of light required per time unit is lower compared to continuous lighting.
- The higher light intensity allows an increased depth of field with flash via stopping-down of the image acquisition optics.
- Better controllability.

Hardware Types - Trigger

Type	Description
Trigger	Fast internal TTL trigger input/output (hardware trigger) of the camera. • Input: image acquisition trigger • Output: Flash trigger, flash duration corresponds to the exposure time of the camera • vicosys: For Hardware Type "Trigger", use the hardware trigger on the corresponding camera. As a flash trigger, use the output on the corresponding camera (model-specific). • When using the camera hardware trigger, setting the trigger-to-image acquisition delay as well as the image acquisition-to-flash delay is not possible.
sercos	The trigger is activated via the sercos III fielbus. No additional output is necessary.
CANopen	The trigger is activated via the CANopen fielbus. No additional output is necessary.
None	The image acquisition is started immediately by a "SoftwareTrigger" when the command is executed. The "TriggerOut" of the camera is used as the output signal.

Hardware Types - IO

ADVICE
Please contact your supplier if you require support for other I/O hardware types.

Hardwaretype	Description
PS8	4 digital input, 6 digital output (lines can be configured individually as inputs or outputs)
DIO2/4	2 digital input, 4 digital output
DIO4/6	4 digital input, 6 digital output
DIO4	4 digital input, 4 digital output
DIO8	8 digital input, 8 digital output
DIO16	16 digital input, 16 digital output

Input Parameters for Trigger Signal

Parameters	Description
Trigger Signal	Designation of the input over which the trigger signal is sent.
Timeout [ms]	Time the system will wait for a trigger signal given in ms. If this time elapses without a signal being recognized, the command is aborted without an image being acquired. Set a timeout of 0 if the command should wait indefinitely for a signal.
Delay [µs]	Delay between the trigger signal and output of the first flash signal in the list given in µs. The delay can be used if, e.g., a delay time is needed for positioning the object under the camera after the trigger signal has been generated by a light barrier.
Flash Duration [µs]	Output duration of the flash signal for given in µs. Determines the output level duration of the lighting. (The **Flash Output > Delay** field is grayed out.)

Input Parameters for Flashing when

Parameters	Description
(Low, High, L->H, H->L)	Status or edge transition of the trigger input which releases the flash.

Output Parameters for Flash Output

Parameters	Description
Delay	This value is given in µs. Image acquisition-to-flash delay. • Only active when trigger, flash and image acquisition are controlled via digital I/Os. • Not valid when controlled via the camera hardware trigger. • Input is required for the use of flash lighting. The delay is a camera constant but cannot be neglected. • Set the value to zero for static exposures.
Duration	Flash duration for vicosys in µs • Can be individually set for each lighting unit of each connected camera. • Determines the output level duration of the lighting. (The **Trigger Signal > Flash Duration** field is grayed out.)
Output	Camera output over which the flash signal is saved.

Output Parameters for Grab Image

Parameters	Description
Delay	This value is given in µs. Trigger-to-image acquisition delay • Only active when trigger, flash and image acquisition are controlled via digital I/Os. • Not valid when controlled via the camera hardware trigger.
Camera	Selection of the Camera.
Page	Image memory page onto which the image is sent. If you wish to configure several image acquisitions for the same camera, use a separate image acquisition page for every acquisition.

To ensure reliable synchronization of flash and exposure

1. First set a very long shutter time (e.g. 20000 µs) and a shorter flash delay (e.g. 12000 µs).
2. Shorten the set times in order to achieve shorter process times.
   How much the times can be shortened depends upon the delay time of the camera and the delay time of the relevant lighting unit.

To edit the image acquisition list

Enter several sequential image acquisitions in the list if necessary.

1. Specify the output, camera and acquisition page in the drop-down menu input fields.
2. Edit the entries in the list with the [Insert], [Edit] and [Clear] buttons.

Tuning Parameters

The Tuning parameters enable program time to be saved.

ADVICE

The parameters »Last Image Acquisition« and “Half-frame Mode and Live Image Mode” are supported by pictor only.

The options for last image acquisition are only active with progressive-scan camera types: e. g. pictor M1208 … M1856. For older systems (e. g. pictor 1004, 1005, 1006), the program always waits until the image acquisition has finished.

Tuning parameters »Last Image Acquisition«	Description
Wait until Image Acquisition finishes	After the flash signal has been released, the program waits until the image has been completely acquired in the image memory.
Start Only	After the flash signal has been released, the program immediately goes on to the next command in the test program. Advantage: further commands can be processed during the image acquisition.

“Half-frame Mode and Live Image Mode” Tuning Parameters	Description
The options are active if the functions are supported by the connected vision system. Note: The camera must always be calibrated in the mode in which the image was acquired. The calibration should be made immediately after the image acquisition.	
Half-frame Mode	In order to save time, only every second image line is acquired. The acquired image has only half the resolution. This option is not released for progressive-scan cameras (pictor M1208 … M1856) – for which a full image acquisition is always performed.

ADVICE

The mode Area Of Interest can be used by users of multi-camera systems with the command "Synchronous flash". The AOI is seperately stored for each camera and is always adopted from the last grab image command. The Area Of Interest can be set by Image > Grab Image. *see "Grab Image", page 138*

6.1.16 Video Mode

Insert a command into the test program to switch between live image and memory image using **Image > Video Mode**.

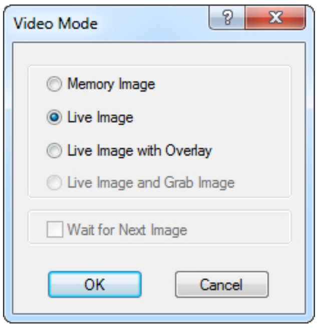
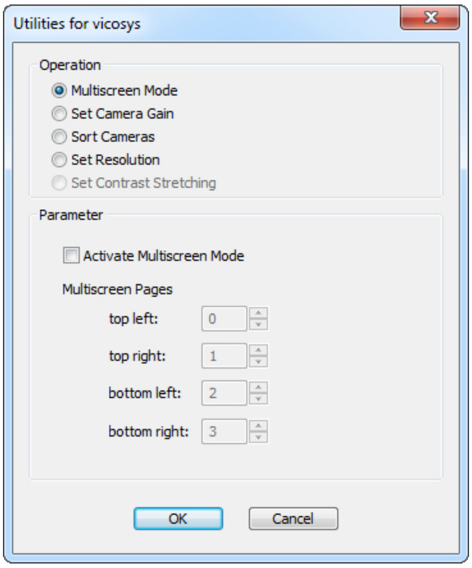


Image 61: Dialogue Video Mode

Options for Vision Systems

Options for pictor	Description
Memory Image	Displays the content of the image memory on the monitor.
Live Image	Displays the in realtime received live image. (This is the fastest mode with the pictor M system.)
Live Image + Overlay	Displays the live image from the camera and the graphic from the overlay memory on the video screen.
Live Image + Grab	Displays the live image from the camera and the graphic from the overlay memory on the video screen and, at the same time, acquires an image in the image memory.
Wait for Next Image	An expert mode for setting the switching of video modes. Use this option as follows: • Activate this option in normal cases. The program waits to change video mode and does not execute further commands until the last image acquired has been loaded into the image memory and processed. This introduces a short delay, but the program is processed more rapidly afterwards and no further resources are tied up with processing the image. • Deactivate the option, if the following commands have to be executed within a definite time frame or if an older interrupt-controlled sensor is being used.

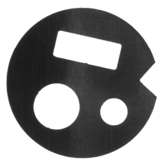
6.1.17 Utilities for vicosys



Using **Image > Utilities for vicosys**, vcwin provides the following functions for use with the vicosys multi-camera system:

- Multiscreen Mode
- Set Camera Gain
- Sort Cameras
- Set Resolution
- Set Contrast Stretching

Multiscreen Mode

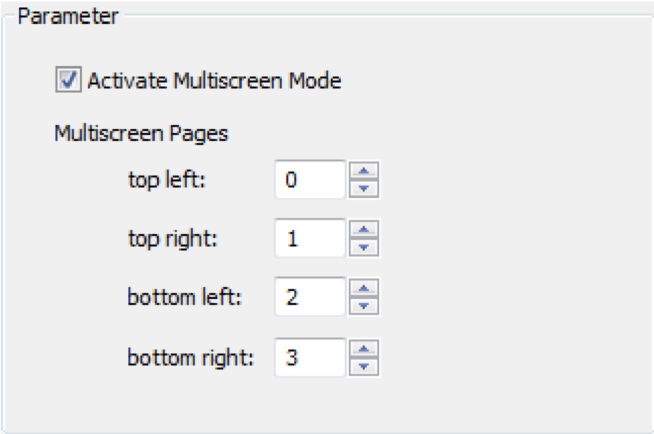
Multiscreen Pages top left: 	Multiscreen Pages top right: 
Multiscreen Pages bottom left: 	Multiscreen Pages bottom right: 

Use **Multiscreen Mode** to display four image pages on the monitor simultaneously.

Use this function e.g. in the following scenarios:

- To display the images from several cameras on the monitor
- Display the nominal image or the live image and use the other three image memory pages for continuous display of particular measurement values

Input Parameters



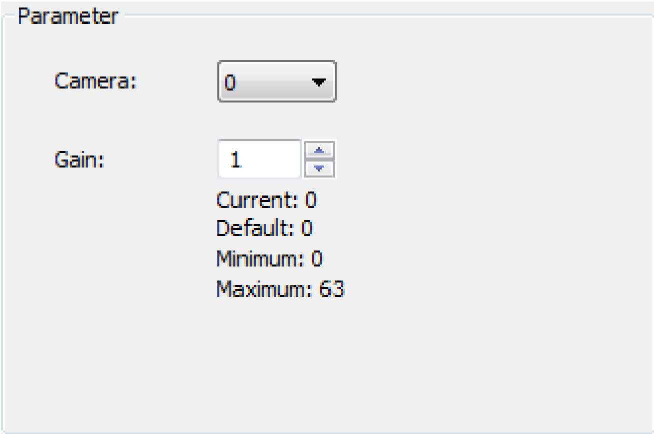
Input Parameters	Description
Activate Multiscreen Mode	Option active: vicosys switches to the multiscreen mode. Option inactive: vicosys switches to the one screen mode.
Multiscreen image memory pages > top left	Number of the upper left displayed image memory page.
Multiscreen image memory pages > top right	Number of the upper right displayed image memory page.
Multiscreen image memory pages > bottom left	Number of the bottom left displayed image memory page.
Multiscreen image memory pages > bottom right	Number of the bottom right displayed image memory page.

Setting Camera Gain

ADVICE

Set Camera Gain alters the image brightness without changing the image acquisition time of the camera (shutter). Use this function to shorten the image acquisition time via increased image brightness. The image quality may become lower with increased gain.

Input Parameters



Input Parameters	Description
Camera	Select the camera that the gain should be applied to.
Gain	Gain of the selected camera. The range within which the gain can be adjusted and the default value both depend on the camera type, and are read and displayed automatically.

Sorting Cameras

Sort Cameras changes the allocation of the connected cameras to the camera numbers in vicosys.

ADVICE

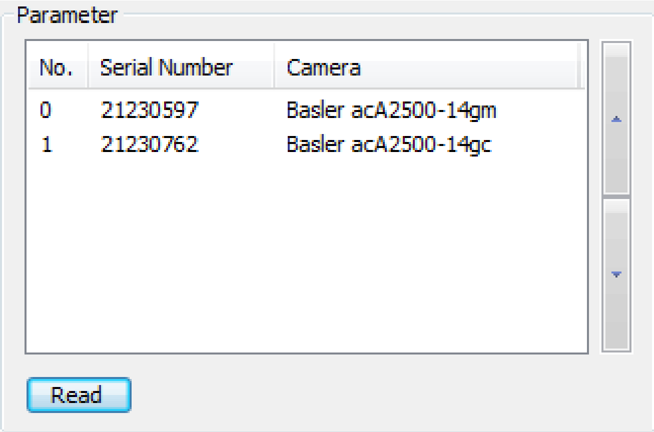
The vicosys sorts the cameras according to ascending serial number. If a serial number changes (e.g. through exchange), a new sorting is performed.

Use this function to restore the original sequence after exchanging a camera. For changing the sequence of the cameras click on the camera number afterwards click on the arrow buttons to the right for upwards and downwards.

Use

- Insert the command into the test program in order to modify the camera list.
- If the quantity of cameras registered in the camera list and the relevant serial numbers on the cameras do not correspond to the cameras currently connected, an error message appears on the screen of the vicosys.
- To read the camera list from the vicosys, click on the [Read] button.

Parameters



Parameters	Description
No.	Sorting number of the camera
Serial Number	Serial number of the camera
Camera	Designation and type of the camera

Set Resolution

Use **Image > Utilities for vicosys > Set Resolution** to change the monitor resolution of the monitor for vicosys .

Use this command to accelerate image display by reducing monitor resolution (camera image is scaled 1:1, 1:2, 1:3, …) . On the other hand, the image quality of high-resolution cameras can be improved by increasing the resolution.

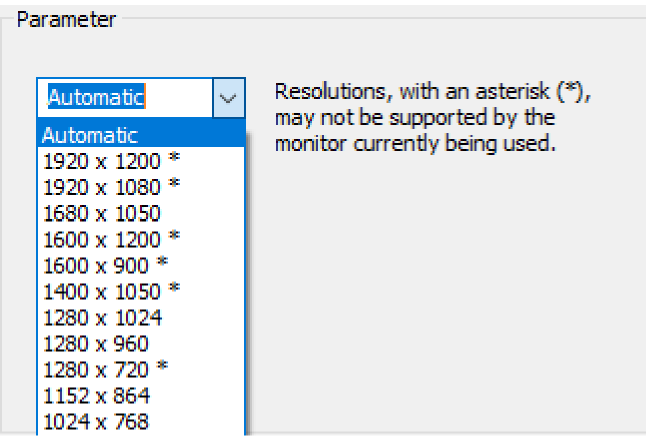
ADVICE

In order to display an image on the monitor, a set scaling principle is used by vicosys. This can result in black bars.

Use

Insert the command in the test program to change the monitor resolution.

Input Parameters



Select the desired monitor resolution from the list in the combo box.

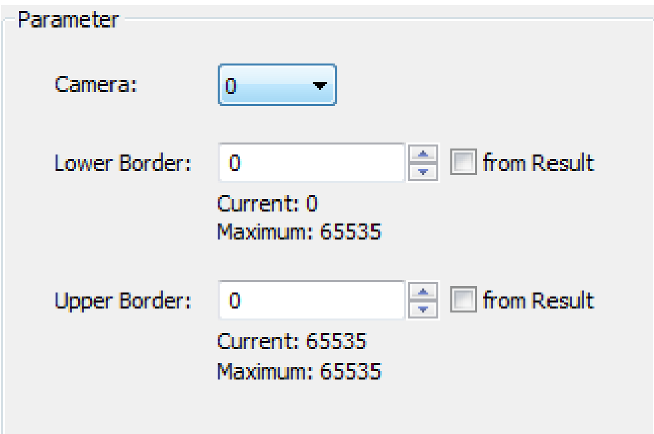
ADVICE
Resolutions, with an asterisk (*), may not be supported by the monitor currently being used.

Input Parameters	Description
Automatic	Recommended resolution of connected monitor.
Reboot	This option can only be used by for versions older then 4.16.200. At newer versions the monitor resolution is changed immediately. Reboots vicosys standalone. A warning appears on the monitor before the reboot. After the reboot, the monitor resolution has been changed. It is possible to interrupt the reboot process through initialization. This is used to be able to interrupt continuous rebooting caused by two contradictory resolution settings in the start program.

Set Contrast Stretching

Use **Image > Utilities for vicosys > Set Contrast Stretching** to automatically stretches the grey values of an image.

ADVICE
Use of the e**Set Contrast Stretching** feature is based on a licence model. This option is only available for thermo-vision cameras.



Parameters	Description
Camera	Here are all cameras displayed that deliver imagedata greater than 8 bit. Select the camera on which you want to apply the contrast stretching.
Lower Border	Enter the lower border here. This value is set to the grey level "0". You can also use a value from a result.
Upper Border	Enter the upper border here. This value is set to the grey level "255". You can also use a value from a result.

A 12-bit camara provides 4095 as an upper border. If you choose 500 as lower border and 2000 as upper border, all values between 500 and 2000 are scaled to 8 bit (0-255).

6.1.18 Linescan Camera

ADVICE
In principle, you can connect linescan cameras at the same time to the vicosys as any other camera types. However, only one line scan camera can be connected to a FireWire controller.

ADVICE
Only pictor M41EL: Perform a shading correction before using this command (*see "Shading Correction", page 99*).

Use the command **Image > Linescan Camera** to integrate lines from a linescan camera to an image and to store it in an image page.

Dialogue structure

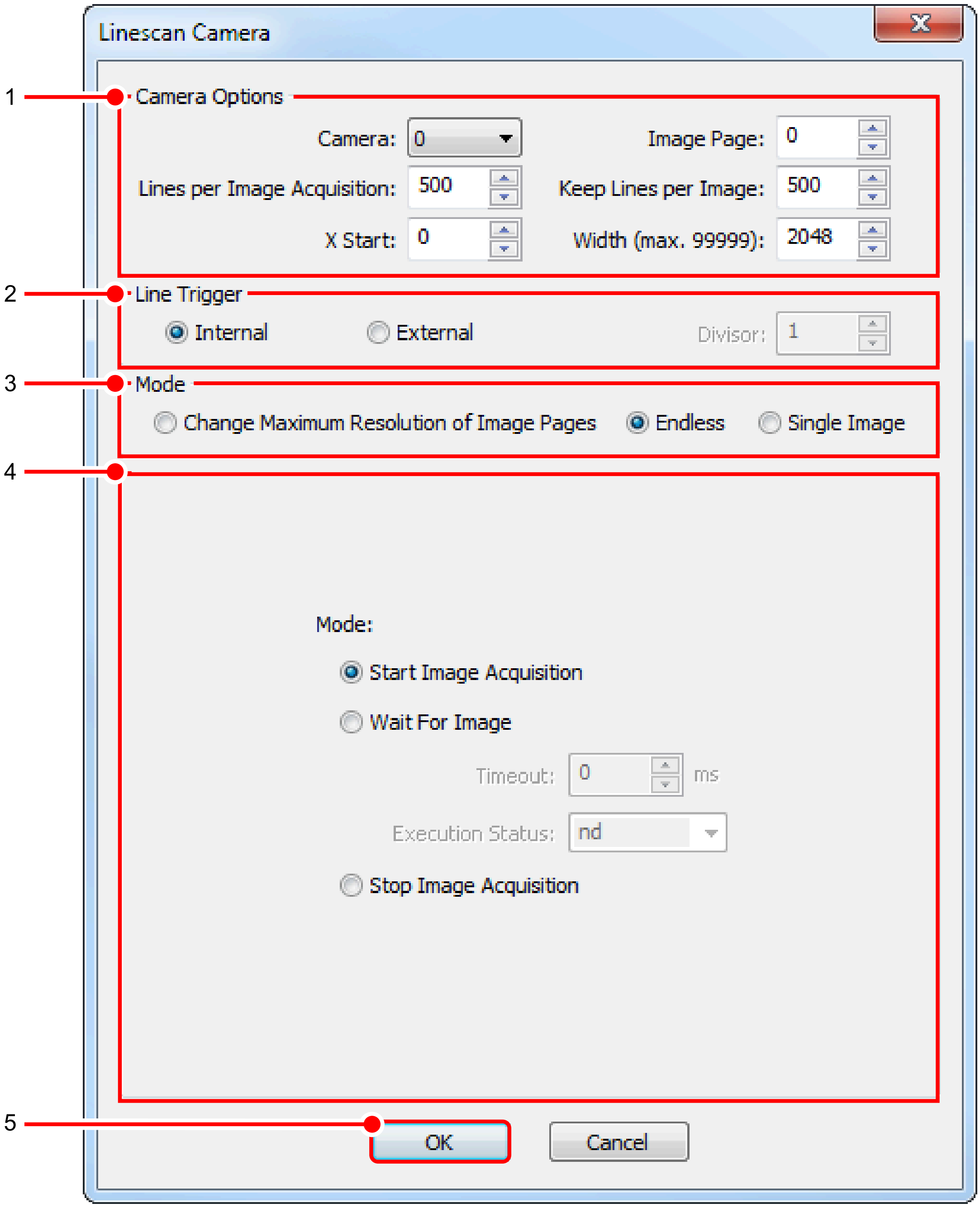


Image 62: Dialogue Linescan Camera

Work Flow Parametrize Camera

1. Settings for the connected linescan camera.

ADVICE
In Mode Endless, the image of the image page has a total height of *Lines per Image Acquisition* + *Keep Lines per Image.* In Mode Single Image, the image of the image page has a total height of *Lines per Image Acquisition*. The total height of the image must not be larger than the *Maximum number of lines* in Mode *Change Maximum Resolution of Image Pages.*

Parameter	Description
Camera	Number of the linescan camera that is configured. Default values of the camera are transferred at the first time selection.
Image Page	Number of the image page on which the recorded linescan image is stored.
Lines per Image Acquisition	Mode Endless: Number of lines that are acquired and attached below the image. Mode Single Image: Number of lines that are merged into a image.
Keep Lines per Image (only Mode Endless)	Number of lines that will kept in the taken image. The image content is pushed up and the new lines are hung down.
X Start	X position of the starting pixel of the line. The display is from that pixel.
Width (max. 4096)	Number of pixels to be read out from a line, starting at X Start. The value after *max*. results from the number of pixels of a line and depends on the camera.

Work Flow Parametrize Line Trigger

2. Settings for the triggering of the image acquisition.

Parameter	Description
Internal	The camera runs based on time as a function of the shutter value with the maximum possible line rate. The shutter value can be set with **Image > Configure Shutter**.
External	The linescan trigger input of the camera is used.
Divisor (only External)	The divisor indicates in which trigger process a linescan image is taken. **Divisor** • 1 = Each trigger process starts an image acquisition. • 2 = Each second trigger process starts an image acquisition. • 3 = Each third trigger process starts an image acquisition. • etc.

Work Flow Mode

3. Select the mode of the camera.

   Modes

   ° Change Maximum Resolution of Image Page (Change Resolution)
   ° Endless
   ° Single Image

4. Here define the settings for the camera mode.

Mode Change Resolution

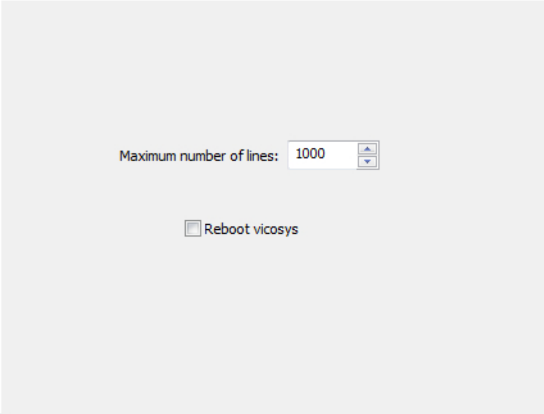


Image 63: Dialogue Linescan Camera: configuration window Change Maximum Resolution of Image Pages

Change Maximum Resolution of Image Pages

- The maximum width of the image page corresponds to the maximum width of the connected cameras.
- To change the maximum height, enter the new height of the image pages under *Maximum number of lines*. The new maximum number of lines is assumed only after the system restarts.
- Reboot vision system.

  vicosys:

  - Activate the checkbox *Reboot vicosys*.
  - The device will reboot if the set line number in the command does not correspond to the set line number in vicosys.

  pictor M41EL:

  - Execute the command with [OK].
  - Go to the "**System Preferences** menu and start the command **Save File System in Flash**.
  - Start the pictor manually (interrupt the operating voltage supply).

Mode Endless

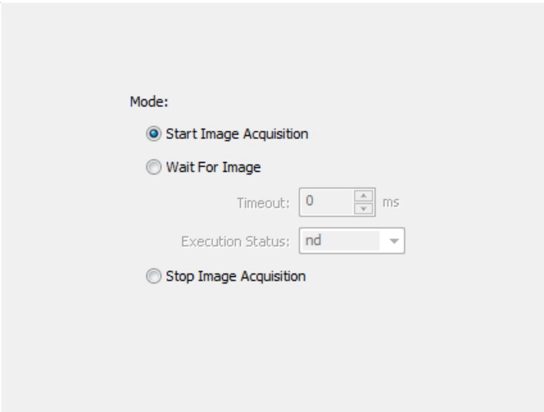


Image 64: Dialogue Linescan Camera, configuration window for Mode Endless

- All-over detection of image lines. (E.g.: for infinite materials processing).
- For the mode **Endless**, the command must be inserted twice in the test program.

- The command with the option **Start Image Acquisition**, must be called once at the start of the program. With this Option you start the endless mode. Now linescans are acquired continuously in the background. In parallel, the test program is further executed.
- The command with the option **Wait For Image** must be in a loop. **Start Image Acquisition** stores the lines in the camera. If the specified number (*Lines per Image Acquisition*) is reached, the image is stored by the vision system in the defined Image Page. Then test functions can be performed on the image.

Parameter	Description
Timeout	Waiting time for the image before the command is terminated without an image acquisition. A timeout of 0 means infinite waiting.
Execution Status (only pictor M41EL)	Result number for status and error messages of the command.

- The command with option **Stop Image Acquisition** stops the endless acquisition.

ADVICE

For vicosys:

If more lines are recorded by the camera than can be processed by the vision system, an error message is displayed on the monitor output (VGA or DVI) of the vision system on the screen.

For pictor M41EL:

If more lines are recorded by the camera than can be processed by the vision system, the vision system writes 4 in the result (*Execution Status*) and runs *Stop Image Acquisition.*

Mode Single Image

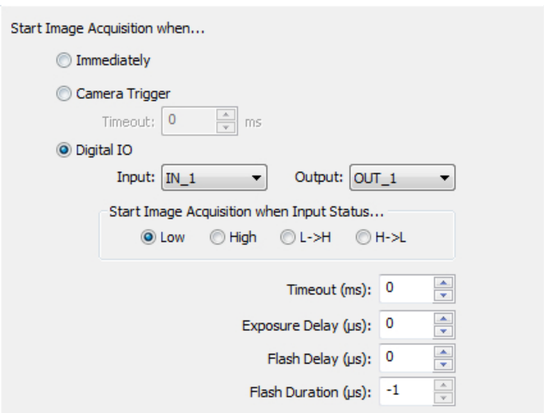


Image 65: Dialogue Linescan Camera, configuration window for Mode Single Image

- In Single Image mode, you integrate a defined number of lines into a image and store them in an image page.
- With the option **Immediately** the image acquisition starts immediately.
- With the option **Camera Trigger** the image acquisition starts with the activation of the trigger input of the camera.

Parameter	Description
Timeout	Waiting time for the image before the command is terminated without an image acquisition.

- With the option **Digital IO** the image acquisition starts with the activation of the adjusted digital input. In parallel, a lighting or similar can be switched for any given time via a digital output (OUT) connectors.

Parameter	Description
Input	Specification of the digital input for the trigger signal for imaging acquisition.
Output	Specification of the digital output for the trigger signal for lighting control or others.
Start Image Acquisition when Input Status...	State or edge transition of the trigger input for starting image acquisition.
Timeout (ms)	Waiting time for the image before the command is terminated without an image acquisition.
Exposure Delay (µs)	Delay between trigger signal and image acquisition. Use the Exposure Delay, if for example after the generation of the trigger signal for a flash of light you have to wait until the light reaches its maximum.
Flash Delay (µs)	Delay between trigger signal and the flash signal. Use the Flash Delay, if for example you have to wait a moment after the generation of the trigger signal by triggering a light barrier until the object is positioned under the vision system.
Flash Duration (µs)	Indication of the duration of the output level.

5. Use the [OK] button to insert the command into the test program.

Status and error messages

0 = No Errors

- The command was successfully executed.

1 = Timeout

- The timeout in Endless mode with *Wait For Image* was achieved.
- The timeout in Single Image mode with *Camera Trigger* or *Digital IO* was achieved.
- Increase the timeout for the image acquisition. Enter a timeout of 0 if the command has to wait endlessly.

2 = Invalid Parameter

- At least one parameter is invalid.
- Check the settings of the command. Check that the used mode of the command is supported by the vision system used.

3 = Selected camera is not a linescan camera

- In the dialogue under point 1 Camera Options -> Camera, select a linescan camera.

4 = Error during image acquisition

- The processing time of the image processing functions in the loop takes longer than the recording time of the required quantity of lines (Lines per Image Acquisition).
- Optimize the processing of the image processing functions.

5 = Tried to capture a endless image while not in endless mode

- In Endless mode, the option *Wait For Image* was performed, without starting before *Start Image Acquisition*.
- Enter the command with mode Endless and the option *Start Image Acquisition* at the start of the program.

6 = Error while initializing camera

- Check whether the line camera is connected correctly and is supplied with operating voltage.

6.2 Locate Commands

6.2.1 360° Pattern Search

Overview

Image 66: 360° Pattern Search - taught-in pattern

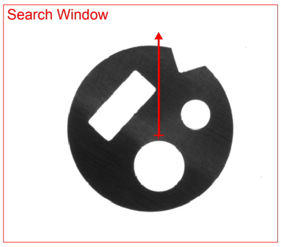


Image 67: 360° Pattern Search - taught-in pattern is detected

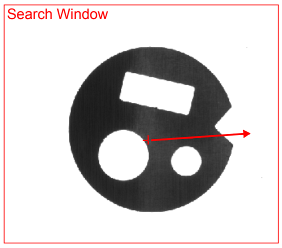
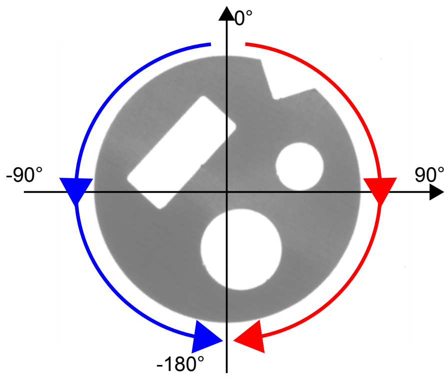


Image 68: 360° Pattern Search - rotated pattern is detected

With **Locate > 360° Pattern Search**, you can locate objects in the image.

In addition to the position, the angular position is determined in relation to the initial taught-in pattern. The command autonomously determines the angular position in relation to the pattern. Searching takes place in multiple phases, first as a rough search and then as a detailed search in a higher resolution. Use the command to find and classify one or more rotated objects.



ADVICE

There are only **nominal number + positive tolerance + 1** objects searched and stored, even if more objects that match the pattern are available.

Work Flow: Teach-in Pattern

1. Switch to the tab control "***Teach-in Pattern***".
2. Determine, by means of the geometry rectangle, which image content has to be read as the pattern. (Information for teaching, see also: *"Teaching in Test, Detection and Search windows", page 26*)
3. With the button [Read Pattern] you read-in all image information located in the geometry as the pattern, and stored it internally in the command.
4. In the reference window, verify if the pattern is complete.

ADVICE

For the 360° pattern search you can also use patterns that are stored on the vision system. These must be stored by using the command Locate > Save Pattern (*see "Save Pattern", page 235*). In this case, no pattern must be learned. The pattern can be selected in the tab control "Parameters".

Work Flow: Define Search Area

1. Switch to the tab control "Search Window".
2. Please set the geometry, size and position of your search window. In this area, your pattern is searched (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow: Parameterizing and Testing

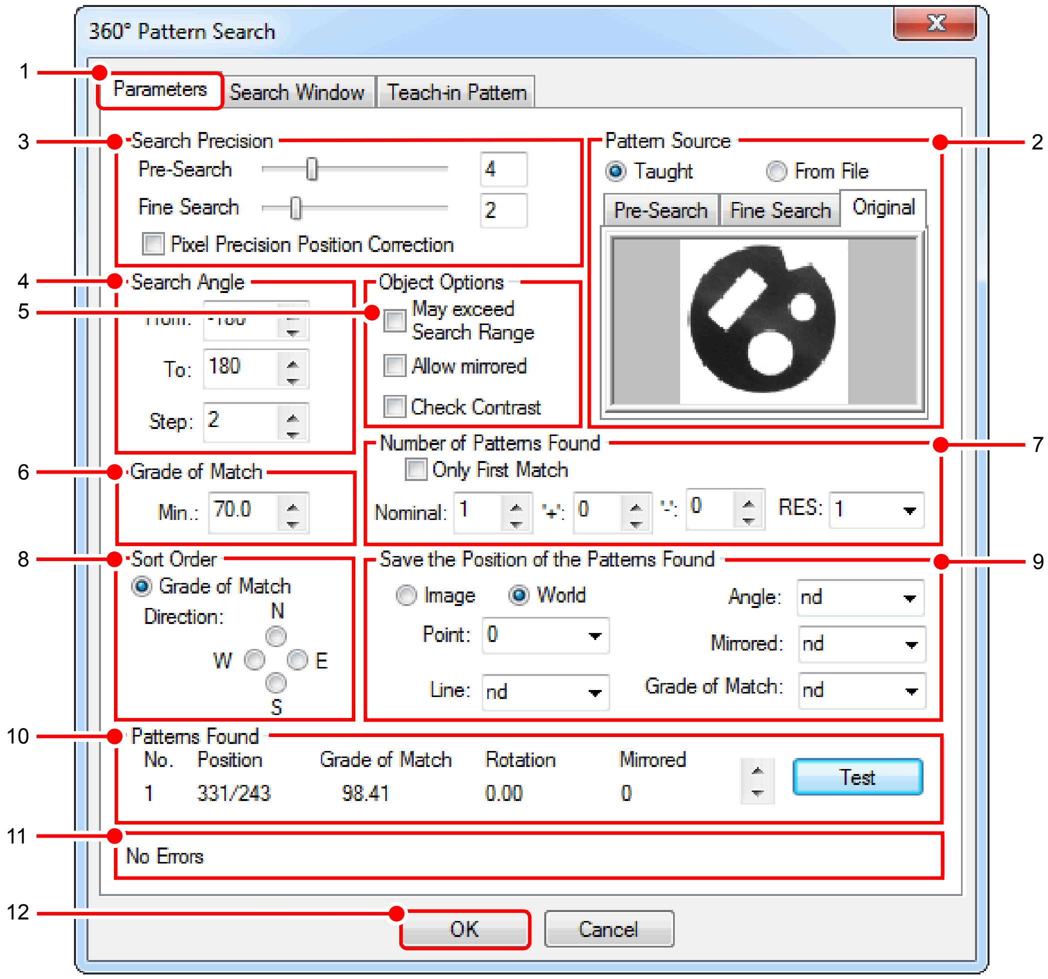


Image 69: Dialog 360° Pattern Search, tab control Parameter

1. Switch to the tab control "***Parameters***".
2. Determine in the area "**Pattern Source**" whether a pattern should be taken from a file or the taught-in pattern.
3. Here you specify the accuracy of the search method using a sliding controller.

   The finer the accuracy is set, the better fine-structured objects or patterns will be found. A coarser search by comparison requires less processing time.

Find the optimum setting by trying out different levels of fineness and observe the corresponding resolutions in the dialog to the right in the area "**Pattern Source**" tab **Pre-Search/Fine Search/Original**. The search accuracy setting should also be verified with different rotation positions of the pattern.

The search algorithm begins its calculations on a very coarse image and tracks the pattern found up to here up to a certain point in a somewhat finer image. With a very fine setting the position found is then pixel-accurate; with a coarser one the corresponding position can, however, only be determined in specific increments. The processing time, however, is significantly reduced because of the lower data volume.

Parameter	Description
Pre-Search	Specifies the degree of precision for the first search phase in the search window.
Fine Search	Specifies the degree of precision when searching for the pattern in the second search phase.
Pixel Precision Position Correc-tion	If you activate this checkbox, with a successful pattern search a pixel-accurate search of the respective x/y position is performed. You receive a precise position value for the object found.

4. Configure search angle and angle range

   You can provide the angle range here in which the pattern is to be found. This means, for example, that an object is only detected in a specific rotation position or it can only be fed into the test in a specific test process-conditional rotation position. As all possible positions and angles of rotation of the pattern are searched for in the image with the pattern search, the amount of possible combinations plays a decisive role for the processing time. With twice as big a search angle range, you therefore generally get twice the processing time. Furthermore with circular- symmetrical objects the pattern must also be found in a complete 360° search, which the following example explains.

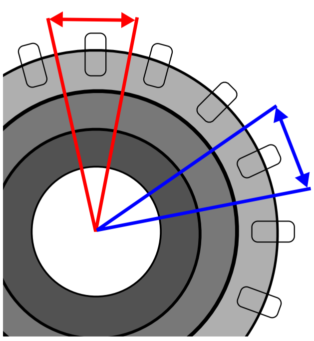


Image 70: Angle Range of a symmetrical object

The red angle range is symmetrical to the blue angle range. As the object consists of 16 such areas, a search must now only be made in the angle range 360°/16 = 22.5°. The processing time decreases significantly.

Parameter	Description
From: - To:	Specifies the angle search area in which the pattern is searched for.
Step	Specifies the angle step increment with which the search steps are to be carried out.

5. Object Options

Parameter	Description
May exceed search range	If you activate this checkbox the object may exceed out of the search area with up to 25% of its area and is still found. This is especially beneficial with the possibility that test objects under certain circumstances are not always within the search area because of position instability or inaccurate provision. This setting, however, means a longer test method processing time.
Allow mirrored	Also mirrored objects are found.
Check contrast	Has a detrimental effect on the level of correlation if the contrast deviates considerably from the contrast in the taught-in pattern.

6. Grade of Match

   You can enter the level of similarity that must be achieved with the pattern comparison here, so that the pattern is marked as found and the tracking was successful. The entry is in percentage and should not be selected too low. Values of more than 75% are proven settings in everyday industrial work.

Parameter	Description
Min.:	Minimum level of correlation, in per cent, for the found pattern to the taught-in pattern in order for a pattern to be processed as a valid pattern.

7. Number of patterns found

   Here you can enter the number of patterns to be found. Only the maximum target number (plus tolerance) + 1 object is sought and found, even if more objects that match the pattern are available.

Parameter	Description
Only first match	Only the first object found, according to the sort sequence, will be evaluated.
Nominal	Number of objects that should be found, so that the command is considered successful.
+/-	Permitted upward and downward deviations from this target value.
RES:	Result number/name in which the number of patterns found is to be saved.

8. Sort Order

Parameter	Description
Grade of Match	Objects are sorted by their grade of match.
Direction	Objects are sorted by their position in the image (according to the specified sequence).

9. Here you can configure the settings for storing the results.

Parameter	Description
Image/World	Saves the pattern’s center of gravity and the angular position line in image coordinates or world units.
Point	From this point, the positions of the objects found will be saved.
Line	From this line the angular position lines found will be saved.
Angle	From this angle the rotation angles found will be saved.
Mirrored	From this result the values for classify a mirrored object will be saved. (0 = not mirrored, 1 = mirrored)
Grade of Match	From this result the grades of match will be saved.

10. Result Patterns Found

    After activating the [Test] button, all patters found are list in the area "Patterns Found" depending on sort. Only the maximum target number (plus tolerance) + 1 object is sought and found, even if more objects that match the pattern are available. In parallel, the patterns found are highlighted in the video image with a red cross as a position and a red arrow (angular position line). Use the arrow keys to scroll through the respective matching pattern.

11. Evaluation Area

    Here you can see the good/bad evaluation of the command, notes for parameterization (tolerance range exceedance) as well as error messages.

12. Use the [OK] button to insert the command into the test program.

6.2.2 Blob Analysis

Overview

A blob (Binary Large Object) is a group of adjacent and contiguous pixels with the same or similar grey values.

Image 71: Blob Analysis - a dark blob found

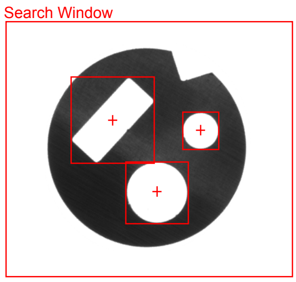


Image 72: Blob Analysis - three bright blobs found

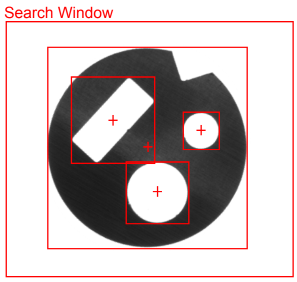


Image 73: Blob Analysis - a dark and three bright blobs found

With **Locate > Blob Analysis** you include a command for finding the center of gravity and area of planar areas of equal gray value within the image. You can save up to 50 the center of gravitys and areas in the points and results structure. Center of gravitys and areas in excess of 50 are not evaluated. You use the command to count objects, to find objects or for position tracking.

Work Flow: Teach-in

1. Switch to the tab control "***Teach-in***".
2. Here you can determine, by means of the geometry rectangle, in which area the image content has to be analysed. (Information for teaching, see also: *"Teaching in Test, Detection and Search windows", page 26*).

3. If necessary, set the parameters (X Point, Y Point) for position tracking (see also: *"Position Tracking of Objects", page 35*)
4. Use the [Test] button to check whether the blobs are found in the detection window with the default parameters.

Workflow: Define Parameters

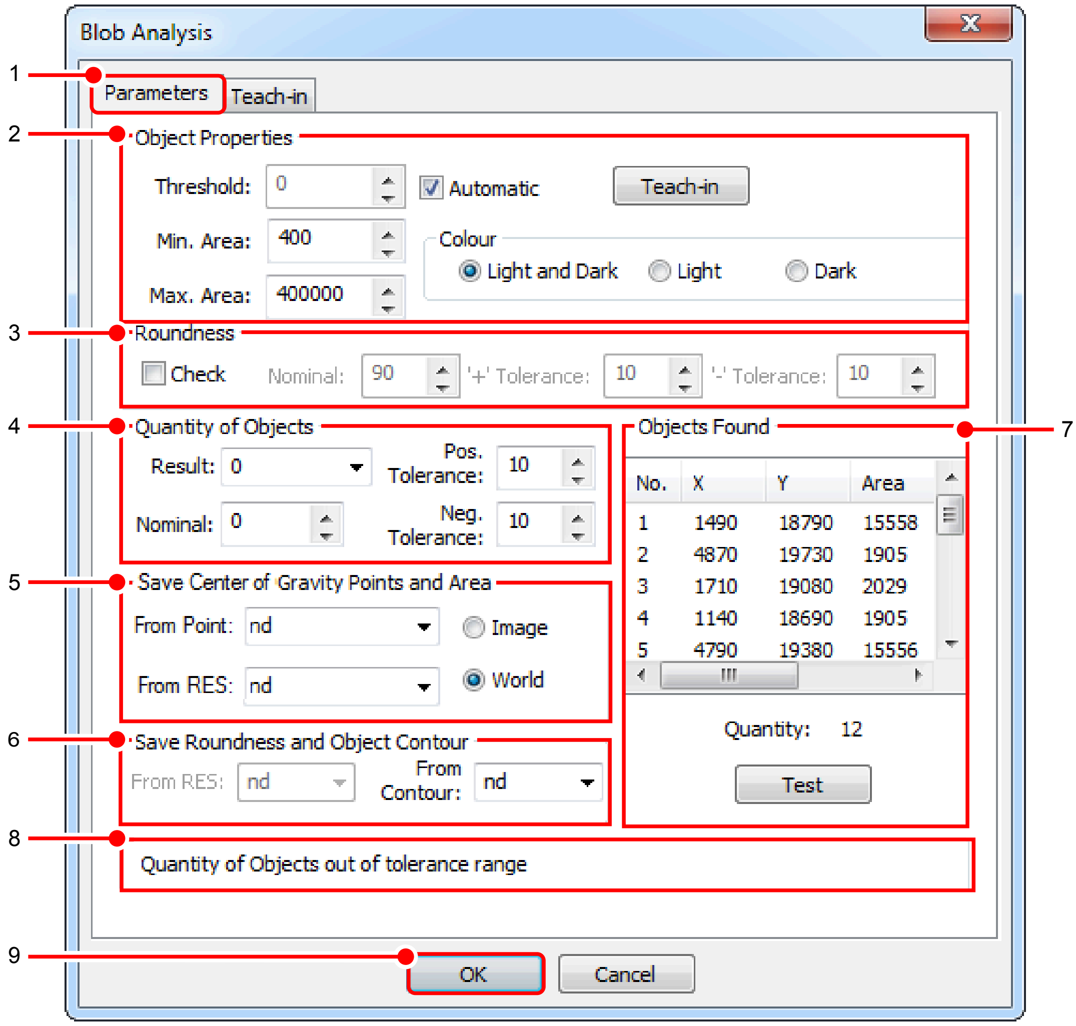


Image 74: Dialog Blob Analysie, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Set in the "Object Properties" the conditions for the threshold of the binary detection and the allowable area (number of pixels) of the objects to be searched.

Parameter	Description
Threshold	Grey value threshold for detection and separation of the objects.
Min./Max. Area	States how many pixels the objects have to encompass in order to be detected. Objects with more or fewer pixels are ignored.
Colour	Light: Only parts of the image will be assigned to the objects that are within of the specified gray value range. Dark: Only parts of the image will be assigned to the objects that are outside of the specified gray value range. Light and Dark: Parts of the image will be assigned to the objects that are within and outside of the specified gray value range.
Automatic	Mark this option in the case of a clear division into bright/dark objects. The threshold is automatically defined during execution.
[Teach-in]	Using the [Teach-in] button you automatically determine the gray value distribution of the threshold, the area, the color and if declared the roundness. The minimal and maximal area is determined from the area of the largest object and the tolerances ±25%. After teaching, you can optimize the parameters and the search window manually.

3. Determine here the parameters for roundness.

Parameter	Description
Check	The Checkbox activates roundness testing for individual blobs.
Nominal	Range, within the roundness of all the blobs must remain. A nominal roundness of 100 is a perfect circle.
Tolerances	Permitted upward and downward deviations from the nominal round-ness.

4. Here you set the nominal quantity of objects to be found. If the number of objects found lies within this range, the command ends with a good result (otherwise with bad).

Parameter	Description
Result	Result number/name for saving the number of objects found.
Nominal	Number of objects that should be found, so that the command is considered successful.
Toleraces	Permitted upward and downward deviations from the nominal number.

5. Save Center of Gravity and Area

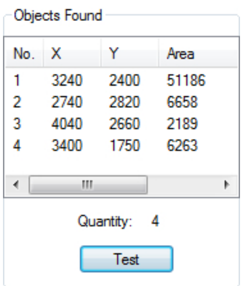
Parameter	Description
From Point	From this point, the centre of gravity of the objects found will be saved.
From Res	From this result in the result structure the area of the objects found will be saved.
Image/World	This radio button determines whether the storage of points and areas are in image or world coordinates.

6. Save Roundness and Object Contour

Parameter	Description
From RES	From this result in the result structure the roundness of the objects found will be saved.
From Contour	From this contour in the contour structure the contour of the objects found will be saved.

7. After activating the [Test] button, all objects and its parameters found are listed in the area "Objects Found". In parallel, the objects found are highlighted in the video image with a red rectangle and a red cross (centre of gravity).
8. Evaluation Area
   Here you can see the good/bad evaluation of the command, notes for parameterization (tolerance range exceedance) as well as error messages.
9. Use the [OK] button to insert the command into the test program.

Result



No.	Number of the current object.
X	X coordinate of the center of gravity of the current object.
Y	Y coordinate of the center of gravity of the current object.
Area	Areas in pixels or world units.
Roundness	Roundness of the current object.
Quantity	Number of objects found.

Example

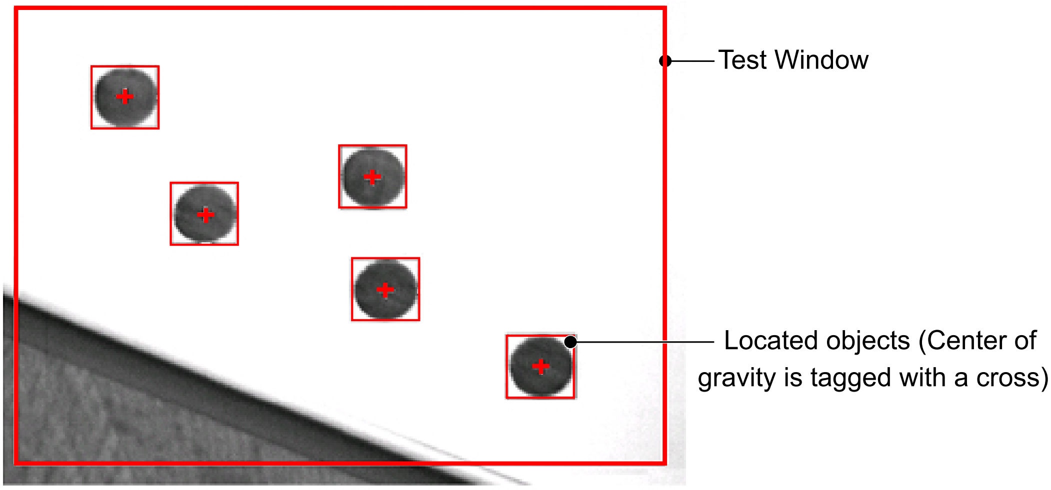


Image 75: Video image: blob analysis with 5 located objects

6.2.3 Data-Matrix Code

General

Vision systems of Vision & Control GmbH are able to read Data-Matrix code with different moduls. One differentiates between standard variants and an extended variant. Both modules are accommodated in a dialogue.

Standard Reader Software

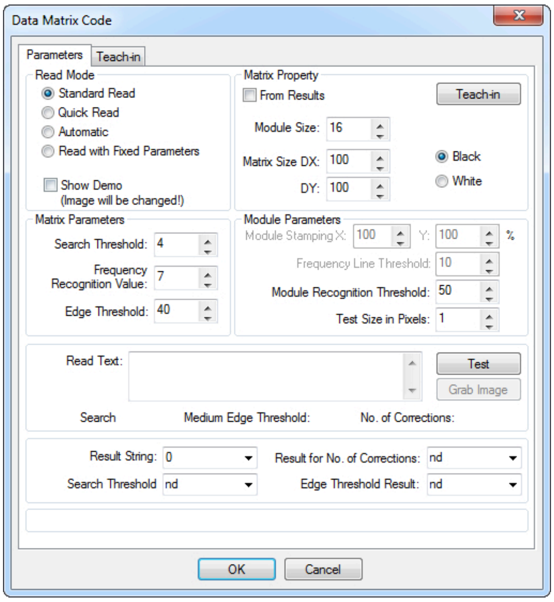
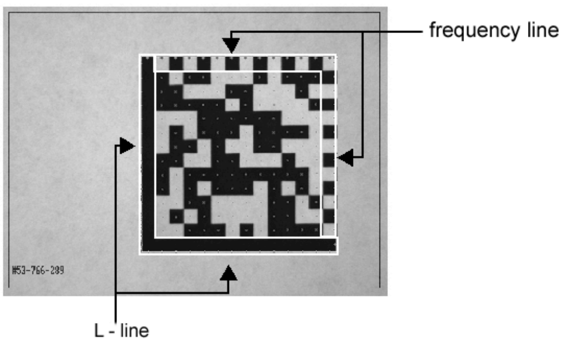


Image 76: Dialogue Data Matrix Code, tab control Parameters

Use the **Locate > Data-Matrix Code** command in the test program to read Data-Matrix Code according to the ECC200 specification.

Always orientate the code perpendicular to the camera:

- Align the Finder Pattern (L) to the bottom-left.
- Align the Alternative Pattern (frequency line) at the top-right.



Example: Data-Matrix-Code

During the evaluation, not only the index but also all the coded information is read and can be saved as necessary. The command functions according to the error handling algorithms specified by the ECC in order to ensure the highest data security.

Use this command for data acquisition if the test program should take over the tasks of de-encrypting the Data-Matrix code and the associated tasks.

Brief Guide

1. Define tin the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Activate teach-in mode by double-clicking with the right-hand mouse button.

3. Change the size and position of the search window in the video image with the controls. The search window must enclose the Data-Matrix code.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Modify and test the read parameters in points 1-4 above according to the read mode selected.
6. Insert the command into the test program with the [OK] button.

Teaching-in the Test Window

The search window must enclose the code symbol.

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Specification of the Data-Matrix Code

The Data-Matrix Code must comply with the ECC200 specification and correspond to a data area and a maximum size of 26 × 26 (88 numeric characters, 63 alphanumeric characters). The symbol cells displayed in the video image must be at least 5-6 pixels in size. Moreover, the search window must be bordered by at least a third of the code size as margins, so that the code can be detected cleanly. Small variations in the angle of rotation are tolerated up to 30°.

The following code conversions are supported:

Specification according to ECC200				Max. figures and characters contained	
Row/Column	**Data Bytes**	**Error Bytes**	**ECC %**	**Numeric**	**Alphanu-meric**
10 x 10	3	5	62.5	6	3
12 x 12	5	7	58.3	10	6
14 x 14	8	10	55.6	16	10
16 x 16	12	12	50.0	24	16
18 x 18	18	14	43.8	36	25
20 x 20	22	18	45.0	44	31
22 x 22	30	20	40.0	60	43
24 x 24	36	24	40.0	72	52
26 x 26	44	28	38.9	88	64

Read Parameters for the Different Modes

ADVICE

For time-critical applications we recommend the Quick Read or Standard Read modes.

ADVICE

If invalid values are inserted in the parameters, the vision system reverts to the standard default values.

The following read modes are available:

- Standard Read
- Quick Read
- Automatic
- Read with standard parameters

The read modes differ in which parameters should be manually adjusted or determined by the user and which are automatically preset by the image processing system.

Parameters	How the parameters can be set in the different read modes			
	Standard Read	**Quick Read**	**Automatic**	**With Standard Parameters**
Demo	Can be enabled	Can be enabled	Can be enabled	Can be enabled
Matrix Property Module Size, Matrix Size, Black/White	Preset values are modified via dialog or determined from the [Teach-in] button.		Not adjustable: The system determines the value automatically.	Not adjustable: The system determines the value automatically.
Matrix Search Parameters Search Threshold, Frequency Recognition Value, Edge Threshold	The preset values can be modified via dialog.			Not adjustable: The system uses the standard default values.
Module Parameters Module Stamping, Frequency Line Threshold	Not adjustable: The system uses the standard default values.	The preset values can be modified via dialog.	Not adjustable: The system uses the standard default values.	Not adjustable: The system uses the standard default values.
Module Parameters Module Recognition Threshold, Test Size in Pixels	The preset values can be modified via dialog.	The preset values can be modified via dialog.	The preset values can be modified via dialog.	Not adjustable: The system uses the standard default values.

Parameters »Matrix Property«	Description
Module Size	16 [(10 × 10) to (26 × 26)]
Matrix Size dX/dY	100 × 100 [(90 × 90) to (330 × 330)]
Colors	Black [Black/White]
Button [Teach-in]	Determine Data-Matrix size, module size and color via teach-in mode in the dialog box.

Parameters »Matrix Search Parameters«	Description
Search Threshold 4 [1 to 100]	Threshold for searching for the position of the code. The program only gives the Data-Matrix Code if the actual threshold lies below the threshold set.
Frequency Recognition Value 7 [4 to 20]	The number of detection beams per module point on the frequency line. Higher values can increase the detection accuracy if required.
Edge Threshold 40 [5 to 200]	Threshold for detecting the L side. Smaller values can increase the contrast if required, larger values can reduce interference.

Parameters »Module Parameters«	Description
Module Stamping % X/Y 100 [50 to 130]	The size of the area of the module points in relation to the area of the gaps on the frequency line. Module Stamping becomes smaller if the area of the module points is reduced compared with the area of the gaps.
Frequency Line Threshold 10 [1 to 200]	Detection from the opposite side to the L side. Reduce values when poor contrast or increase when there is interference, as required
Module Recognition Threshold 50 [0 to 100]	Influences averaging of all module gray scale values from their pixels. 50: Mean value >50: If the light module points are smaller than the dark ones <50: If the dark module points are smaller than the light ones
Test Size in Pixels 1 [1 to 10]	Number of pixels per module

Evaluation Parameters

Parameters	Description
Result String	Result no. for saving the text read.
Result for No. of Corrections	Result no. for saving the no. of corrections.
Search Threshold Result	Result no. for saving the search threshold.
Edge Threshold Result	Result no. for saving the edge threshold.

Test

Clicking on Test displays the recognized parameters in the display field.

Parameters	Description
Read Text	Detected content of Data-Matrix Code
Search Threshold	Values determined by program
Medium Edge Threshold	Values determined by program
No. of correc-tions	Values determined by program
Button [Test]	Execution of Data-Matrix decoding
Button [Grabl-mage]	New image acquisition

Advanced Read Function

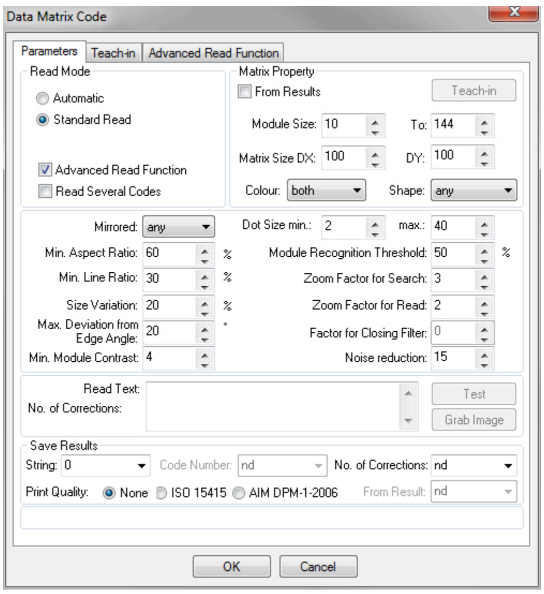


Image 77: Dialogue Data Matrix Code - Advanced, tab control Parameters

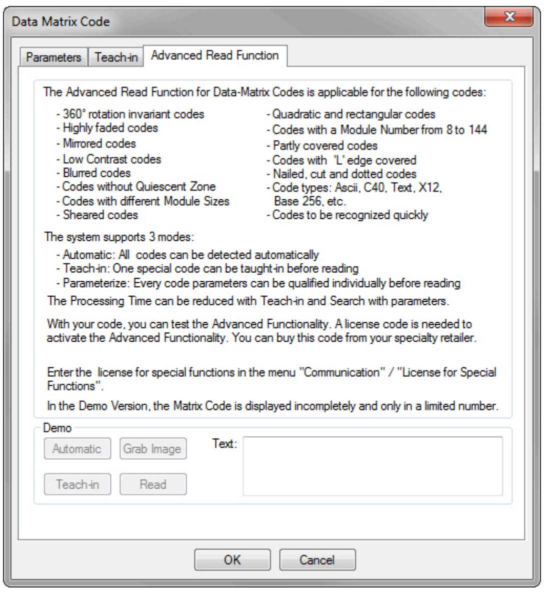


Image 78: Dialogue Data Matrix Code - Advanced, tab control Advanced Read Function

To use the advanced functionality please activate the checkmark **Advanced Read Function**.

Brief Guide

- Define the position tracking in the **Teach-in** tab: no, X point, Y point.
- Activate teach-in mode by double-clicking with the right-hand mouse button.
- Change the size and position of the search window in the video image with the controls. The search window must enclose the Data-Matrix code.
- Deactivate teach-in mode by double-clicking with the right-hand mouse button.
- Change register to **Advanced Read Function**
- Please test the data matrix code with the button **Automatic**
- Insert the command into the test program with the [OK] button.

The use of the extended Data-Matrix Code software is based on a license model. They can test your code in register **Extended Read Function** with the button **Automatic** before you buy a license. With acquisition of the license afterwards also all other modes (Parameterisation) of the extended read function are to you at the disposal.

Teaching-in the Search Window

The search window must enclose the code symbol.

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Specification of the Advanced Read Function software modul

The extended Data-Matrix Code software reads ECC200 specified code. The symbol cells (modules), represented in the video picture, must be large at least 2-3 pixels. The algorithm tolerates the following general features:

- 360° rotation invariant codes
- Code sizes of 10 to 90% of the image size
- Codes with a module number from 8 to 144 (2335 alphanumeric characters) for square code and from 8 x 18 to 16 x 48 (72 alphanumeric characters) for rectangle codes
- Reads nailed, cut and dotted codes

Further the module offers extended features.

- Complete automatic mode without any parameters
- Fast algorithm detects and reads codes with up to 150 fps
- Reading of mirrored codes
- The “L” corner can be covered (not visible)
- The “L” pattern can be partly destroyed
- The frequency lines can be partly destroyed (to about 20%)
- The algorithm is robust towards distorted and sheared codes
- Evaluation of the code's print quality according to ISO/AIM DPM (>> *"Evaluating the print quality", page 432*)

Description of the registers

Register	Description
Parameter	Options for parameterising the software module, the change-over between standard and extended software as well as the test and expen-diture part.
Teach-In	Options for the configuration of the teach-in window.
Advanced Read Function	Description of the codes, which can be read by the software, the demo part, as well as the license part to activate the extended software.

Read modes

Automatic	**Parameter-free reading from Data-Matrix Code without previous knowledge by the user - fast decoding of high-quality, printed codes**
Standard Read	Parameter dependent read function
Read Several Codes	The command reads all codes found in the image and saves the content into different strings.
Teach Mode	Teach in mode makes it possible to train a Data-Matrix Code samples. This sample wil be decoded in the automatic mode and all corresponding parameters of this individual code will be stored in a parameter list. At this point such codes with similar characteristic can be read very fast. • Code size varies from +20% to - 20% • Similar number of modules • Similar module size • Similar module proportion (lenght-wide ratio) The stored parameters can be manually refined afterwards.

Parameter description

Parameter		Description
Module Size		Adjust the module size for the code which can be read of 8 to 144.
Matrix Size		Definition of the size of the data matrix code in pixels.
Colour	both	The algorithm seeks for black modules on bright background as well as white modules on dark background.
	black	The algorithm seeks for black modules on bright background.
	white	The algorithm seeks for white modules on dark background.
Shape		The form can vary between square and rectangular. The option **any** looks for both forms in the picture.
Mirrored	any	The algorithm seeks for mirrored and non-mirrored codes.
	no	The algorithm seeks for non-mirrored codes.
	yes	The algorithm seeks for mirrored codes.

Parameter	Description
Min. Aspect Ratio	Defines the maximal proportion of opposite lines in percent. 50% means, that at opposite lines (sides of the code) the smaller line (b) should be at least 50% of the longer line (a). 
Min. Line Ratio	Defines the maximal proportion of neighbour lines ("L") in percent. 20% means, that at neighbour lines the smaller line should be at least 20% of the longer line. 
Size Variation	The deviations in per cent from the matrix size dx and dy.
Max. Deviation from Edge Angle	The angle of everyone of the 4 data matrix corners should lie in a range from 90°. The maximum deviation (Shearing) amounts to thereby 45°. 
Cell Size	The minimum and maximum size of a cell indicates in pixels. It can be appropriate between for min. 1 and max. 50.
Module Recognition Threshold	It takes a module probe colour in percent (0% = darkest colour / 100% = brightest colour).
Zoom Factor for Search in Pixels	This parameter means which zoom shot factor is to be used for finding the code in the image. It can be appropriate between min. 1 and max. 4.
Zoom Factor for Read in Pixels	This parameter means which zoom shot factor is to be used for reading the code in the image. It can be appropriate between min. 0 and max. 4.
Filter for Matrix Cell	Use of a Closing filter for preprocessing the image information. This function needs the choice of a colour (black, white). Otherwise the algorithm does not know, in which direction (colour) it shall work.
Noise reduction	Reduces background image noise. Values between 5 (slight/minimal noise) and 35 (strong noise) are possible.

ADVICE

The matrix characteristics can be determined by the software automatically. In addition you press the button **Teach-in** in the extended read function (register **Parameters**).

Use from results

Different parameters can be determined out of results. Please note, this mode needs a result for everyone of the following parameters.

- Module Size
- Matrix Size

- Colour
- Shape
- Mirrored
- Min. Aspects Ratio
- Min. Line Ratio
- Cell Size min./max.
- Zoom Factor for Search in Pixels
- Zoom Factor for Read in Pixels

Output parameters

Parameter	Description
Read Text	Expenditure of the decoded text in the dialogue.
Number of Corrections	Number of accomplished corrections for decoding the Data-Matrix Code.
Test	Button for executing the software module.
Grab Image	Button for grabbing an image of the sensor into the current image memory page.
Code Number	Result number for the storage of the number of codes.
String from	Result number for the storage of the read text.
No. of Correc-tions from	Result number for the storage of the corrections.
Print Quality	Radio buttons with the options "none", "ISO 15415" and "AIM DPM-1-2006" for selecting a standardised print quality evaluation for the Data Matrix Code. **Note**: For evaluating the print quality according to one of the guidelines, you should use the corresponding lighting definition for the respective guideline.
[Print Quality] From Result	First result number for saving the results of the individual print quality parameters.

Error display

Error	Description
No errors	No error was determined.
Error in module size	Use of invalid parameters.
No Data-Matrix Code found	No Data-Matrix Code could be found in the image.

6.2.4 Angular Position

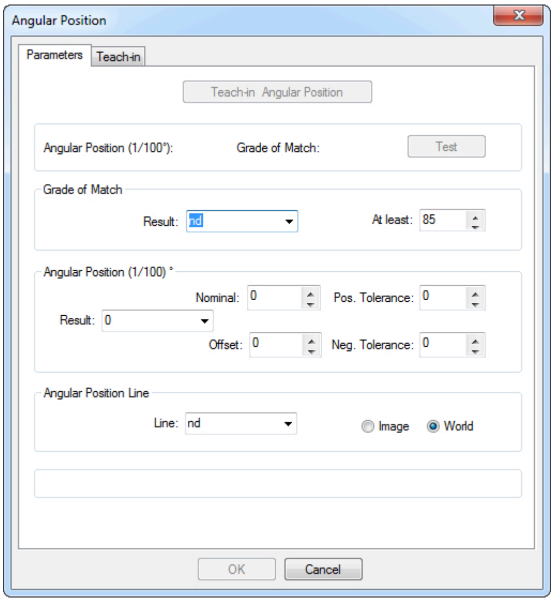


Image 79: Dialogue Angular Position, tab control Parameters

The **Locate > Angular Position** command determines a structure's angle of rotation. This structure must be taught previously. An angle and a straight line can be saved. The given angle is relative to the taught object with a proportion of 1/100°. The straight line is used for further processing, and is called phi-straight line.

ADVICE

To compensate for noise in the radial direction, increase the width of the arc. In this way you determine the average of the gray values of several concentric rings. To accelerate processing, lessen the width of the arc.

Teaching-in the Test Window

ADVICE

A center or focal point must be defined in order to update the window. Otherwise measurement errors may result. If no tracking must be carried out, the object may only be rotated and not moved in the X or Y directions. Otherwise, measurement errors arise due to position variations.

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input Parameters

You can teach-in an object in the video image with the [Teach-in Angular Position] button. After this, set the grade of match. If the preset grade of match is not reached during the detection then the command is unsuccessful.

Parameters	Description
Grade of Match	Minimum required gray value compliance (correlation) of the determined gray value distribution with the nominal gray value distribution along the circular ring-shaped detection window for teaching.

Features of Grade of Match:

- Indicates to the compliance, which exists after the internal rotation of the determined gray value distribution .
- Is set directly and must be varied if the test is unsuccessful.
- Is not influenced by the brightness in the test window.
- Is stated in per cent, 100% means exact compliance, 0% means no compliance.
- Useful settings are 60 to 80%, even 95% under good illumination.

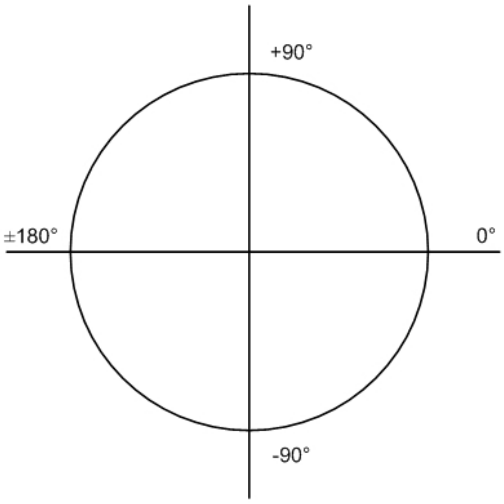
Evaluation Parameters

Angular Position (1/100 degree)	Description
Result	Result number for saving the determined angular position (1/100 degree).
Nominal and Tolerances	Nominal angular position and the permitted ± tolerances. Set suitable parameters interactively with the aid of the [Test] button. For technical reasons, tolerances cannot exceed 32768 in value (corresponds to 215). The tolerance range is from +32767 to –32767. The angular position in the first half circle is from +90° to -90° clockwise and further from -90° to +90°. The value given by the program should be divided by 100 to maintain the correct angle in degrees (°). This division factor should also be used when superimposing. Examples: 9000 corresponds to +90° / –9000 corresponds to -90°. See Diagram for angle specifications under the table.
Offset	Offset for the angular position (1/100 degree). If the object already had an initial rotation during the image acquisition, the offset can be added in order to relate the determined angular position to the X or Y coordinate.

ADVICE

The straight line with a 0° angle is defined as a parallel to the X axis through the circle center.

Angle value:

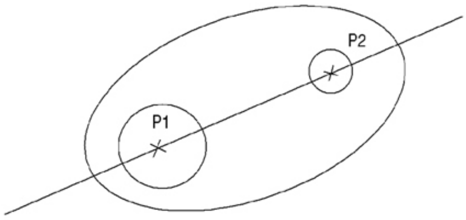


Angular Position Line	Description
Line	Line structure number for saving the line through the central point of the circular detection ring which has been rotated to the determined angle of rotation to the x-axis. If you select {-1} or {nd}, the data will not be stored.

Co-ordinate system	Description
Image/World	Storing the lines in image coordinates or world coordinates In contrast to angular position tracking for detection beams and test windows with the help of a pre-defined line, this is concerned with the determination of angular position of an object as a location parameter for gripping by robot and positioning systems. An angular position tracking is only required, as a rule, in the range between ± 90° and – as the direction of the line used is not defined – is also not possible outside this range. For the employment of pictor / vicosys within a robot system it is essential to be able to recognize the rotary angle of an object over the range of 0° to 360°.

Rotary Angle Detection Without Use of the Angular Position Command

The following section describes how to determine the angular position in a range of 0° to 360° without using the **Angular Position** command, and by conventional means. You calculate the angle of a line with two clearly defined points.



In the case of objects with a clear line which can be determined through two points, the angle of rotation can be determined as an angle between 2 points on the X and Y axes. The direction of the line is defined from point P1 to point P2. In this case, one obtains an angle between –180° and +180°.

In order to obtain the angle in the other range (e.g. 0° to 360°), the angle value determined must be added to a specific offset.

The angle between a line and an axis lies in the range from –90° to +90° because the direction of the line is not defined.

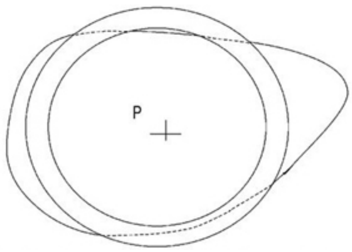
The line through the two circle center points is used to determine the angular position of the above object.

Procedure

1. Acquire image.
2. Determine the center point of circle 1 by blob analysis for the objects with the surface in the area of the circle.
3. Determine the center point of circle 2 by blob analysis for the objects with the surface in the area of the circle.
4. Determine the angle between P1, P2 and x-axis, nominal angle 0°, tolerance (0/+180°).
5. If successful, jump to step 7.
6. Add 360° to the angle (in order to convert from a negative angle in the value range 180° to 360°).
7. Insert a command to superimpose or send the angular position to the host computer.

Rotary Angle Detection with the Angular Position Command

The following section describes how to determine the rotary angle using the **Angular Position** function.



For objects in which no clear lines can be defined, the rotation can be determined with the **Angular Position** command. The requirement is a homogenous background.

Procedure

Determine the angle of rotation of the object at the time of testing compared with the time of teaching as follows:

1. Acquire image.
2. Determine the center point of contour of the object during teaching by **Blob Analysis** or by combining **Locate Contour** and **Center of Gravity of Contour**.
3. Specify the nominal gray scale value distribution along an annulus around the contour focal point with the **Angular Position** command: During the test, the determined gray value distribution is compared with the nominal gray value distribution along a circular ring around the center point of the contour. In this, the angle of rotation is determined by an internal correlation calculation.
4. Insert a command to superimpose or send the angular position to the host computer.

The rotary angle to the determined axes is obtained by adding a defined offset in the Angular Position command. The circular ring diameter must be selected so that the angular position can be clearly determined during teaching.

Test

The determined **angular position (1/100 degree)** and the **grade of match** are displayed in the test.

6.2.5 Rotation based on Moments

Overview

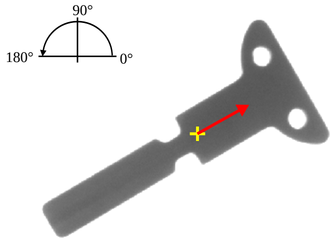


Image 80: simple rotation detection 180°

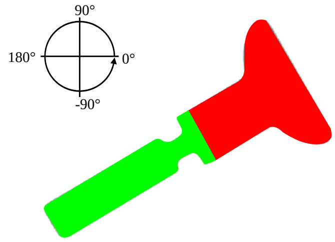


Image 81: advanced rotation detection 360° with extensity display

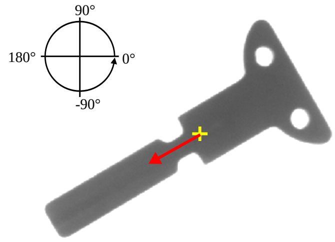


Image 82: advanced rotation detection 360° result

With **Locate > Rotation based on Moments** you can locate objects very fast in the image and determine its angular position. The object is found by its specific brightness distribution (moment). The command is only suitable for scattered objects in high-contrast images.

The test object must have a clear axis of inertia (long, straight, narrow test parts).

For the advanced rotary position sensing, the eccentricity (expansion along the axis of inertia) is considered to determine the location from 0° to 360°. The test object should be wider on one side.

Workflow
Teach-in

1. Switch to the tab control "***Teach-in***".
2. Determine here by means of the geometries in witch area moments should be searched (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow
Parameterizing and Testing

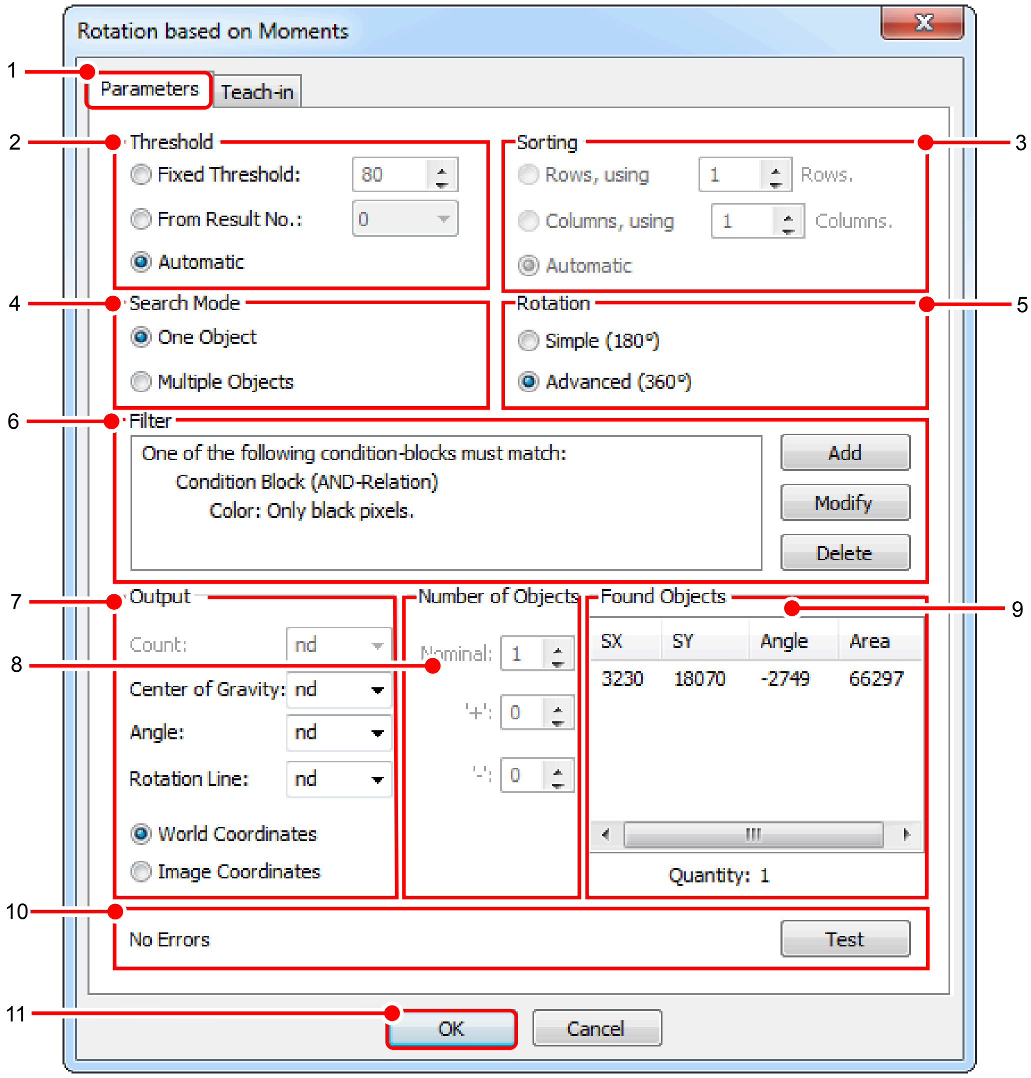


Image 83: Dialog Rotation based on Moments, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Choose in the area "Threshold" the threshold for detection and subdivision of the subjects of the image.

Parameter	Description
Fixed Threshold	Enter the gray value threshold for detection and subdivision of the objects.
From Result No.	You can choose a gray value threshold from the result structure here.
Automatic	With definite distribution in light and dark objects, you can use this option. The threshold is set automatically.

3. Choose in the area "Sorting" if a special sorting of the results should take place. After a test, the rows or columns (except automatic) and also the bonding-box around each object are displayed in the monitor window.

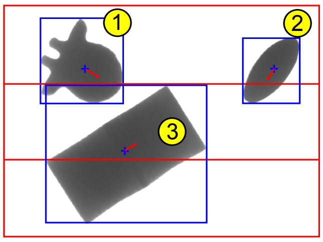


Image 84: Sorting: 3 Rows

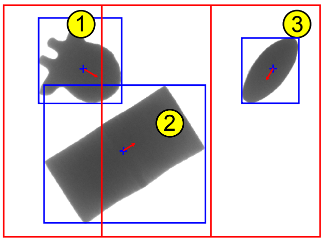


Image 85: Sorting: 3 Columns

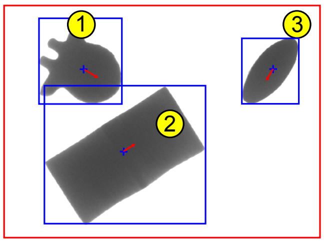


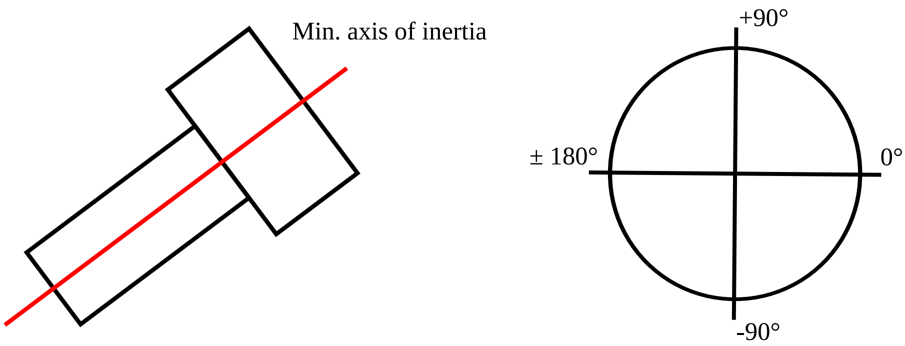
Image 86: Sorting: Automatic

Parameter	Description
Rows, using n Rows.	Division of the inspection area into n-rows. The sorting of the objects in the rows is performed after the x-coordinates of the center of gravity points, then the y-coordinates of the center of gravity points.
Columns, using n Columns.	Division of the inspection area into n-columns. The sorting of the objects in the columns is performed after the y-coordinates of the center of gravity points, then the x-coordinates of the center of gravity points.
Automatic	Sorting of all objects whose bounding-box in the y-direction overlaps (vertical projection), first in the x-direction. Then, the so created rows are sorted in the y-direction.

4. Search Mode

Parameter	Description
One Object	All found areas are interpreted as a single object.
Multiple Objects	All found areas are interpreted as multiple objects.

5. Rotation



Parameter	Description
Simple (180°)	Sets the angular position according to the minimum axis of inertia. The minimum axis of inertia is an undirected straight line. Use this straight line only if the object rotates by less than 180°.
Advanced (360°)	Carries out a 360° recognition based on eccentricity after simple angular position recognition.

6. In the area "Filter" you can specify conditions for the objects to be evaluate. Read more: *see "Use of filters", page 201*.
7. In the area "Output" set the options for saving the results.

Parameter	Description
Count	Here you enter the number or the name of the result for saving the number of objects found in the result structure.
Center of Gravity	Enter the number or name for saving the center of gravity point in the point structure. If "Multiple Object" was selected multiple center of gravity points are saved with index extension (see also: *"Names for Geometry Variables", page 59*).
Angle	Enter the number or name for saving the angle in the result structure. If "Multiple Object" was selected multiple angles are saved with index extension (see also: *"Names for Geometry Variables", page 59*).
Rotation Line	Enter the number or name for saving the rotation line in the line structure. If "Multiple Object" was selected multiple rotation lines are saved with index extension (see also: *"Names for Geometry Variables", page 59*).
World Coordinates	The storage of the values appears in world coordinates.
Image Coordinates	The storage of the values appears in image coordinates.

8. In the area "Number of Objects" whether the detected number of objects is within the tolerance range, and therefore the command is evaluated with good or bad during execution.

Parameter	Description
Nominal	Here you set the nominal quantity of objects to be found.
'+' / '-'	Permitted upward and downward deviations from this target value.

9. In the area "Found Objects" all to the corresponding requirements found objects are listed.

Parameter	Description
SX	X-coordinate of the determined Center of gravity point.
SY	Y-coordinate of the determined Center of gravity point.
Angle	Angle of the axis of inertia of the found object in 1/100° relative to 0°.
Area	Number of pixels of the found object.
Quantity	Number of found objects, corresponding to the conditions.

10. Testing and setting the appropriate parameters
By activating the [Test] button the command with the set parameters is executed. In the area to the left of the [Test] button, the evaluation of the command (error-free) as well as instructions for setting parameters (tolerance range errors) and error messages are displayed. In the "Found Objects" area, all matching objects are displayed. In the video image the center of gravity point and the vector of the axis of inertia are displayed.
If necessary, change the settings in step 2 till 6 until the result meets your requirements.

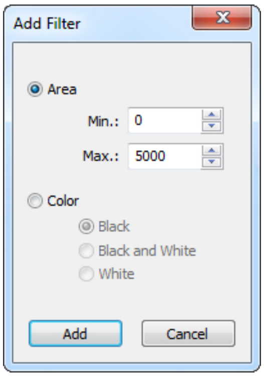
11. Use the [OK] button to insert the command into the test program.

Use of filters

You can use filters to filter by criterias (Area or Color) within the algorithm.

A filter contains condition blocks. These blocks are linked by a logical OR. Only one condition block must fit the requirements.

Each condition block can contain up to two filters. These are linked by a logical AND. Both of these filters have to be fulfilled, so that the condition block is fulfilled.



In the area "Filter", click on the button [Add]. In the dialog "Add Filter" you can define conditions. In the area "Filter" the logical links are shown as full text.

To modify or delete a filter, select it and click [Modify] or [Delete].

To insert a new filter, click into the corresponding condition block and then click [Add].

To insert a new condition block, select the top line "One of the following condition-blocks must match:" and click [Add].

Parameter	Description
Area	
Min.	The object must contain at least this number of pixels, so that it is detected. Objects with fewer pixels are ignored.
Max.	The object must be less than this number of pixels, so that it is detected. Objects with more pixels are ignored.
Color	
Black	Only objects with darker color than the set threshold will be considered.
Black and White	Objects with brighter and darker color than the set threshold will be considered.
White	Only objects with brighter color than the set threshold will be considered.

6.2.6 Color Blob Analysis

Overview

Image 87: Color Blob Analysis - test object

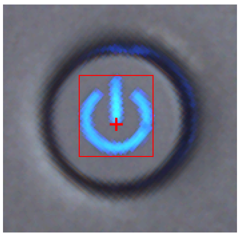


Image 88: Color Blob Analysis - blob detected

With **Locate > Color Blob Analysis** you determine center of gravity points and areas of colored areas in the image. For this, internally the color image is binarized and then evaluated. You can save up to 50 center of gravity points and values in the point- and result structure. Center of gravity points and areas in excess of 50 are not evaluated. Use this command for counting colored objects, finding objects or for position tracking.

Work Flow

Define Area of Binarization

1. Switch to the tab control "***Teach-in***".
2. Determine here by means of the geometries in witch area of the image the objects are to be searched. In this area your image is converted into a black-and-white image (see also *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow
Parameterizing and Testing

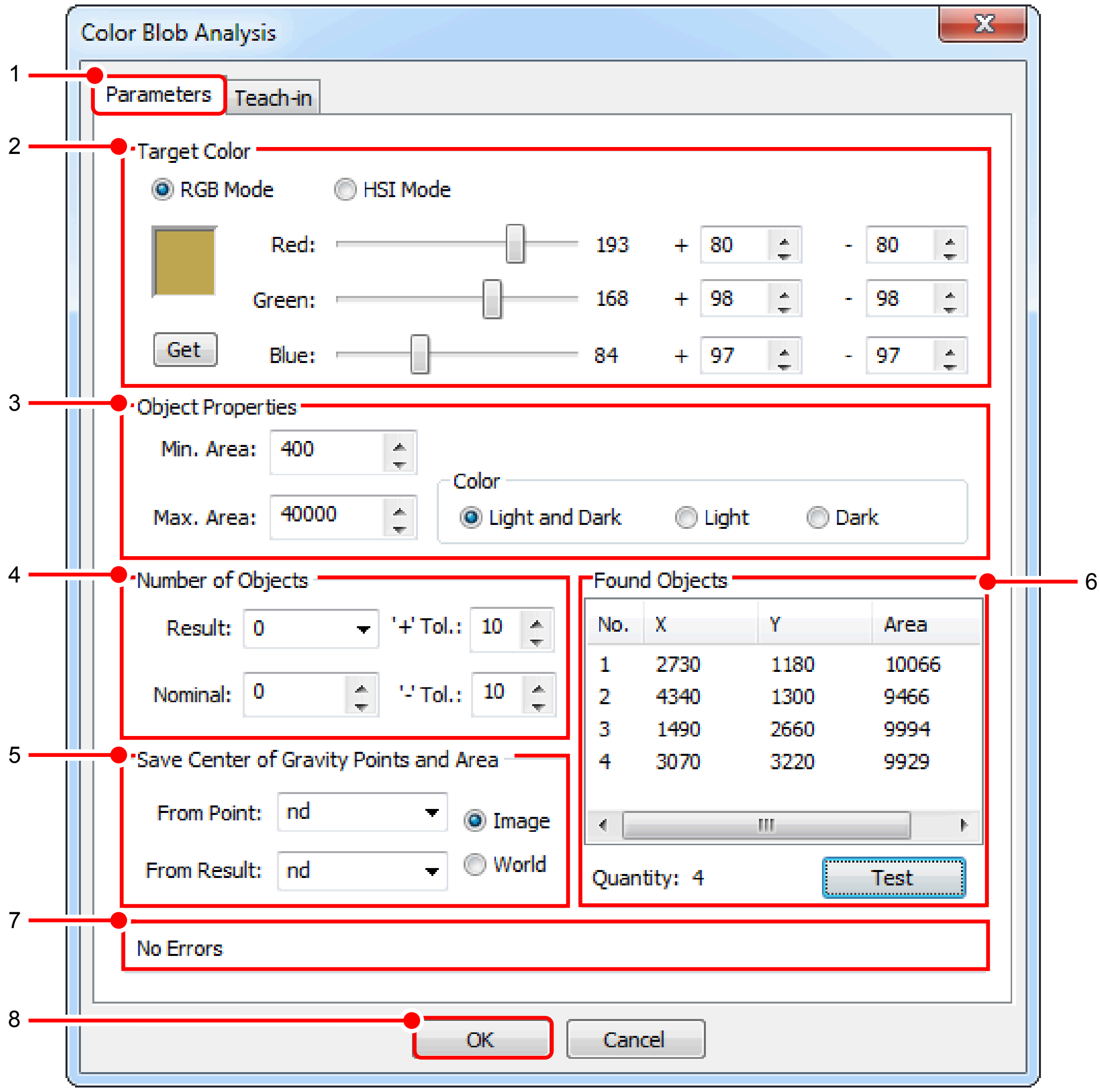


Image 89: Dialog Farbblobanalyse, Register Parameter

1. Switch to the tab control "***Parameters***".
2. Please set here the parameters for color range, nominal color and color deviation.

Parameter	Description
RGB Mode/ HSI Mode	Selection whether in the RGB or HSI color space should be checked. Use the HSI mode if you want to check the hue or the saturation irrespective of the brightness.
Slider Red, Green, Blue	If the selection "RGB Mode" is active. Values of the corresponding color channel.
Slider H, S, I	If the selection "HSI Mode" is active. Values for color (Hue), saturation and intensity.
Tolerance	Minimum and maximum tolerances of the respective color channel.
[Get]	The average color of the test window is taken as the default for the target color.

ADVICE

Using the button [Get]

For best results, go to the tab control "***Teach-in***", here reduce the binarization window and place it in a unique color range. Then, go to the tab control "***Parameters***" and click the button [Get] to get the values of the color of the binarization window.

Then adjust the test window as usual.

3. Determine in the area "Object Properties" the minimum and maximum area of the objects to be found.

Parameter	**Description**
Min. Area/ Max. Area	Sets the minimum and maximum number of pixels of the object. Only objects with pixel quantity within this range are detected.
Color	
Light and Dark	Parts of the image will be assigned to the objects that are within and outside of the specified color range.
Light	Only parts of the image will be assigned to the objects that are within of the specified color range.
Dark	Only parts of the image will be assigned to the objects that are outside of the specified color range.

4. Here you set the nominal quantity of objects to be found. If the number of pixels found lies within this range, the command ends with a good result (otherwise with bad).

Parameter	**Description**
Result	Here you enter the number or name of the result in the result structure in which the determined objects should be saved.
Nominal	Enter here the nominal quantity of objects to be found.
Tolerances	Allows an upper and lower deviation from the nominal quantity.

5. Here you can specify the storage options for the center of gravity and the area of the color blobs.

Parameter	**Description**
From Point	From this point, the center of gravity points of the objects found are stored.
From Result	From this position in the result structure, the area values of the objects found are stored.
Image/World	The ratio buttons determine if the storage of the points is made in image coordinates or world units.

6. Testing and setting the appropriate parameters
   By activating the [Test] button the command with the set parameters is executed. In the area "Found Objects" all corresponding blobs are shown. In parallel, the objects found are marked in the video image with a red rectangle and a cross as center of gravity point.
7. Evaluation Area
   Here you can see the good/bad evaluation of the command, notes for parameterization (tolerance range exceedance) as well as error messages.
8. Use the [OK] button to insert the command into the test program.

6.2.7 Test Colour

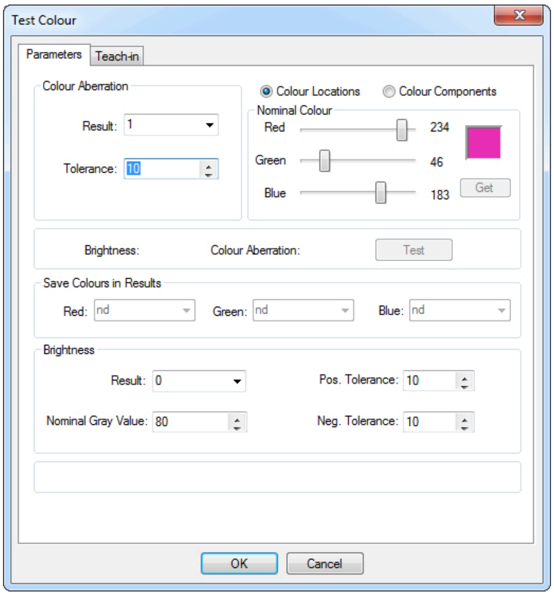


Image 90: Dialogue Test Color, tab contol Parameters

You can add a command for the colour test into the test program using **Locate > Test Colour**.

A colour search is performed within a test window in order to recognize surface defects. The colour location and brightness or the colour components red, green and blue can be individually tested. The proportions of the red, green and blue colour components which have been determined can be stored.

Parameters

- Position and size of the test window
- Number of a reference point (with position tracking)
- Defined RGB nominal colour location and tolerances
- Result
- Nominal gray value and tolerances

Teaching-in the Test Window

The test window is for teaching positioning on an object with the reference colour with the **Chromaticity** method.

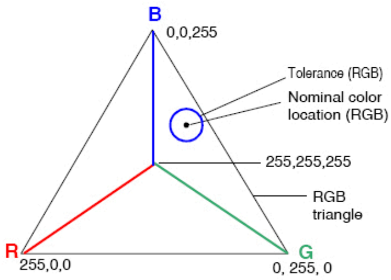
Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Method 1: Defining Colour using Colour Location



Color location (RGB model)

1. Select the **Colour Locations** option. Click on the [Read] button.

   ◦ The colour is taken from the test window.
   ◦ The colour is displayed in miniature.
   ◦ The colour is set as the nominal chromaticity coordinate.

2. Define permitted colour deviation
3. Specify how colour deviation is saved in the result structure.

»Colour Aberration« Input Parameters	Description
Result	Result number for saving the determined colour deviation.
Tolerance	Tolerance for nominal chromaticity, see chromaticity (RGB model).

Method 2: Defining Colour using Colour Components

1. Select the **Colour Components** option.
2. Set nominal colours with the **Red**, **Green**, and **Blue** slide controls.
   The colour is displayed in miniature.
3. Define permitted colour deviation.
4. Specify how the colour components in results are saved.

Parameters	Description
Colour Aberration > Red, Green, Blue	Set tolerances with the red, green and blue slide controls.
Save Colours in Results > Red, Green, Blue	Result numbers for saving the determined colour components.

Evaluation Parameters

Brightness	Description
Result	Result number for saving the determined average gray value.
Nominal Gray Value and Tolerances	Nominal average gray value in the test window and the permitted tolerances (0 to 255).

Test

The test displays:

- Brightness … Determined average gray value.
- Colour Deviation … Determined deviation from nominal chromaticity.

6.2.8 Focus

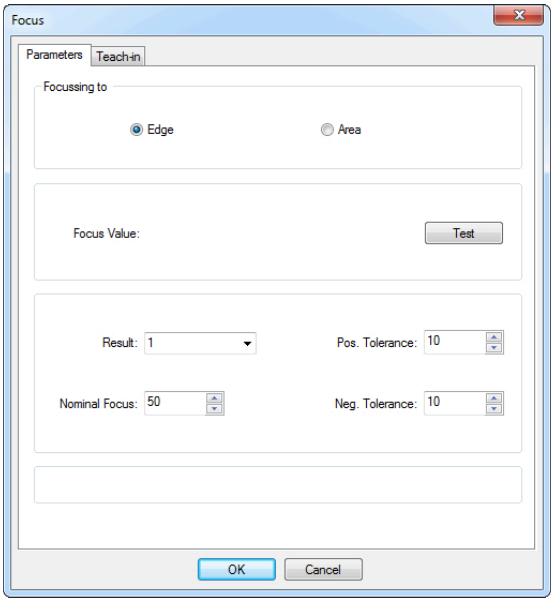


Image 91: Dialogue Focus, tab control Parameters

You can insert a command for determining a focus value as a criterion for the image sharpness into the test program using **Locate > Focus**. A prerequisite for this command is a location in the measurement window containing information about sharpness.

You can use this command for the following options:

- automatically focusing the camera when it is connected to motion systems
- automatically post-focusing cameras by combining the Focus command with an adjustment command for the camera

Brief Guide

1. Define in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Specify the following in the **Parameters** tab:
   - Structure used for focusing
   - Result no.
   - Nominal focus and tolerances (quality criterion)
3. Test the command with the [Test] button and modify the parameters set in points 1 to 5 until the test result is error free.
4. Insert the command into the test program with the [OK] button.

Teaching-in the Test Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input Parameters

ADVICE
For the sharpness information required it is preferable to use multiple lines with edges that are as sharp as possible.

Focusing	Description
Edge	Focusing on an edge.
Area	Focusing on an area.

Evaluation Parameters

Parameters	Description
Result	Result number for storing the focus value.
Nominal Focus and Tolerances	The nominal focus value and the allowed ± tolerances.

The **Focus Value** test result is calculated as a normalized sum of all focuses of the edges/ areas in the image area. In order to exclude edges/areas of low focus (e.g. noise) only those edges are considered which lie within 5% of the sharpest focus and areas which lie within 20% of the sharpest focus.

Test

Change the focusing of the camera in the same section of the image. Using test produces the focus value. The higher the value, the better the focus is. Take note that the focus values between different images cannot be compared to each other.

6.2.9 Locate Line

Overview

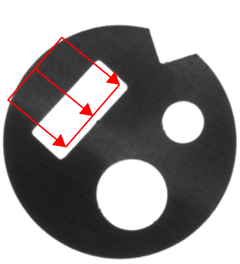


Image 92: Detection window at the desired position

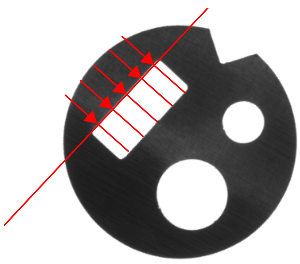


Image 93: Composed line, using 5 detection points

With **Locate > Locate Line** you write a command into your test program, that detects points and composes a line from these points.

Work Flow
Teach-in

1. Switch to the tab control "***Teach-in***".
2. Set here the position of your detection window (see also: *"Teaching in Test, Detection and Search windows", page 26*). In this window points are detected (single detection points) and the line is defined.
3. Define the quantity of detection points (single detection points). These are evenly distributed over the width of the detection window.
4. Configure if all detection points must be found so that the line can be defined from all single detection points.
5. If necessary, set the parameters (X Point, Y Point) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Workflow
Parameterizing and Testing

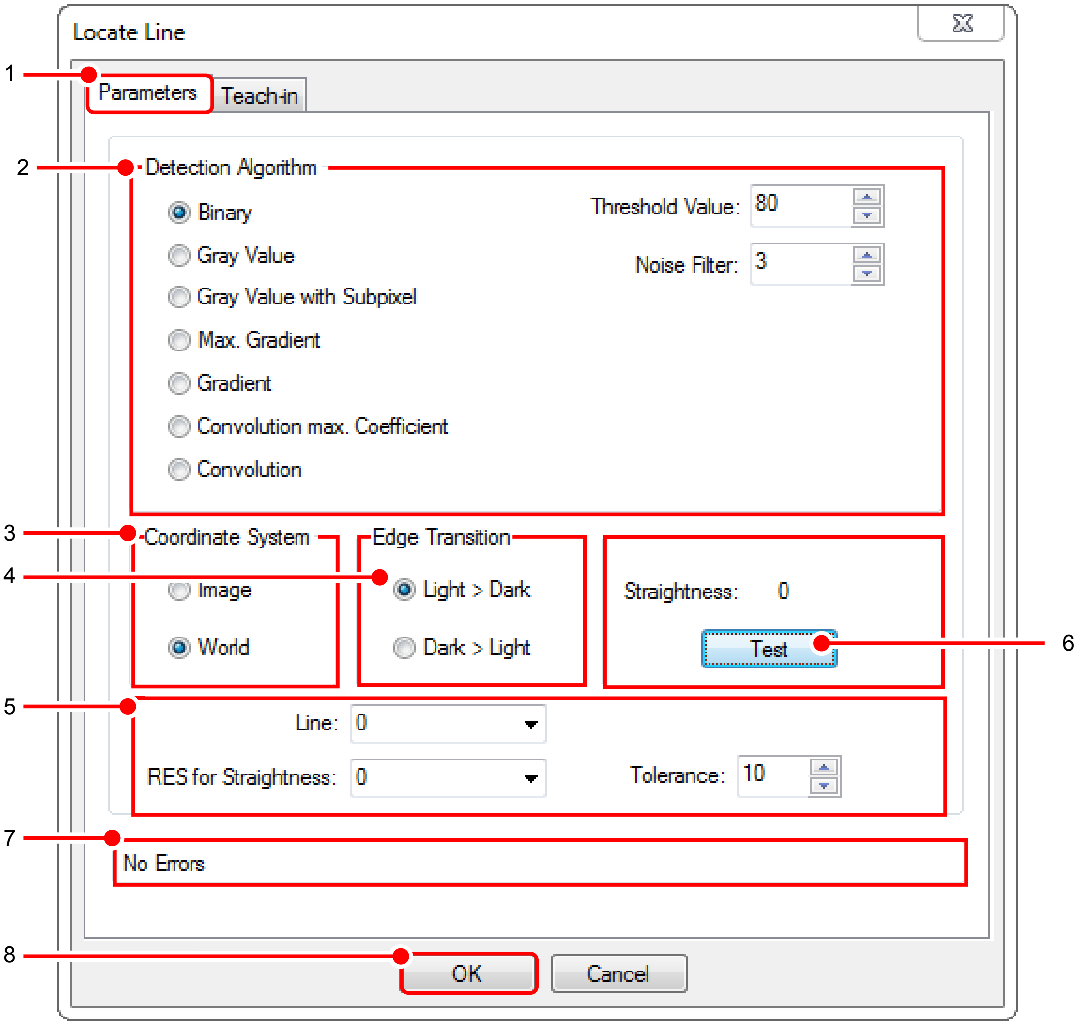


Image 94: Dialog Locate Line, tab control Parameters

1. Switch to the tab control "***Parameter***".
2. Choose in the area "Detection Algorithm" the best fitting algorithm and its parameters for your image (see also: *"Principles of Detection", page 417*).
3. In the area "Coordinate System", set whether you want to save the line in image- or world coordinates.

Parameter	Description
Image	The storage of the line is carried out in image coordinates.
World	The storage of the line is carried out in world coordinates.

4. In the area "Edge Transition", specify the type of an edge along the detection direction in the window as light to dark or dark to light.

Parameter	Description
Light->Dark	Only edge transitions from light to dark will be considered.
Dark->Light	Only edge transitions from dark to light will be considered.

5. Set here the options for saving the detected line.

Parameter	Description
Line	Number or name under which the coordinates of the found line are stored in the line structure.
RES for Straightness	Number or name under which the straightness is stored in the result structure.
Tolerance	Perpendicular distance in pixels between the furthest above and underneath point. These points form a tolerance band around the straight line.

6. Testing and setting the appropriate parameters
   By activating the [Test] button the command with the set parameters is executed. The values of the straight lines found will be displayed in the docking window Lines. If declared, the result of the straightness is shown in the docking window Results and in the dialog (above the [Test] button). In parallel, the composed line with the corresponding detection points is shown in the video image.
   Change the settings in step 2 until the result meets your requirements. If necessary, switch to the tab control "***Teach-in***" and adjust the number of detection points.
7. Evaluation Area
   Eventually error messages are shown here.
8. Use the [OK] button to insert the command into the test program.

6.2.10 Test Gray Value

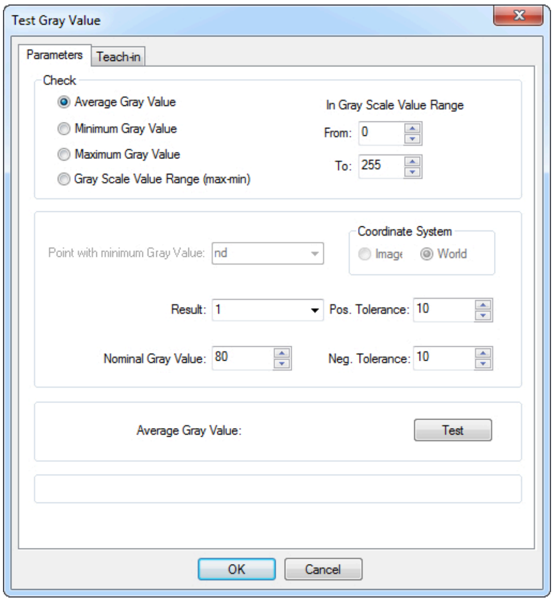


Image 95: Dialogue Test Gray Value, tab control Parameters

With Locate > Test Gray Value, you include a command in the test program allowing you to determine a specific grayscale value. You can choose here between four kinds of measurement:

- **Average Gray Value** - determines the average grayscale value in a specified grayscale range
- **Minimum Gray Value** - determines the lowest grayscale value (darkest point) in a specified grayscale range, and returns its coordinates
- **Maximum Gray Value** - determines the highest grayscale value (brightest point) in a specified grayscale range, and returns its coordinates
- **Gray Scale Value (max-min)** - determines the difference between the highest and the lowest grayscale values in a specified grayscale range ("greyscale dynamics")

The command checks whether the value determined is within the preset tolerance range. The determined value is then saved in the result structure. When measuring the **Minimum Gray Value** and **Maximum Gray Value**, you can also save the coordinates of the determined point in the result structure. The position of the first point that is found with the respective minimum and maximum gray scale values is always used. Use this command to monitor and control the brightness of the lighting. In the case of more complex structures, you can change the settings for the Gray Scale Value Range to adjust and narrow down the measurement.

Brief Guide

1. Define the following parameters in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the search window in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Specify the following in the **Parameters** tab:

- Result
- Nominal gray value
- Tolerances
- with Minimum/Maximum Gray Value: point with the minimum/maximum greyscale value (Point number), Image/World coordinates

6. Test the command with the [Test] button and modify the parameters set in points 1 to 5 until the test result is error free.
7. Insert the command into the test program with the [OK] button.

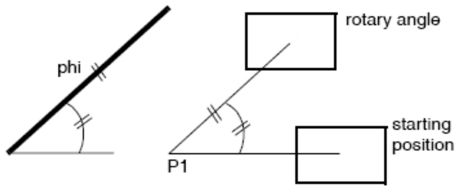
Teaching-in the Test Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*



Example: rotation tracking

The rotary angle results from the angle between the position tracking line and the X axis. The direction and size of the window are maintained.

Evaluation Parameters

Parameters	Description
Gray Scale Value Range	Grayscale limits specifying the maximum and minimum grayscale values that are allowed for the value to be determined.
Point with the minimum/ maximum gray scale value	Point number for storing the position of the determined point.
Image/World	Saving in image coordinates or world coordinates.
Result	Result number used for storing the result in the result structure.
Nominal Gray Value	Nominal gray value
Tolerances	Accepted positive and negative tolerances in gray values from 0 to 255.

Test

The test displays the **Average Gray Value** calculated in the test window.

6.2.11 Test Brightness Percentage

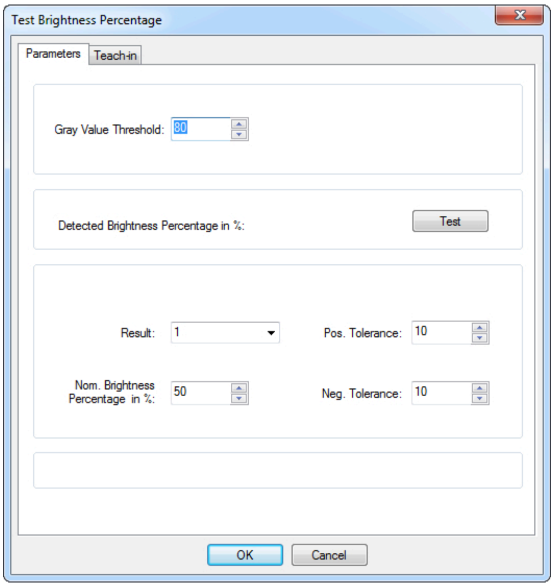


Image 96: Dialogue Test Brightness Percentage, tab control Parameter

You can determine the percentage of bright points in an image window using **Locate > Test Brightness Percentage**. The command tests whether the result lies within a certain tolerance range. Any points whose gray values exceed the gray value threshold are considered “bright”. The detected brightness percentage is stored in the result structure.

Brief Guide

1. Define the following parameters in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the search window in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Specify the following in the **Parameters** tab:
   - Gray value threshold
   - Result
   - Nominal brightness percentage
   - Tolerances for brightness percentage in %
6. Test the command with the [Test] button and modify the parameters set in points 1 to 5 until the test result is error free.
7. Insert the command into the test program with the [OK] button.

Teaching-in the Test Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input Parameters

Parameters	Description
Gray Value Threshold	Gray value threshold above which a point will be considered bright.

Evaluation Parameters

Parameters	Description
Result	Result Number for storing the percentage of bright points.
Nom. Brightness Percentage and Tolerances	Nominal percentage of bright points and accepted ± tolerances in per cent.

Test

The test displays the **Detected Brightness Percentage** calculated in the test window.

6.2.12 Brightness Offset

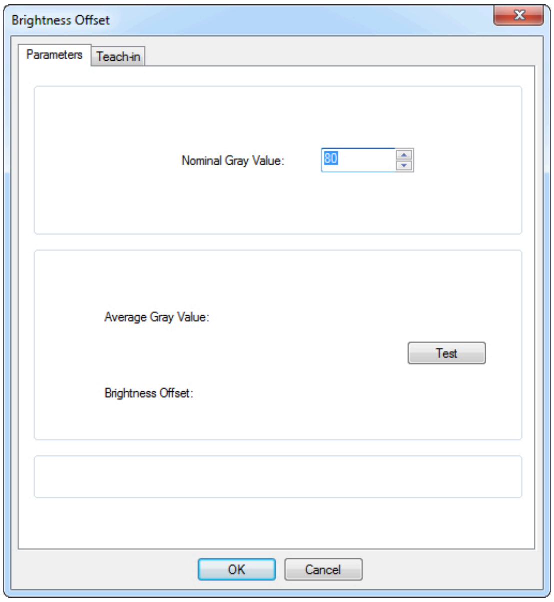


Image 97: Dialogue Brightness Offset, tab control Parameters

You can determine the deviation of the actual mean gray value of a test object within a test window from an expected nominal gray value using **Locate > Brightness Offset**. The brightness offset is saved internally for the image memory page with vicosys. The Brightness Offset is automatically added to the threshold value set with commands using the Binary algorithm (Locate, Blob Analysis). The Brightness Offset is reset with each new image acquisition.

Brief Guide

1. Define the following parameters in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the search window in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Define the nominal gray value in the **Parameters** tab.
6. Test the command with the [Test] button.
7. Insert the command into the test program with the [OK] button.

Teaching-in the Test Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input Parameters

Parameters	Description
Nominal Gray Value	A reference gray value (0 to 255) to which the offset is related.

Evaluation Parameters

Parameters	Description
Brightness Offset	The determined difference from the reference gray value: The brightness offset is stored internally.

Test

The test displays:

- Average Gray Value … Nominal pixel quantity determined in the test window.
- Brightness Offset … Determined brightness offset in ± gray values.

6.2.13 Locate Edges on Circle

Overview

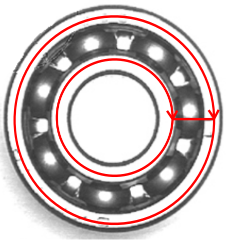


Image 98: Detection Window at position

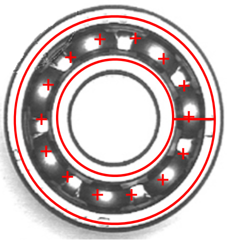


Image 99: Found edge transitions at the balls

With **Locate > Locate Edges on Circle** you can teach-in a command for counting the edges on an arc. The number of edges found is used as a quality criterion and is stored in the result structure.

To compensate for the noise in the radial direction, increase the width of the arc. In this way you determine the average of the gray values from several concentric rings. If required, we recommend changing the radius by linking it to a result number. A binary and a gradient algorithm are available for contour determination.

Work Flow Teach-in

1. Switch to the tab control "***Teach-in***".
2. Set the position of the detection area Circular Ring Segment as well as the angles and radii: (see also: *"Teaching in Test, Detection and Search windows", page 26*). In this section edges (single detections) are sought.
3. Set the search direction (clockwise / counterclockwise).
4. If necessary, set the parameters (X Point, Y Point, Rad. RES) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow Parameterizing and Testing

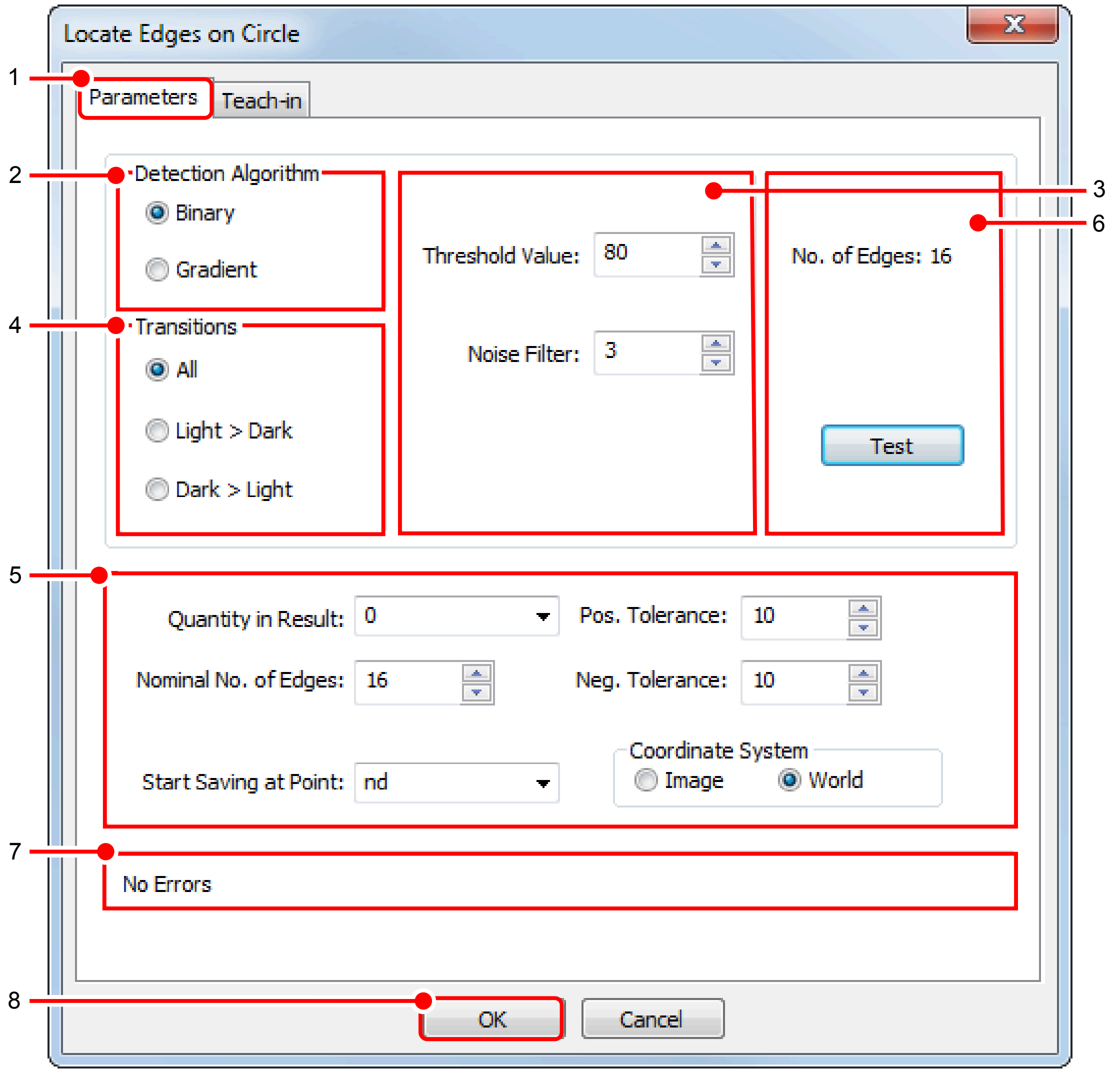


Image 100: Dialog Locate Edges on Circle, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Set here the detection algorithm. You will find further information at: *"Principles of Detection", page 417*.

Parameter	**Description**
Binary	With the binary algorithm, the edges are recognized when a defined gray value threshold is reached. Parameters: threshold value, noise filter
Gradient	In edge detection with gradient, the edges are determined with the gradient process. Parameters: threshold value, noise filter, diffusion

3. Here you can set the parameters for the detection algorithm. You will find further information at: *"Principles of Detection", page 417*.
4. The positions of the edges are determined in the middle of the arc and can be stored in the point structure in the order in which they were found. The type of transition can be either light-dark or dark-light.

Parameter	Description
all	All dark-light and light-dark edge transitions are counted.
Light > Dark	Only light-dark transitions are counted.
Dark > Light	Only dark-light transitions are counted.

5. Here you can specify the storage options for the edges found.

Parameter	Description
Quantity in Result	Here you enter the number or name of the result in the result structure in which the determined edge number should be saved.
Nominal No. of Edges	Here you set the nominal quantity of edges to be found. If the number of edges found lies within this range, the command ends with a good result (otherwise with bad).
Tolerances	Allows an upper and lower deviation from the nominal quantity.
Start Saving at Point	Number or name of the point structure, in which the detection point for the first edge found, is stored.
Coordinate System	Saving in image coordinates or world coordinates.

6. Testing and setting the appropriate parameters
   By activating the [Test] button the command with the set parameters is executed. In this area the number of edges is shown. In parallel, the edges found are marked in the video image with a red cross.
7. Evaluation Area
   Here you can see the good/bad evaluation of the command, notes for parameterization as well as error messages.
8. Use the [OK] button to insert the command into the test program.

6.2.14 Count Edges

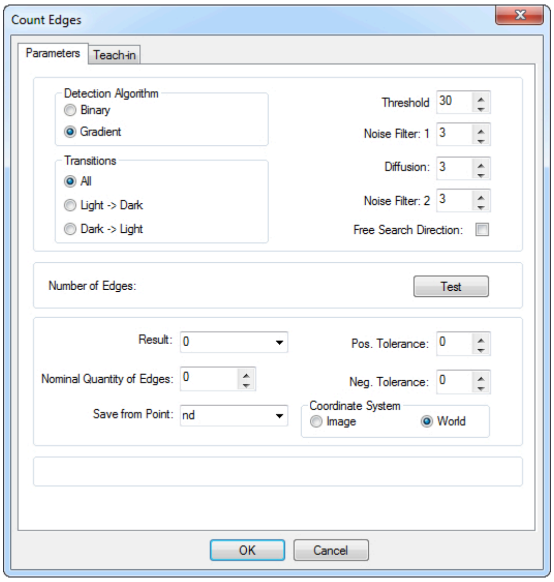


Image 101: Dialogue Count Edges, tab control Parameters

You can Teach-in a command for counting edges along a detection beam with **Locate > Count Edges**. The number of edges found is used as a quality criterion and is stored in the result structure.

To compensate for the noise at right-angles to the detection direction, increase the width of the detection. In this way you can determine the average of the gray values from several parallel detections.

A binary and a gradient algorithm are available for contour determination.

Brief Guide

1. Specify the following in the **Parameters** tab:
   - Detection algorithms with corresponding parameters
   - Edges (light > dark, dark > light, all)
   - Free search direction
   - Nominal number of edges
   - Result number and tolerances
   - Point number from which all the edge positions are stored
   - Co-ordinate system
2. Test the command with the [Test] button and modify the parameters set in points 1 to 5 until the test result is error free.
3. Insert the command into the test program with the [OK] button.

Teaching-in the Detection Beam

Position and Appearance of the Search Window
After double-clicking with the right mouse button on the **Teach-in** tab, a detection beam dialog box and the detection beam will appear on the video screen.
More Information:

- *"Teaching in Test, Detection and Search windows", page 26*
- *"Teaching-in the Detection Beam", page 249*

Tracking the Position

The detection beam can either always remain at the same image position or follow a reference object.

>> *"Position Tracking of Objects", page 35*

Input Parameters

To find edges:

- Use the [Test] button.
- Use the binary algorithm or the gradient algorithm.

The search for edges is carried out at a line. >> *"Principles of Detection", page 417*

Detection Algorithm	Description
Binary	With the binary algorithm, the edges are recognized when a defined gray value threshold is reached. Parameters: threshold value, noise filter
Gradient	In edge detection with gradient, the edges are determined with the gradient process. Parameters: threshold value, noise filters 1 and 2, diffusion (noise filter 1 refers to the pixel before the edge in the detection direction, noise filter 2 refers to the pixel after the edge, see example)
Free Search Direction	If **Free Search Direction** is inactive, only detection in a 45° grid is possible, i.e., the detection arrow jumps at an angle of 45° to the position with the best contrast. The speed of the command is hence maximized. Activate **Free search direction** to set the search direction for max. gradient, gradient, convolution max. coefficient, and convolution.

The type of transition can be either light-dark or dark-light.

Edge Transition	Description
All	All dark-light and light-dark edge transitions are counted.
Light > Dark	Only light-dark transitions are counted.
Dark > Light	Only dark-light transitions are counted.

Evaluation Parameters

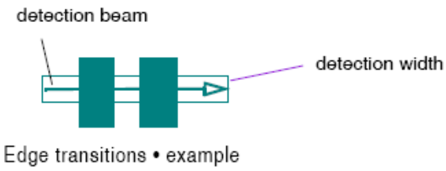
Parameters	Description
Result	Result number for saving the number of detected edges.
Nominal Quantity of Edges and Tolerances	Nominal quantity of edges and permitted ± tolerances (in number of edges)
Image/World	Saving in image coordinates or world coordinates.
Save from Point	Number of the point structure, in which the detection point for the first edge found is stored. The detection points of further edges are stored in the subsequent numbers of the point structure.

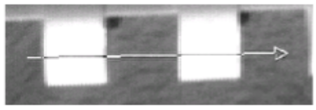
Test

The **quantity of transitions** determined in the test is displayed.

Examples

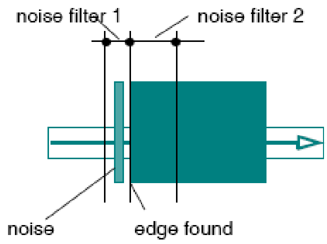
Counting the Edge Transitions





Parameters	Description
All	Four edge transitions are counted.
Light > Dark / Dark > Light	Two edge transitions are counted.

Using 2 Noise Filters for Detection with Gradient



Two noise filters for detection with gradient • Example of bright ->dark edge detection

The two noise filters can be so combined that noise is eliminated. Select noise filter 2 (refers to the pixel after the edge found) to be larger than noise filter 1 (refers to the pixel before the edge found).

6.2.15 Edge Based Object Search

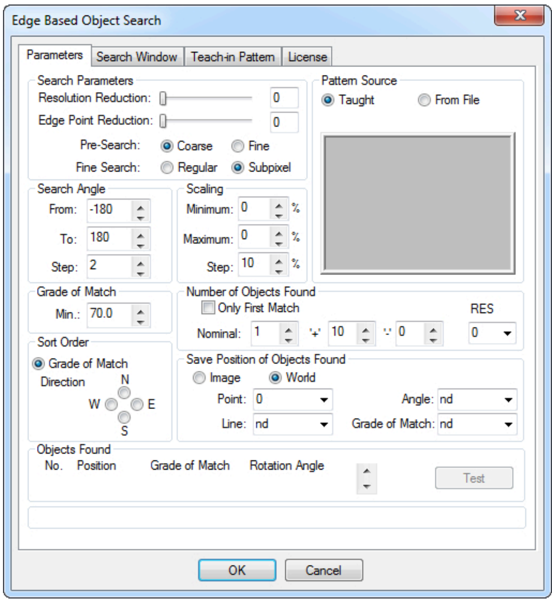


Image 102: Dialogue Edge Based Object Search, tab control Parameters

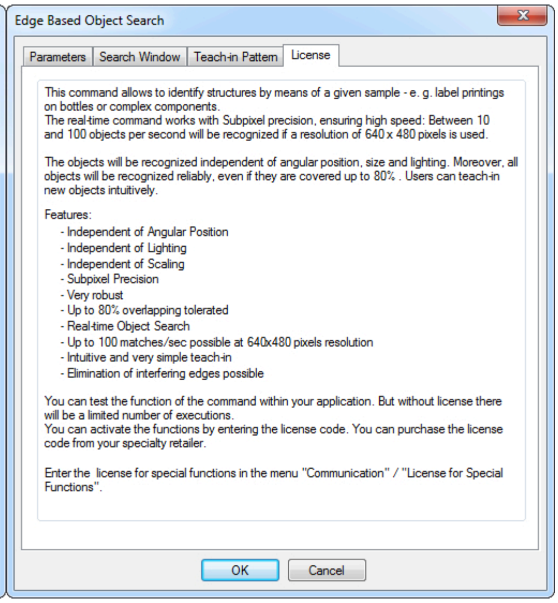


Image 103: Dialogue Edge Based Object Search, tab control License

With **Locate > Edge Based Object Search**, you can find objects in the camera's image. To do this, you must first teach in a reference object digitally as a pattern. The locate algorithm then searches the image to be tested looking for corresponding edges and can then identify the object in this way within the search area. When doing so, the position as well as the angular position in relation to the taught-in pattern is determined. Use the command to find and classify one or more objects in the image.

The Edge Based Object Search is a very robust and fast object recognition process which can also find multiple objects in the image. It is independent of lighting and angular position as well as being scaling invariant up to +/- 50% of the original size and it can even recognise patterns with overlaps of up to 80%. Further strengths are the subpixel-accurate search (+/- 0.1 pixels and +/- 0.3 degrees, depending on the size of the pattern), the real-time object search (up to 100 hits/s possible at 640x480) and parameter-based elimination of interfering edges.

Searching takes place in two phases, first as a rough preliminary search and then as a detailed search at a higher resolution. The accuracy and the processing speed of the search depend heavily on the parameters specified such as the search angle step increment, the size of the area to be searched and the size of the pattern to be compared, among others. Higher search accuracy always means longer processing.

Use of the edge-based object recognition feature is based on a licence model. You can test the function of the command from within your application. However, you can only run the function a limited number of times. When you enter the licence code, the command is released and can be used with all its parameters and configuration options. You can purchase the licence code from Vision & Control GmbH and can then enter it via the **System Preferences > Licence for Special Functions** menu.

Brief Guide

1. Select in the tab control **Parameters** under **Pattern Source,** whether you want to teach the pattern itself (go to step 2) or use a in the in the vision system stored pattern (>> *"Save Pattern", page 235*) (select the pattern in the selection list and proceed with step 5).
2. Go to the tab control **Teach-in Patterns**. Activate the teach-in mode by double-clicking the right mouse button and switch to the video image. Change the size and position of the teach-in window with the controls pf your mouse (>> *"Teaching in Test, Detection and Search windows", page 26*).
3. Close the teach-in mode by double clicking with the right mouse button and then click the [**Read Pattern**] button.
4. In the dialogue put a checkmark at the edges, that should be taught to recognize the pattern (*see "Image 104: Select Edge Segments dialog"*). Refer to the red marks of the edges in the Monitor Window. If there is only one edge found by the teaching of the pattern, it is automatically selected and the dialog box will not even appear.
5. Switch to the "Search Window" tab and define the position tracking settings (>> *"Position Tracking of Objects", page 35*).
6. Activate the teach-in mode with double click the right mouse button and change the size and position of the search window with the controls (>> *"Teaching in Test, Detection and Search windows", page 26*). Close the teach-in mode with a double right-click with your mouse.
7. Switch to the "Parameters" tab and adjust the input and evaluation parameters (see next section).
8. Test the command with the [Test] button and then modify the parameters set in steps 1-7 until "No errors" is shown as the test result in the bottom line.
9. Add the command to the test program with the [OK] button.

Selecting edge segments

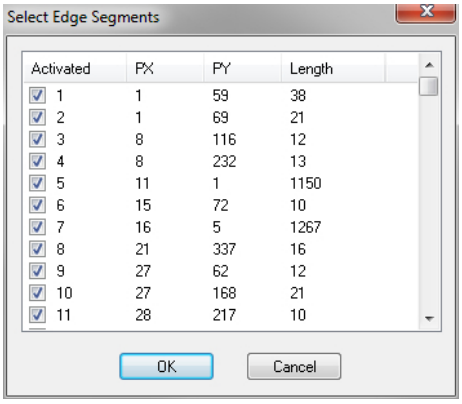


Image 104: Select Edge Segments dialog

In the "Select Edge Segments" dialog, you can specify which edges are to be taught in as object edges. In this way, you can remove irrelevant edges from the pattern for object recognition so that the test only searches for the edges that you want. The edges that are found are coloured red in the monitor window. The red marking is removed as soon as you deactivate an edge. If you select an edge in the list, it will be highlighted in red bold in the monitor window. You can select and display multiple edges by holding down the Ctrl key.

As standard, all edges that are found will be listed in the dialog sorted by their lowest X-coordinate. The Y-coordinates and the edge lengths are also shown. The edges that are found can be sorted by any parameter as desired. In the first column "Activated", you can specify whether the respective edge is to be taught in (checked) or not (not checked).

ADVICE

The "Select Edge Segments" dialog is only available when teaching in a pattern.

Input parameters

Parameter	Description
Resolution Reduction	Reduces the resolution of the original image, thereby making the search faster but less precise. The parameter values range from 0 = original resolution to 5 = greatly reduced resolution and represent the value (2^{2n}) of the reduction of the number of pixels. You can check the resolution that is set by means of the monitor window.
Edge Point Reduction	Reduces the number of edge points used for the search. Reducing edge points accelerates searching but might make it too imprecise, under circumstances, if parameter values are too high (7 or greater).
Pattern Source	Specifies whether the pattern is to be taught via the "teach-in patterns" tab or as an existing pattern file saved on the system. Gibt an, ob das Muster über den Reiter "Muster einlernen" eingelernt werden soll oder ein bestehendes, als Datei abgelegtes Muster verwendet werden soll (>> *"Save Pattern", page 235*).

Parameter		Description
Pre-Search		Specifies the precision with which the first search level searches for the pattern. • Coarse - preliminary search with the standard rough search pattern • Fine - preliminary search with a detailed search pattern, however with longer processing time
Fine Search		Specifies the precision with which the second search level searches for the pattern. • Regular - standard detailed search • Subpixel - search pattern with high subpixel and angle precision, however with longer processing time
Search Angle	From - To	Specifies the boundaries of the angle search area in which the pattern is searched for [-180 to 180°]. A smaller angle search area and, with it, a significantly shorter processing time is especially useful when, due to the test process, the object is only added to the test in a certain angular position or range of angular positions.
	Step	Specifies the angle step increment with which the angle search area is to be searched. A greater step increment allows faster processing while a smaller one increases the precision of the search. With the angle step increment, you also specify the maximum search precision at the same time.
Scaling	Minimum	Allows the object to have a smaller scale [-50 to 0%] in the image in comparison with the taught-in pattern.
	Maximum	Allows the object to have a larger scale [0 to 50%] in the image in comparison with the taught-in pattern.
	Step	Percentage value specifying the step increment in which the scaled object is searched for. A greater step increment allows faster processing while a smaller one increases the precision of the search.
Grade of Match		Percentage value specifying the minimum level with which the found pattern must correlate to the taught-in pattern in order for it to be processed as a valid pattern. You must specify or adjust the value according to the test results. The grade of match is not affected by the brightness in the test window.
Sort Order	Grade of Match	Specifies that the found objects (and therefore also their result values) are to be sorted by their degree of match.
	Direction	Specifies the direction in which the objects found in the image (and therefore also their result values) are to be sorted.

Parameter		Description
Number of Objects Found	Only First Match	Specifies that only the first found object will be evaluated that corresponds to the grade of match. This means that the object search process ends as soon as a corresponding object is found.
	Nominal	Specifies how many objects are to be found and evaluated. The entire image is examined and the search is not terminated prematurely.
	"+" / "-"	Specifies the permissible upper (+) and lower (-) deviations of the found objects from the nominal value.
	RES	Result number in which the number of patterns found is to be saved.

Evaluation parameters

Parameter	Description
Image/World	Specifies whether the geometries are to be saved as image or world coordinates.
Point	First point number for storing the centres of gravity of the found patterns.
Line	First line number for storing the angular position lines of the found patterns.
Angle	First result number for storing the angular positions of the found patterns. The angle is specified as an integer value with 3 decimal places (100° → 100000).
Grade of Match	First result number for storing the grades of match of the found patterns.

6.2.16 Locate Contour

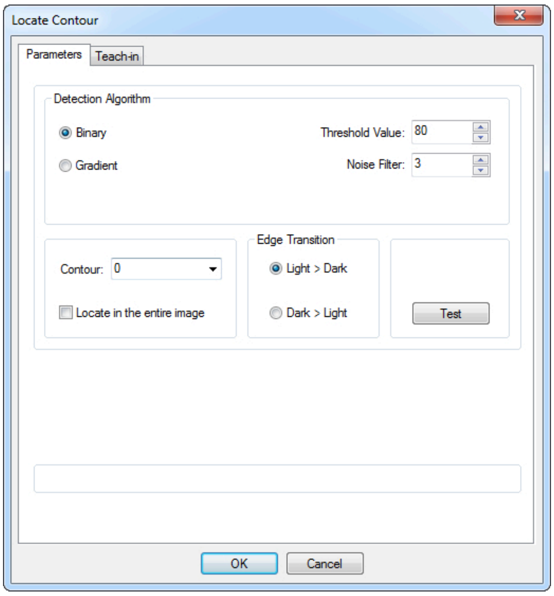


Image 105: Dialogue Locate Contour, tab control Parameters

You can teach in a command for finding a contour within the image using **Locate > Locate Contour**. A contour is a series of neighboring points. The command always finds the first contour in the direction of detection. At the bottom left is a checkbox used for setting whether the points of the contour are located until the edge of the window is reached (open contour) or until the contour meets itself again (closed contour).

The found contour is buffered in a contour buffer. You can select between a gradient algorithm and a binary algorithm for locating contours. The maximum size of a contour and the number of contours to be saved depends upon the system.

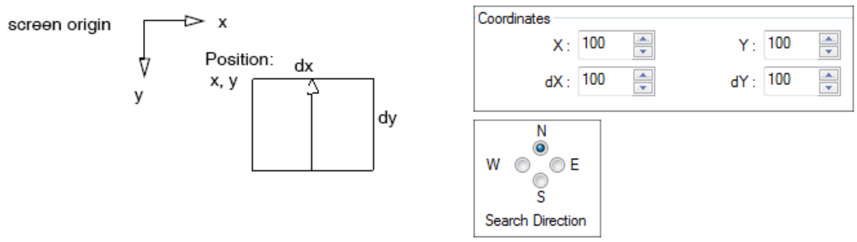
Brief Guide

1. Define the following parameters in the **Teach-in** tab:
   - Position tracking: no, X point, Y point
   - Search direction
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the detection window in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Specify the following in the **Parameters** tab:
   - Detection algorithm
   - Edge transition
   - Contour no.
   - Contour location in the image or in the contour window (closed/open contour)
6. Test the command with the [Test] button]:
   Modify the parameters set in points 1-5 until the test result is error free.
7. Insert the command into the test program with the [OK] button.

Teaching-in the Detection Window

After calling the **Locate > Locate Contour** function, you will see the dialog box for finding a contour in the **Teach-in** tab and a detection window on the video screen.

Variant 1: Entering Parameters in the Edit Mask



Enter the X and Y values for the starting point of the detection window and its dimensions (dX, dY) into the fields. Change the values as required with the arrows on the right hand side. Check the changes made to the detection window on the video screen.

The **Search Direction** determines the direction of detection. In the above example it will be from south to north i.e. upwards.

ADVICE

The search direction can only be changed within the dialog box. It cannot be changed on the video screen.

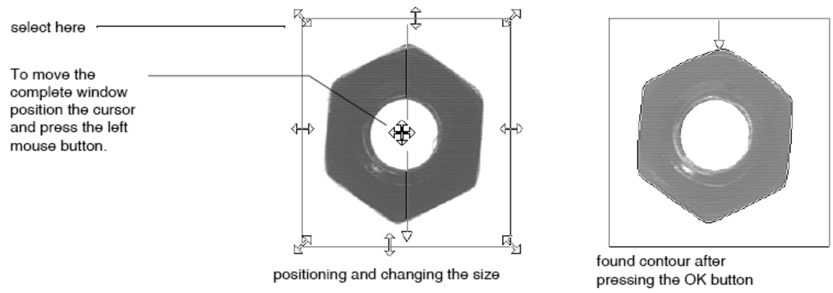
Variant 2: Setting Parameters on the Video Monitor

Double click the right-hand mouse button to switch from the **Teach-in** tab to the video screen. If you move the mouse pointer over the perimeter of the detection window, it will turn into a small cross or double-headed arrow.

- at the corners... to resize the window in diagonal direction
- from the edges … to move the window edge horizontally or vertically

To move the detection window around the screen, position the mouse cursor in the center of the window and hold down the left mouse button while moving the mouse.

Double clicking with the right mouse button switches back to the edit window in the **Teach-in** tab.



Tracking the Position

The detection window can either be at the same image location (no position tracking) or it can follow a reference object.
For position tracking: >> *"Position Tracking of Objects", page 35*

Input Parameters

Detection Algorithm
Set suitable parameters interactively with the aid of the [Test] button.

Detection Algorithm	Input Parameters / Description
Binary	Threshold value, noise filter
Gradient	Gray value threshold, width of detection, min. threshold value
>> *"Principles of Detection", page 417*	

Edge Transition
The type of transition can be either light-dark or dark-light.

Edge Transition	Input Parameters / Description
Light -> Dark	Only light-dark transitions are included.
Dark -> Light	Only dark-light transitions are included.
Locate in the entire image?	If the option is deactivated, contours are only followed until the edge of the search window. Open contours can be detected with this setting. Otherwise, only closed contours can be detected.

Evaluation Parameters

Evaluation Parameters	Description
Contour	Number or name of the contour buffer used to store the found contour.

6.2.17 Locate Circle

Overview

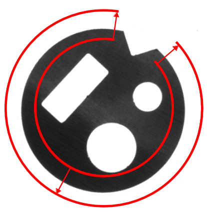


Image 106: Detection Window at position

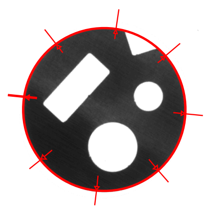


Image 107: composed circle with 8 detection points

You can teach in a command for finding a circle or fractions of a circumference within the image using **Locate > Locate Circle**. The command relates to circular contours within a detection window in the image. The direction of detection is shown by the arrowheads.
A circle found within the detection window is stored as a geometrical circle element. The number of circles to be saved depends upon the system.

Work Flow
Teach-in

1. Switch to the tab control "***Teach-in***".
2. Set the position of the detection window as well as the angles and radii (see also: *"Teaching in Test, Detection and Search windows", page 26*). In this area points are searched along detection beams and the circle formation is made.

3. Determine the number of detection points (single detection). These are distributed evenly over the detection window.
4. Determine whether all of the detection points must be found so that a circle can be formed from the single detection.
5. If necessary, set the parameters (X Point, Y Point) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow
Parameterizing and Testing

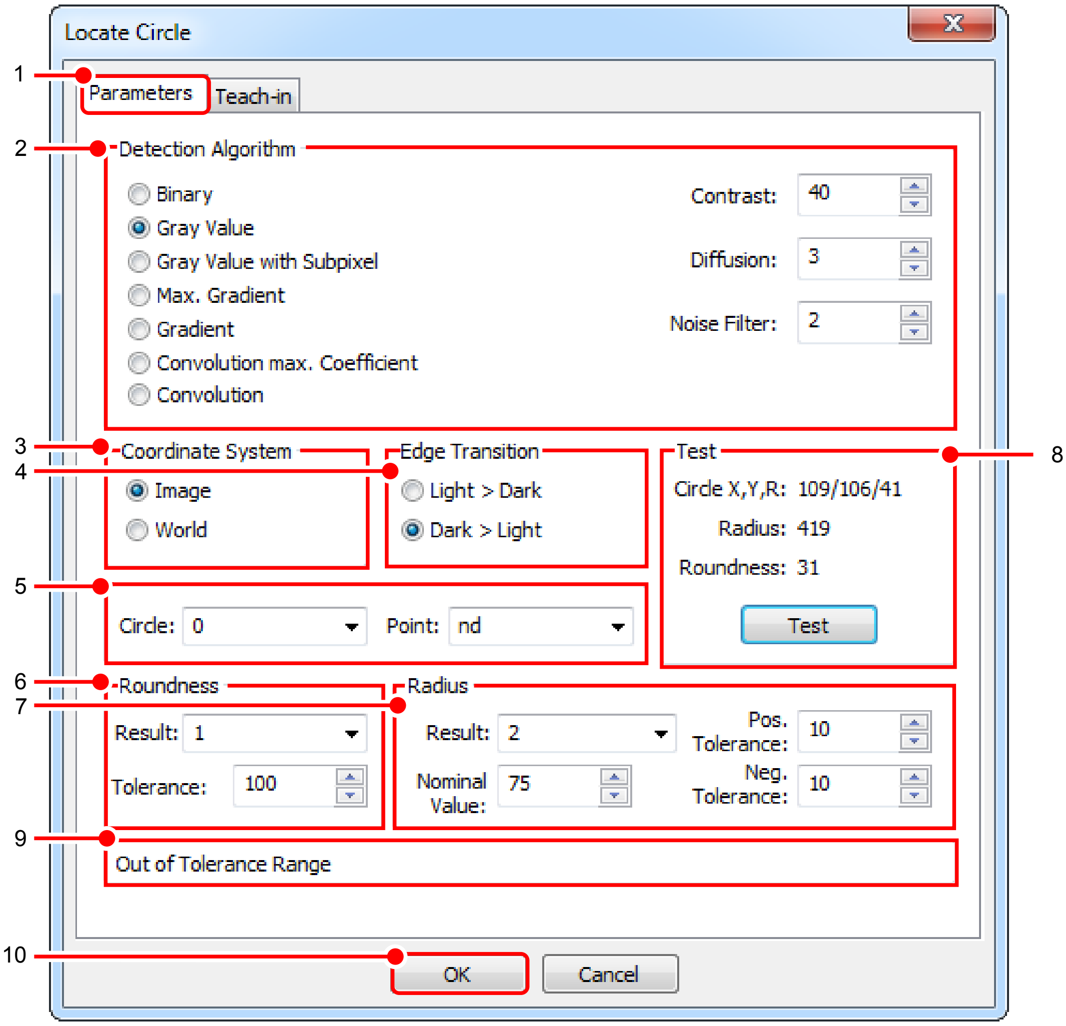


Image 108: Dialog Locate Circle, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Set here the detection algorithm. You will find further information at *"Principles of Detection", page 417*.
3. In the area "Coordinate System", set whether you want to save the circle and point in image or world coordinates.

Parameter	Description
Image	The storage of the values of the circle is carried out in image coordinates.
World	The storage of the values of the circle is carried out in world coordinates.

4. In the area "Edge Transition", specify the type of an edge along the detection direction in the window as light to dark or dark to light.

Parameter	Description
Light > Dark	Only edge transitions from light to dark will be considered.
Dark > Light	Only edge transitions from dark to light will be considered.

5. Set here the options for saving the detected circle.

Parameter	Description
Circle	Number or name under which the coordinates of the found circle are stored in the circle list. (Coordinates: X,Y,R)
Point	Number or name under which the coordinates of the found circle center point are stored in the point list.

6. Here you can specify the storage options for the roundness of the circle. The roundness is the difference between the largest and smallest radius (relative to gravity).

Parameter	Description
Result	Number or name under which the roundness is stored in the result list.
Tolerance	Allows an upper and lower deviation from the result.

7. Here you can specify the storage options for the radius of the circle found.

Parameter	Description
Result	Number or name under which the radius is stored in the result list. The value of the radius is always stored in world coordinates.
Nominal Value	Nominal value of the radius.
Tolerances	Allows an upper and lower deviation from the nominal value.

8. Testing and setting the appropriate parameters
   By activating the [Test] button the command with the set parameters is executed. The values of the circle found will be displayed in the docking window Circles. If declared, the result of the roundness is shown in the docking window Results and in the dialog (above the [Test] button). In parallel, the composed circle with the corresponding detection points is shown in the video image.
   Change the settings in step 2 until the result meets your requirements. If necessary, switch to the tab control "Teach-in" and adjust the number of detection points.
9. Evaluation Area
   Eventually error messages are shown here.
10. Use the [OK] button to insert the command into the test program.

6.2.18 Mask Check

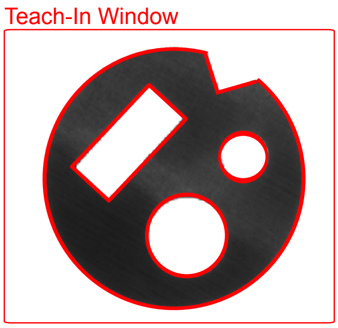


Image 109: Mask Check, contour of the taught mask

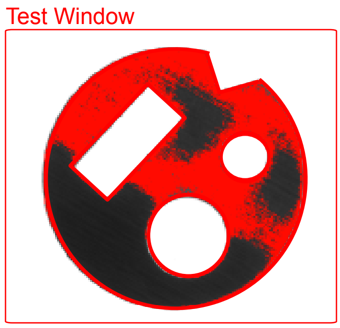


Image 110: Mask Check, Count Pixels with gray value range 50-70

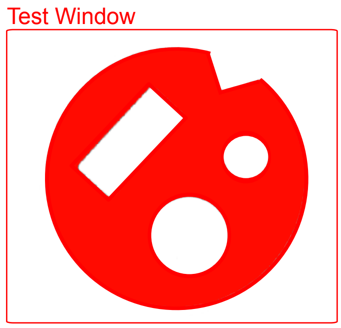


Image 111: Mask Check, Count Pixels with gray value range 0-80

Use **Locate > Mask Check** to test complex structures for completeness and accuracy. In contrast to the "**Count Pixels**" command, the structures do not have to be contiguous.

The command counts the pixels in the test image that are covered by a freely definable mask. The mask is a binary image that can either be loaded as a file or taught-in in the dialog. Depending on the definition of the color of the mask (light or dark), the pixels are counted that lie under either the black (dark) part or white (light) part of the mask.

The command permits the covered part of the image to be saved to a separate image memory page. The copied image content can then be further processed with other algorithms.

There is always one mask taught-in per command.

Define Mask

You have two options to define a check mask.

On the one hand, you can teach-in a mask; on the other hand you can use an image file already existing on the vision system as a check mask. The latter variant is useful if the test mask should be re-taught or new-through by a separate subroutine (via overwriting the image file by the **Save Image > Image Section** command).

Work Flow
Teach-in Mask

1. Switch to the tab control "***Teach-in Mask***".
2. Here you can determine, by means of the geometry rectangle, which image content has to be read as the check mask. (Information for teaching, see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. Enter the gray value threshold for the binarization of the image.
4. With the button [Read Mask] you read-in all image information located in the geometry as the mask, and stored it internally in the command. The detected mask is shown as a thumbnail in the reference window.
5. In the reference window, verify if the mask, now converted into a binary image, is complete. In the video image the edges of the mask are shown in red at the same time.
6. In case of an incorrect mask, change the gray value threshold until the mask fits your needs.

Work Flow
Use External Mask

Preparation: Image File as a Mask

You have the option to use a mask from an image file, in which a image section was saved. These must be stored on the vision system. Make sure the size of your mask is smaller than the size of the test image.

- **Image Section from an Image**
  Using the Command **Control > Save / Load Image** you can save an image section in the image directory of the vision system.
- **Externally processed image (optional)**
  With the option **Utilities > Receive Image from Vision System / Send Image to Vision System** you can send an Image to your PC. Here you can edit it with an image editing program and then transferred it back to the vision system by using **Send Image to Vision System**.

Select External Mask:

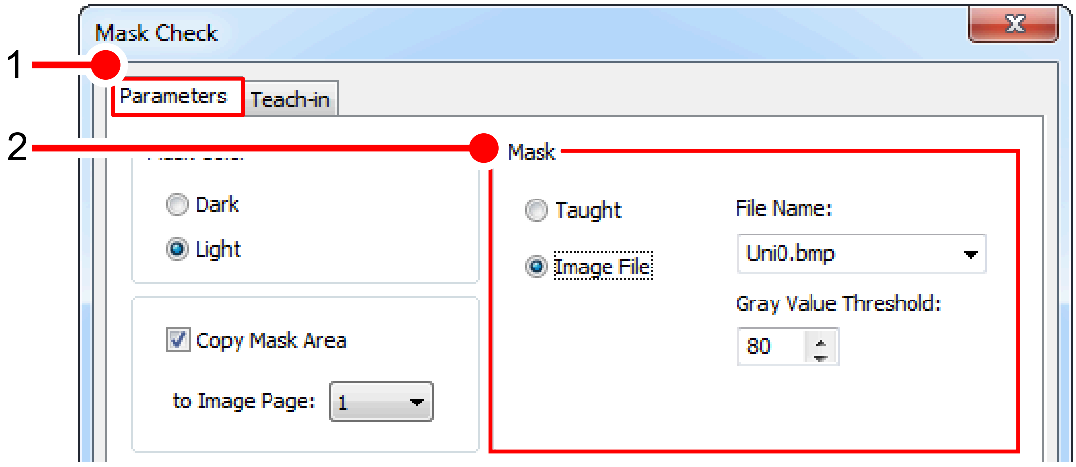


Image 112: Dialog Mask Check, tab control Parameter

1. Switch to the tab control "***Parameters***".
2. In the area "Mask" choose the option "Image File".
   - Select from the drop-down list the appropriate image file.
   - Enter the gray value threshold for the binarization of the image.

Work Flow
Define Test Area

1. Switch to the tab control "***Teach-In***".
2. Set here the position of the test window, the size is already given through the window of the mask (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also *"Position Tracking of Objects", page 35*).

Work Flow
Parameterize and Testing

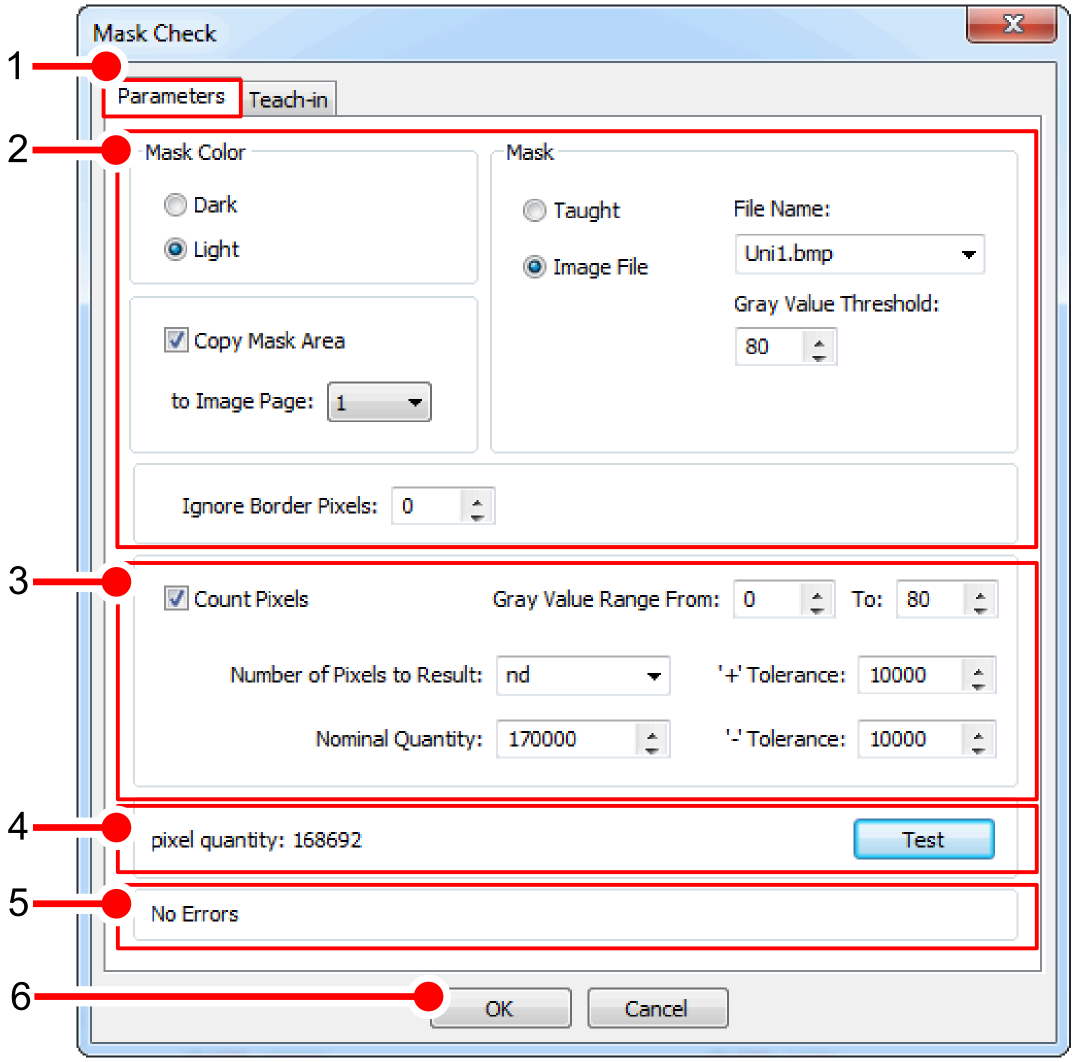


Image 113: Dialog Mask Check, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Adjust the setting parameters for the use of the mask.

Mask Color

Parameter	Description
Dark	Pixels that are covered by the black part of the mask are checked.
Light	Pixels that are covered by the white part of the mask are checked.

Mask

Parameter	Description
Taught	The mask that is taught-in with the control tab "**Teach-in Mask**" is used. It is shown as a thumbnail in the reference window to the right.
Image File	An image file that is located in the vision system is used. The specified image file is converted to a test mask. The required threshold is specified under **Gray Value Threshold**.
Filename	Only available if the option **Image File** is active. Choice of the image file with the image section.
Gray Value Threshold	Only available if the option **Image File** is active. Indicates the binary threshold for extracting the test mask from the image file. We recommend using the image report function to provide the gray value threshold required.

Copy Mask Area

The command Mask Check copies the image that is determined by the mask (regardless of its location) to the center of the target page. In combination with position tracking images can be brought in to a stable position in which tests can then be carried out without further tracking.

Parameter	Description
To Image Page	Specify the image page, to which the copied and (if required per phi line position tracking) rotated image should be sent to for further processing.

Border Pixels

Parameter	Description
Ignore Border Pixels	When using **Count Pixels** and **Copy Mask Area** only pixels that have at least this defined distance from the edge of the mask are counted.

3. Adjust here the parameters for the testing and evaluation of the mask.

Parameter	Description
Count Pixels	The pixels are counted, within the set Gray Value Range, in the region of the window covered by the part of the mask that corresponds to the set mask color.
Gray Value Range	Only pixels with gray values within the gray value range are counted.
Number of Pixels to Result	Here you enter the number or name of the result in the result structure in which the determined pixel number and the good/bad evaluation of the nominal quantity (incl. tolerances) should be saved.
Nominal Quantity and Tolerances	Here you set the nominal quantity of pixels to be found. If the number of pixels found lies within this range, the command ends with a good result (otherwise with bad).

4. Testing and Evaluate suitable parameters
The command is executed with the parameters by activating the [Test] button. The quantity of pixels found is shown in the test area. In parallel, the determined pixels are marked red in the video image.

5. Here you can see the good/bad evaluation of the command, notes for parameterization (tolerance range exceedance) as well as error messages.

   Repeat step 2 to 4 until the result meets your requirements.

6. Use the [OK] button to insert the command into the test program.

Result Geometry Variables

In the docking windows geometry lists the following values are given:

Docking Window	Description
Results	Determined quantity of pixels (if Count Pixels are activated) and good/ bad rating (if nominal Quantity and Tolerances are activated) according to nominal quantity and tolerance.

6.2.19 Save Pattern

With **Locate > Save Pattern** you write a command for recognizing of a pattern into your test program. The pattern is stored as a file on your vision system.

After inserting this command into the test program, patterns can be taught in and stored separately from the program without vcwin. The number of stored patterns depends of their size and the amount of memory available on the vision system.

Existing image files with the same names will be overwritten. All existing patterns can be displayed and deleted with the function **System Preferences > System Resources / Initial Program**.

ADVICE

For vision systems of the type pictor MxxE, the patterns must be saved on the vision system's flash, to keep them after the vision system is switched off. This can be done by the **System Preferences** menu > **Save File System in Flash** or **Direct code Input** with FB.

You can use the stored patterns in the following commands as a reference pattern (option: Pattern File):

- **Locate > 360° Pattern Search**
- **Locate > Edge Based Object Search**
- **Locate > Search Pattern**
- **Obsolete > Advanced Pattern Search**

You can assign a pattern to multiple search commands in a program or to different programs.

Brief Guide

1. Define in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Assign a name to the pattern in the **Parameters** tab.
3. Insert the command into the test program with the [OK] button.

Teaching-in Pattern Window

Position and Appearance of the Pattern Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the pattern window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The pattern window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input Parameters

Pattern Name	Description
Name of Pattern in the Vision System	Name under which the pattern is saved in the vision system.

Test

Test the command with **Utilities > Test Step**, and check with **System Preferences > System Resources / Initial Program** that the pattern has been successfully stored.

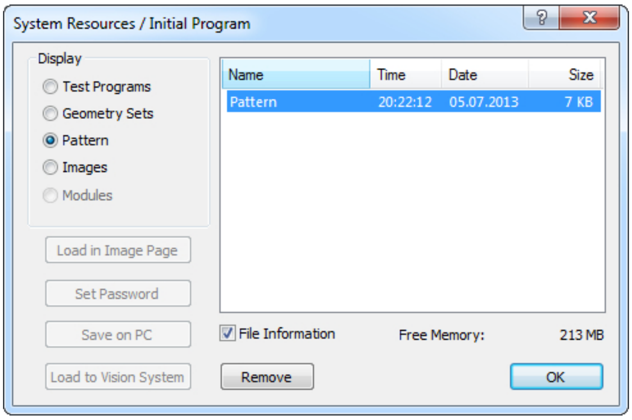


Image 114: Dialog System Resources / Initial Program - saved pattern

6.2.20 Search Pattern

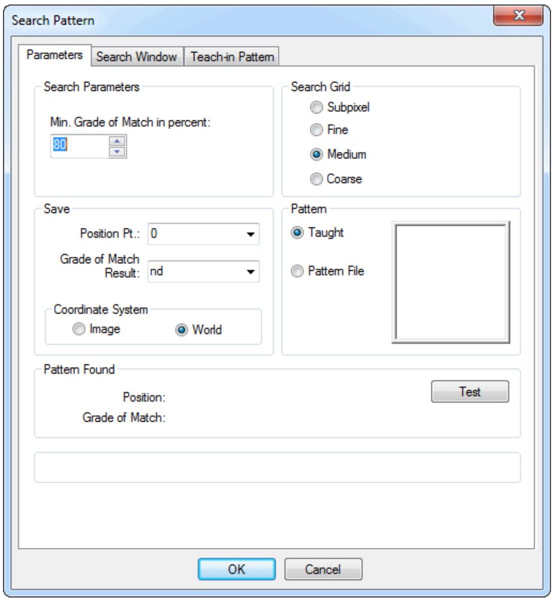


Image 115: Dialogue Search Pattern, tab control Parameters

You can teach in a command for detecting patterns within the image using **Locate > Search Pattern**. The pattern is sought within a pattern window which must not contain the pattern within a test window. You can use the command to determine both the position of a reference pattern to be determined in a search window, as well as making statements about the quality or presence of a corresponding pattern based on a respective pattern object.

Brief Guide

1. Define in the **Search Window** tab:
   - Shape of the test window
   - Position tracking
2. If no pattern already loaded on the vision system is to be used, teach in the reference pattern in the **Teach-in Pattern** tab using the following steps:
   - Double click the right-hand mouse button to activate the teach-in mode.
   - Modify the size window in the video image using the arrow controls, then position the pattern window within the search window.
   - Double click the right-hand mouse button to deactivate the teach-in mode.
   - Teach in the pattern using the [**Read Pattern**] button.
3. Specify the following in the **Parameters** tab:
   - Minimum grade of match of the found pattern to the taught pattern
   - Search grid
   - Saving of position of point number; grade of match
   - Pattern (taught or load from vision system)
   - Coordinate system pattern found (grade of match, position)
4. Test the command with the [**Test**] button:
   Modify the parameters set in points 1-6 until the test result is error free.
5. Insert the command into the test program with the [**OK**] button.

Teachin-in Patterns

Position and Appearance of the Teach-in Pattern Window

After double clicking with the right mouse button on the **Teach-in Pattern** tab, you can position and adapt the teach-in pattern window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Teaching-in the Search Window

Position and Appearance of the Search Window

After double clicking with the right mouse button on the **Search Window** tab, you can position and adapt the search window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The search window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

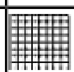
Input Parameters

Search Parameters	Description
Min. Grade of Match	Minimum compliance (correlation) of the pattern found with the taught pattern in per cent, so that a pattern may be further processed as a valid pattern. • You can set the value directly. The values must be changed if testing is unsuccessful. • The degree of match is not influenced by the brightness in the test window. • 100% means exact compliance, 0% means no compliance. • Useful settings are 60 to 80%.

Search Grid	Description
Define the degree of precision of the search process by the search grid.	
Fine	Fine search, e.g. for writing or gravure.
Medium	Search for medium-fine structures.
Coarse	Coarse search, e.g. for a rectangle without fine structure.

Strategy for Setting Search Grid

- For time-critical applications, always search for the pattern with Coarse first.
- If Coarse is not successful in the [Test], activate Medium or Fine.

Function	Sensitivity	Speed	Remarks	Sketch
Fine	Grid search 2 pixels	-		2 x 2
Medium	Grid search 4 pixels	4 x faster than fine		 4 x 4
Coarse	Grid search 8 pixels	16 x faster as fine	quickest but least accurate algorith	 8 x 8

There is an internal exact pixel search after each search procedure:

Function	Sensitivity	Speed	Remarks	Sketch
Internal	exact pixel search	time added to fine/ medium/coarse	seeks structure around the position found with fine/medium/ coarse	1 x 1

Defining and Selecting the Reference Pattern

Options for Defining the Reference Pattern

You have two options for defining the reference pattern:

- Defining the pattern in the Teach-in Pattern tab
- Using a pattern stored on the vision system

Option 1: Reading the Pattern in the »Teach-in Pattern« Tab

You will see the dialog box for finding a pattern in the **Teach-in Pattern** tab and a rectangular detection window on the video screen. >>*"Teaching in Test, Detection and Search windows", page 26*
The pattern window can have a maximum size of 256 × 256 pixels and must be located in the test window. Using the [**Read Pattern**] button you can read the image information within the pattern window. The reference pattern detected is transferred as a miniature into the display and stored internally as a nominal pattern.

Option 2: Using a Pattern Stored on the vision system

The prerequisite for this is that a pattern is loaded onto the vision system with **Locate > Save Pattern**. >> *"Teaching in Test, Detection and Search windows", page 26*

Check the pattern saved on the vision system using **Communication > System Resources / Initial Program**.

Setting a Reference Pattern in the »Parameters« Tab

During teach-in you set the reference pattern in the **Parameters** tab with the **Pattern** option:

»Parameter > Pattern« Option	Description
Taught	The pattern taught with **Teach-in Pattern** is used.
Load File from Sensor	The pattern stored on the vision system is used.

Evaluation Parameters

Save	Description
Position Pt.	Point number for saving the position of the pattern.
Grade of Match Result	Gray value compliance (correlation) of the pattern found with the taught pattern in per cent.
Image/World	Saving in image coordinates or world coordinates.

Test

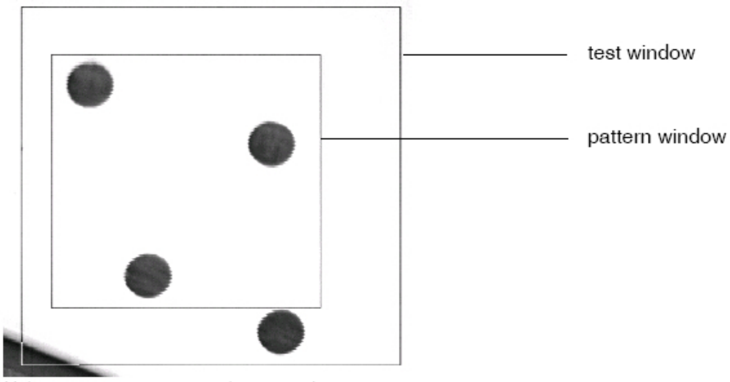
Pattern Found

The test displays:

- Position … Position determined for the pattern found.
- Grade of Match … Gray value compliance (correlation) of the pattern found with the taught pattern in per cent.

Example

Three points serve as the reference pattern in the example.



Video image: pattern search • example

6.2.21 Inspect Surface

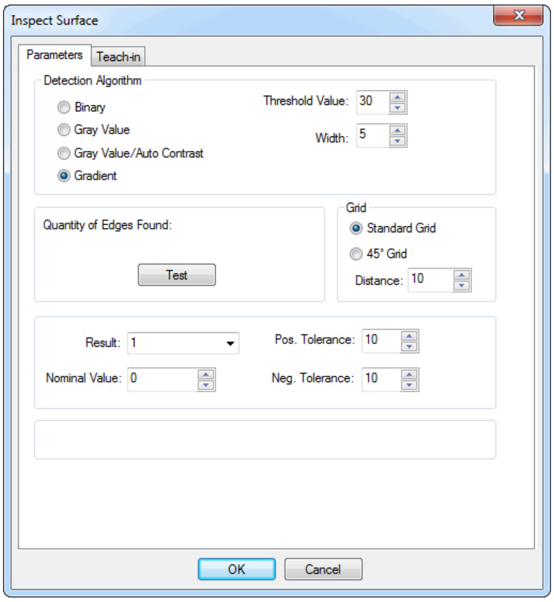


Image 116: Dialogue Inspect Surface, tab control Parameters

You can insert a command which tests the quality of a surface by detecting flaws using **Locate > Inspect Surface**. Defects/flaws are recognized by searching for transitions using a grid within a test window. Select between one of two grid types and set the width of the grid corresponding to requirements. The quantity of edges found is used as a quality criterion and is stored in the result structure.

Brief Guide

1. Define the following parameters in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the search window in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Specify the following in the **Parameters** tab:
   - Detection algorithm
   - Grid direction and distance
   - Result
   - Nominal value (nominal number of edges)
   - Tolerances (quality criterion)
6. Test the command with the [Test] button and modify the parameters set in points 1 to 5 until the test result is error free.
7. Insert the command into the test program with the [OK] button.

Teaching-in the Test Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

In angular position tracking only the center of the window is rotated. The rotary angle is produced from the angle between the position tracking line and the X axis. The test window's direction and size remain intact during angular position tracking.

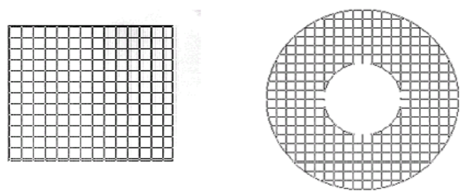
Input Parameters

Set suitable parameters interactively with the aid of the [Test] button. The number of transitions found determines the quality of the surface.

Detection Algorithm	Description
Binary	The Binary algorithm recognizes transitions along the grid when a predefined gray value threshold is reached. Parameters: threshold value, noise filter
Gray Value	The Gray Value algorithm recognizes transitions along the grid by comparing the actual contrast with a nominal contrast. Parameters: contrast, diffusion, noise filter
Gray Value / Auto Contrast	This is similar to Gray Value above, but differences in the color/brightness of objects or even fluctuations in lighting conditions can be taken into account. The mean gray value within the test window is determined. The contrast here is the deviation allowed from this value (expressed as a percentage) without a flaw being registered. Application: when there is very low contrast and strong brightness variations. Parameters: percentage deviation from mean gray value, diffusion, noise filter
Gradient	The Gradient algorithm recognizes transitions along the grid using the gradient method. Parameters: threshold value, width

Grid Direction and Width

Set the raster grid for detection with **Raster**, either parallel to the X/Y axis (left selection button) or in a 45° angle (diagonally) to the X/Y axis (right selection button). The optimum raster grid spacing must be determined through tests.



Examples: Standard grid in rectangular test window (left) and ellipsoid ring test window (right)

Evaluation Parameters

Parameters	Description
Result	Result number for storing the quantity of edges found.
Nominal Value and Tolerances	Nominal number of edges and the deviations (quantity of edges) allowed.

Test

The **quantity of edges found** is displayed in the test.

6.2.22 Search and Identify Objects

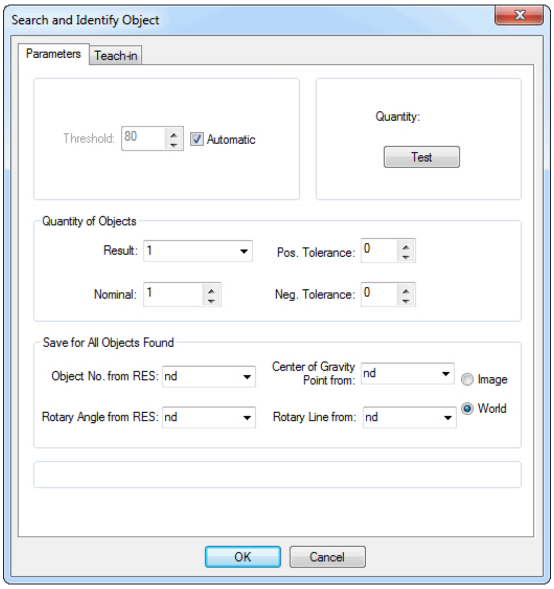


Image 117: Dialogue Search and Identify Objects, tab control Parameters

With **Locate > Search and Identify Object** you add a command to your test program that search and identify a previously teached-in object. See also: *"Teach-in Objects", page 244*

Brief Guide

1. Set the threshold for the search in **Threshold**. If vcwin should automatically set the threshold, set the **Automatic** option.
2. Set the number of objects to be searched for in **Quantity of Objects** and set the tolerance range for the quantity of objects within the object search that should be taken as successful.
3. Use the **Save for All Objects Found** section to set in which elements of the results structure the objects found and the rotary angles produced should be saved.

Teaching-in the Search Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

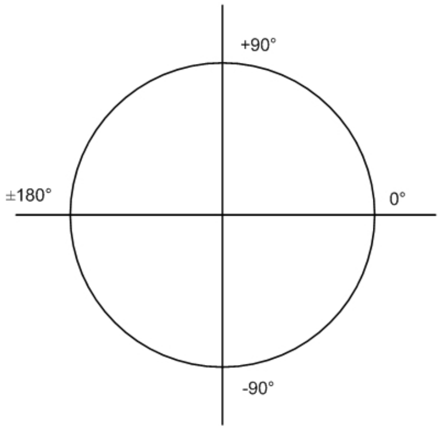


Image 118: Angular Position Output

6.2.23 Teach-in Objects

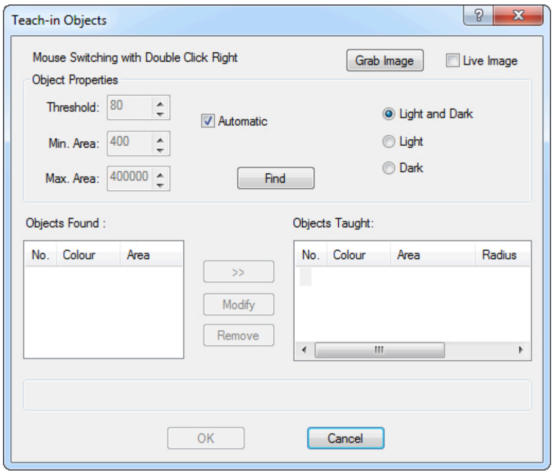


Image 119: Dialogue Tech-in Objects

Insert a command for teaching-in objects and for configuring the properties and tolerances for every single object that should be inspected using **Locate > Teach-in Objects**.

Brief Guide

1. Set up the video image.
2. Use the [**Grab Image**] button to acquire an image that contains the objects required.
3. Teach-in the required objects, one after the other, as follows:
   - Enter the investigation parameters and the type of object (light and dark, light, dark) in **Object Properties**.
   - Double click the right-hand mouse button to activate the teach-in mode.
   - Modify the size and position of the detection window in the video image using the arrow controls. The detection window must enclose the object.
   - Double click the right-hand mouse button to deactivate the teach-in mode.
   - Click the [**Find**] button: The object is read-in and added to the list of objects found.

4. Mark the objects in the left hand list of Objects Found and transfer them to the right hand Objects Taught list with the [**>>**] button.
   This transfers the objects from the vision system to vcwin.
5. Mark the entries in the right hand Objects Taught list, one after the other, and use [**Edit**] to set the individual test parameters and tolerances required.
6. Use [**OK**] to Teach-in the objects in the command to the test program. This simultaneously loads the objects back into the vision system.

Grab Image

At first do an image acquisition using the [Grab Image] button.

Live Image … Switch between live image and memory image

Teaching-in the Search Window

After double clicking with the right-hand button in the **Teach-in** tab, you will see a rectangular detection window on the video screen. This window cannot be numerically set up, but only set up in the video image. >> *"Teaching in Test, Detection and Search windows", page 26*

Input Parameters

Object Features	Description
The Object Properties decide the threshold of the Binary algorithm and the permissible areas for the objects to be searched for.	
Automatic	When the distribution into light/dark objects is clear, this option should be activated. The threshold is automatically defined during execution.
[Find] button	The [Find] button automatically determines the gray value range for the threshold. Further properties and the tolerances are taken up by the Object Properties. Smaller objects must be enclosed by the teaching window at first. After teach-in they can be enlarged by later program processing into usable search windows.
Threshold	The gray value threshold for object recognition can be changed.
Min./Max. Area	Determines how many pixels the object is allowed to contain in order to be recognized. Objects with more or fewer pixels are ignored.

Colors	Description
Light and Dark	The colour of the objects is brighter or darker than the surrounding brightness.
Light	The colour of the objects is brighter than the surrounding brightness.
Dark	The colour of the objects is darker than the surrounding brightness.

Transfer List

The list of Objects Found is on the left hand side of the dialog window.

You can transfer the objects found into the right hand part of the dialog window by clicking on the [>>] button. The [Clear] button removes entries. The [Edit] button allows the test parameters of objects entered with teach-in to be defined and the tolerances for the objects to be set.

Parameters and Tolerances of the Objects

The parameters and tolerances of all the objects entered with teach-in can be tested and changed in the **Object Properties** dialog box. This dialog box can be opened using the [Edit] button.

Parameters	Description
As Object Number	Number of object.
Area	The area of the object in pixels or measurement units, depending on the coordinate type.
Min./Max. Radius	Smallest/largest radius of the object contour.
Angle between Radii	Angle between the smallest and largest radii.
Grade of Match	Correspondence to the outer contour in %.
Offset Angle	Offset for the angular position (1/100 degree). If the object already had an initial rotation during the image acquisition, the offset can be added in order to relate the determined angular position to the X or Y coordinate.
Color	The color of the object (only relevant with color cameras, otherwise black).
Covering Area	Correspondence to the outer contour in %.
Min./Max. Diameter	Smallest/largest radius of the object contour.
Angle between Diameters	Angle between the smallest and largest radii.

6.2.24 Count Pixels

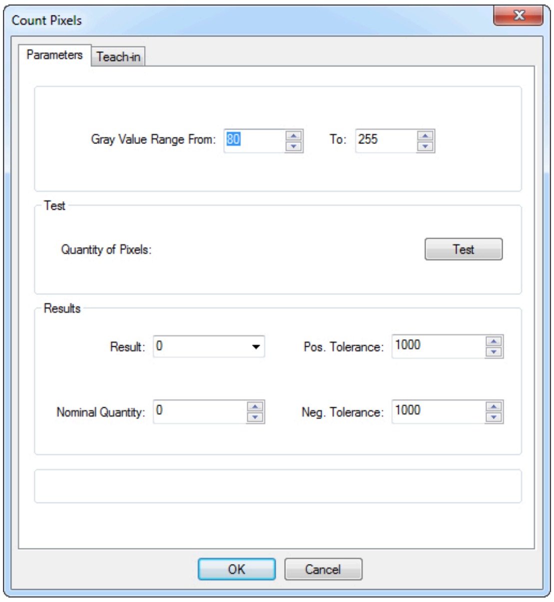


Image 120: Dialogue Count Pixels, tab control Parameters

You can insert a command to count pixels into the test program using **Locate > Count Pixels**. The command is an extension of **Locate > Test Brightness Percentage**. The command tests whether the number of pixels between two defined gray levels determined lies within a preset range. The number of pixels determined is stored in the result structure.

You use the command for monitoring and for regulating the brightness of a lighting installation and for surface testing.

Brief Guide

1. Define the following parameters in the **Teach-in** tab:
   - Shape of the test window
   - Position tracking
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the search window in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Specify the following in the **Parameters** tab:
   - Gray value range
   - Result
   - Nominal number and tolerances (of pixels in the gray value range)
6. Test the command with the [Test] button and modify the parameters set in points 1 to 5 until the test result is error free.
7. Insert the command into the test program with the [OK] button.

Teaching-in the Test Window

Position and Appearance of the Test Window

After double clicking with the right mouse button on the **Teach-in** tab, you can position and adapt the test window on the control monitor or on the monitor window.
>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The test window can either always remain at the same image position or follow a reference object.
>> *"Position Tracking of Objects", page 35*

Input Parameters

Parameters	Description
Gray Value Range	The gray scale value range is specified with two gray scale values, and all pixels that lay between these two values are counted.

Evaluation Parameters

Parameters	Description
Result	Result number for saving the pixel count determined in the gray value range.
Nominal Quantity and tolerances	Nominal pixel quantity and the permitted ± tolerances (in pixels).

Test

The detected **quantity of pixels** is displayed in the test window during the test. In the process, the pixels that are within the gray scale value range are superimposed into the overlay.

6.2.25 Locate Point

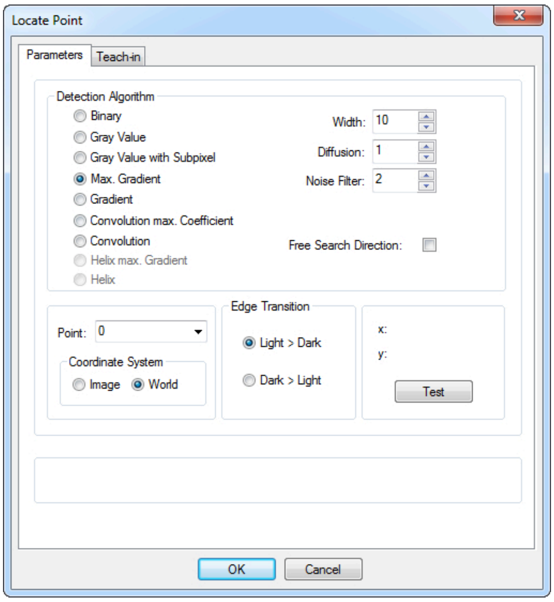


Image 121: Dialogue Locate Point, tab control Parameters

You can find transitional points in the video image using **Locate > Locate Point**. The command applies to a gray value edge cut by the detection beam. The detection beam is defined by its start and end positions. The direction of detection is shown by an arrowhead. A point found along the detection beam is stored as a geometrical point element. The number of points to be saved depends upon the vision system.

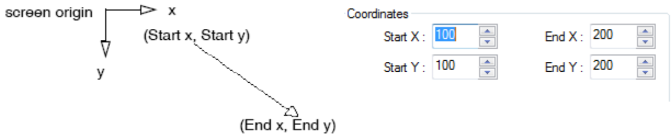
Brief Guide

1. Define the position tracking in the **Teach-in** tab: no, X point, Y point or phi line. If required, activate the **Endpoint Only** option.
2. Activate teach-in mode by double-clicking with the right-hand mouse button.
3. Modify the size and position of the detection beam in the video image using the arrow controls.
4. Deactivate teach-in mode by double-clicking with the right-hand mouse button.
5. Specify the following in the **Parameters** tab:
   - Detection algorithm
   - Co-ordinate system
   - Point no.
   - Edge transition
6. Test the command with the [Test] button:
   Modify the parameters set in points 1-5 until the test result is error free.
7. Insert the command into the test program with the [OK] button.

Teaching-in the Detection Beam

After calling **Locate > Locate Point**, you will see the dialog box in the **Teach-in** tab and a detection beam on the video screen.

Variant 1: Entering Parameters in the Edit Mask



Input fields for defining the detection beam

Enter the X and Y values for the start and end points of the beam into the respective fields. Change the values as required with the arrows on the right hand side. Watch the shift in position and scale of the beam on the video screen.

Variant 2: Setting Parameters on the Monitor Window / Control Monitor

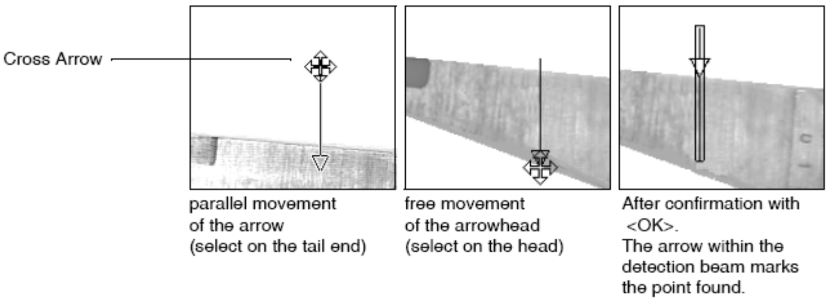
Double click the right-hand mouse button to switch from the **Teach-in** tab to the monitor window or the control monitor. Moving the mouse pointer over the arrowhead or the tail of the detection beam causes a small cross to appear.

To edit the detection beam, hold down the left mouse button while the cross is still visible:

- from the tail … to move beam in parallel
- from the head … to reposition the arrowhead

To move the arrow around the screen, position the mouse cursor on the arrow and hold down the left mouse button while moving the mouse.

Double clicking on the detection beam with the left mouse button will reverse its direction. Double clicking with the right mouse button switches back to the edit window in the **Teach-in** tab.



Positioning and sizing a detection beam

Tracking the Position

The detection beam can either be at a fixed image location (no tracking) or it can follow a reference object.

Activation of the **Endpoint Only** option means that position tracking uses only the front point of the detection beam as a reference point. The base always remains in the same image position.

For position tracking >> *see "Position Tracking of Objects", page 35*

Input Parameters

Detection Algorithm

When detection parameters (thresholds, widths, filters) are modified, the dialog box automatically tests the configuration. However, you can also set adequate parameters interactively with the [Test] button. Use the transitions found to define points.

Detection Algorithm	Input Parameters / Description
Binary	Gray value threshold, noise filter
Gray Value	Contrast, edge diffusion, noise filter
Gray Value with Subpixel	Contrast, edge diffusion, noise filter, width
Max. Gradient	Width of detection, edge diffusion, noise filter
Gradient	Gray value threshold, width, edge diffusion, noise filter
Convolution max. Coefficient	Detection width
Convolution	Gray value threshold, width of detection
Helix max. Gradient	Width of detection, edge diffusion, noise filter
Helix	Gray value threshold, width of detection, edge diffusion, noise filter
Free Search Direc-tion	If **Free Search Direction** is inactive, only detection in a 45° grid is possible, i.e., the detection arrow jumps at an angle of 45° to the position with the best contrast. The speed of the command is hence maximized. Activate **Free Search Direction** in order to to set the search direction for maximum gradient, gradient, convolution max. coefficient, and convolution freely.
Position with best Contrast	Searches within the detected edge for the local maximum.
>> "Principles of Detection", page 417	

Edge Transition

The type of transition can be either light-dark or dark-light.

Edge Transition	Input Parameters / Description
Light -> Dark	Only light-dark transitions are included.
Dark -> Light	Only dark-light transitions are included.

Evaluation Parameters

Evaluation Parameters	Description
Point	Number or name under which the coordinates of the found point is stored in the point structure.
Image/World	Saving in image coordinates or world coordinates.

Test

The test displays:

- X … X coordinate of the point found.
- Y … Y coordinate of the point found.

6.2.26 Measure Temperature

Overview

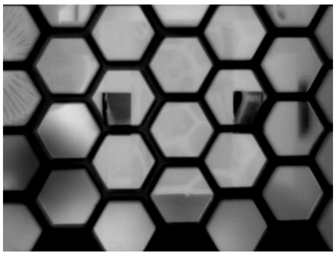


Image 122: Monitor Window: thermal image

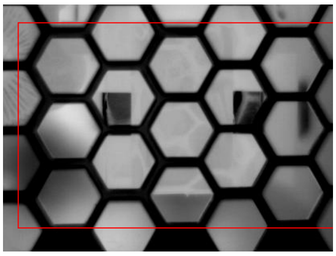


Image 123: Monitor Window: Detection Window at position

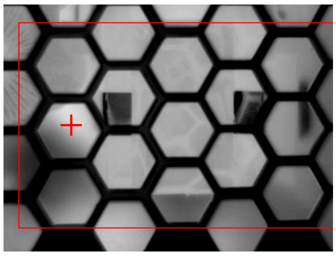


Image 124: Monitor Window: Display Point max. Temp.

Characteristics

- Determining the temperature in a thermal image.
- Determining of minimum, maximum and average temperature.
- Determining of the extreme points of minimum or maximum temperature.

Definitions

Emissivity

- The emissivity is the heat emission emanating from the measurement object, compared to that of a black body.
- The value is between 0 and 1.

Transmissivity

- The transmissivity is the characteristic of media, which are located between the camera and measurement object, to allow heat radiation to pass. For example, these media can be air, cover glasses or protective housing.
- The value is between 0 and 1.

Reflected Temperature

- The reflected temperature indicates the heat emission in the vicinity of the measurement object reflected by the measurement object.

Function

- The command calculates the temperature of an object surface with the specified parameters.
- The camera sensor receives the thermal radiation from the environment and converts it into the total temperature T_{total}. Note that the sensor value is not equal to the visible 8-bit image, but the respective full image value of the camera used.
- Then the command calculates from the transmitted camera values and the input parameters the object temperature T_{obj}.

$$T_{obj} = \frac{1}{\varepsilon * \tau} * T_{total} - \frac{1-\varepsilon}{\varepsilon} * T_{refl} - \frac{1-\tau}{\varepsilon * \tau} * T_{amb}$$

T_{obj}	Object temperature	T_{amb}	Ambient temperature
T_{total}	Total temperature	ε	Emissivity
T_{refl}	Reflected temperature	τ	Transmissivity

Work Flow: Teach-in

1. Switch to the tab control "**Teach-in**".
2. Determine here, by means of a geometry, in which image region the temperature should be determined (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also: *35*).

Work Flow: Parameterizing

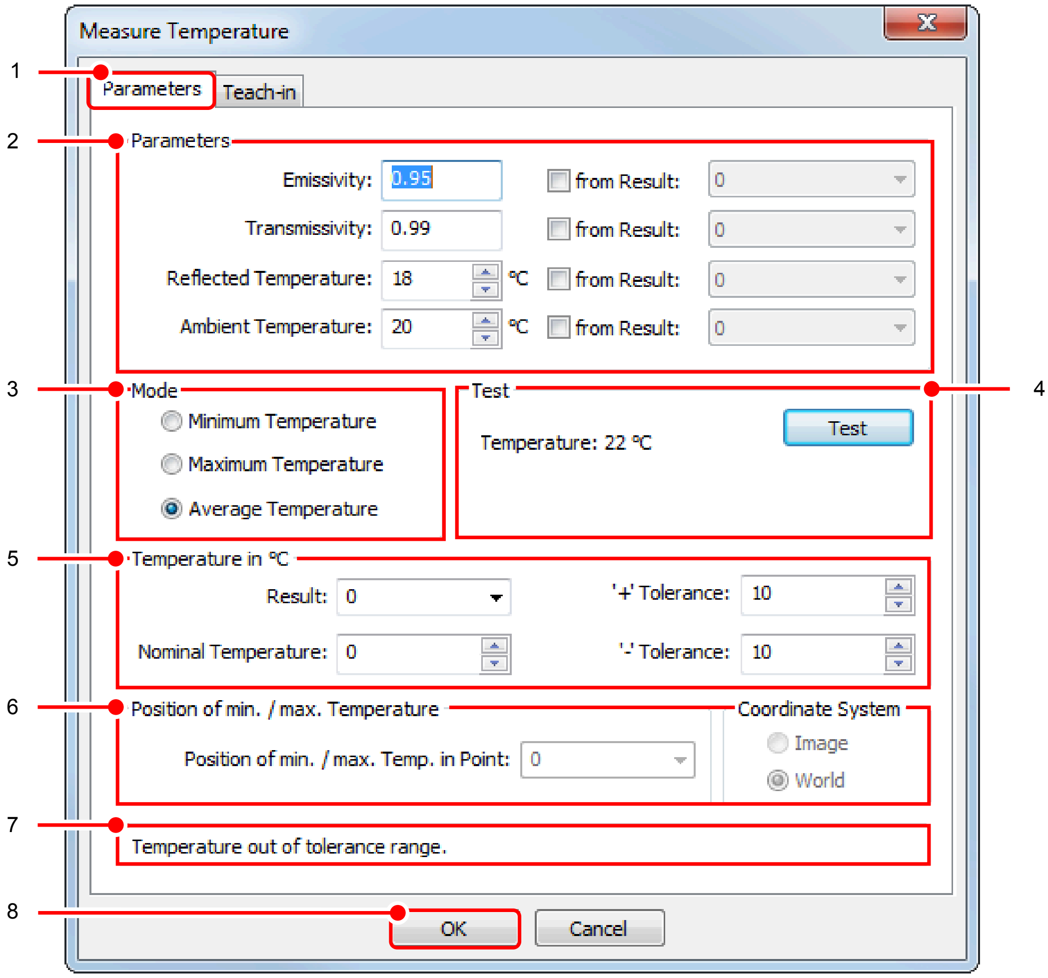


Image 125: Dialog Measure Temperature, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. Enter in the group box "**Parameters**" the ambient and material parameters for the measurement.

Parameter	Description
Emissivity	Enter here the emissivity of the material to be measured. If the value is taken from a result, activate *from Result* and select the appropriate result from the drop-down list. The emissivity is between 0 and 1. If you take this value from the result multiply it by 1000. 1 = 1000; 0.5 = 500 etc.
Transmissivity	Enter here the transmissivity of the material between the measuring object and the camera. If the value is taken from a result, activate *from Result* and select the appropriate result from the drop-down list. The transmissivity is between 0 and 1. If you take this value from the result multiply it by 1000. 1 = 1000; 0.5 = 500 etc.
Reflected Temperature	Enter here the reflected temperature. If the value is taken from a result, activate *from Result* and select the appropriate result from the drop-down list.
Ambient Temperature	Enter here the ambient temperature. If the value is taken from a result, activate *from Result* and select the appropriate result from the drop-down list.

3. Set in the group box "**Mode**" the kind of temperature to be determined.

Parameter	Description
Minimum Temperature	The command determines the minimum temperature and the position in the selected geometry area.
Maximum Temperature	The command determines the maximum temperature and the position in the selected geometry area.
Average Temperature	The command determines the average temperature over all pixels in the selected geometry area.

Work Flow: Testing and Saving

4. Test the settings here

   By activating the [Test] button the command is executed with the set parameters. In the group box "**Test**" the determined temperature is shown. In parallel, the location of the minimum or maximum temperature is marked in the video image with a red cross.

5. Configure under "**Temperature in °C**" the settings for the evaluation of the command and for storing the result.

Parameter	Description
Result	Enter here the number or the name of the result under which the measured temperature is to be stored in the result list.
Nominal Temperature	Enter here which temperature must be measured so that the command is considered successful.
'+' Tolerance	Permitted upward deviations from the Nominal Temperature.
'-' Tolerance	Permitted downward deviations from the Nominal Temperature.

6. Set under "**Position of min. / max. Temperature**" and "**Coordinate System**" the saving option of the temperature extreme point.

Parameter	Description
Position of min. / max. Temp. in Point	Enter the number or name in the point structure under which the coordinates of the temperature extreme point is to be saved.
Image / World	Indication whether the point is to be stored in image or in world coordinates.

7. Evaluation Area

   Here you can see the good/bad evaluation of the command, notes for parameterization (tolerance range exceedance) as well as error messages.

8. Use the [OK] button to insert the command into the test program.

Status- and Error messages

0 = No Errors

- The command executed successfully with the specified parameters.
- The measured temperature is within the specified target range (incl. tolerances).

1 = Temperature out of tolerance range.

- The command determines a temperature that is not within the specified target range (incl. tolerance).

2 = Invalid Parameter

- At least one parameter is invalid.
- Check the settings of the command. Check that the used mode of the command is supported by the vision system used.

3 = Tracking variable not defined.

- The specified point for the X- or Y- position tracking is not defined.
- The specified phi Line for the position tracking is not defined.

4 = Not a thermographic image.

- The command was not executed on a thermal image.

5 = Not a thermographic camera.

- The command was not executed on a thermographic camera.

7 = No valid thermography license available.

- The command is not supported by the used vision system.

6.2.27 Locate Helix

Overview

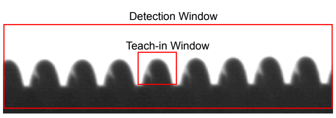


Image 126: Position of the Teach-in and Detection Windows

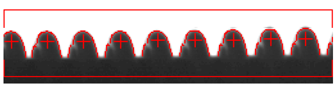


Image 127: Edge image in Monitor Window

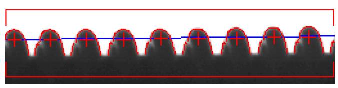


Image 128: Display of reference line in Monitor Window

Characteristics

- The calculation is performed in sub-pixel accuracy.
- Rotations of the coil cusp from up to 10° will be tolerated.
- Edge points with large aberration (e.g. interfering edges as highlights) are ignored.
- The accuracy is achieved even if the helix is in tilt (rotation of the edge points of the taught pattern based on the total position of the helix).

Function

1. Searching of coil cusps by comparing with a thought pattern of a single coil cusp.
2. Determining the position of the cusps.
3. Determining of edge transitions (horizontal and vertical) at the coil cusps found.
4. Comparing of the determined edge transitions with the existing edge transition pattern.
5. Determining of a reference line through the found positions of the coil cusps.
6. Checking the amount of coils and storing the results.
7. Determining of other parameters by using the command **Evaluate > Check Point Distances** *see "Check Point Distances", page 313*.

Work Flow: Teach-in Pattern

1. Switch to the tab control "***Teach-in Pattern***".
2. By means of the geometry rectangle, determine a single coil cusp that has to be read-in as the pattern. (Information for teaching, see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. With the button [Read Pattern] you read-in all image information located in the geometry as the pattern, and stored it internally in the command.
4. In the reference window, verify if the pattern is complete.

Work Flow: Define Teach-in Area

1. Switch to the tab control "***Teach-in***".
2. Please set the geometry, size and position of your detection window. In this area, your pattern is searched (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point) for position tracking (see also: *35*).
4. With the button [Test] you can check whether cusps corresponding to the thought pattern are found by using default settings.

Work Flow: Parameterizing

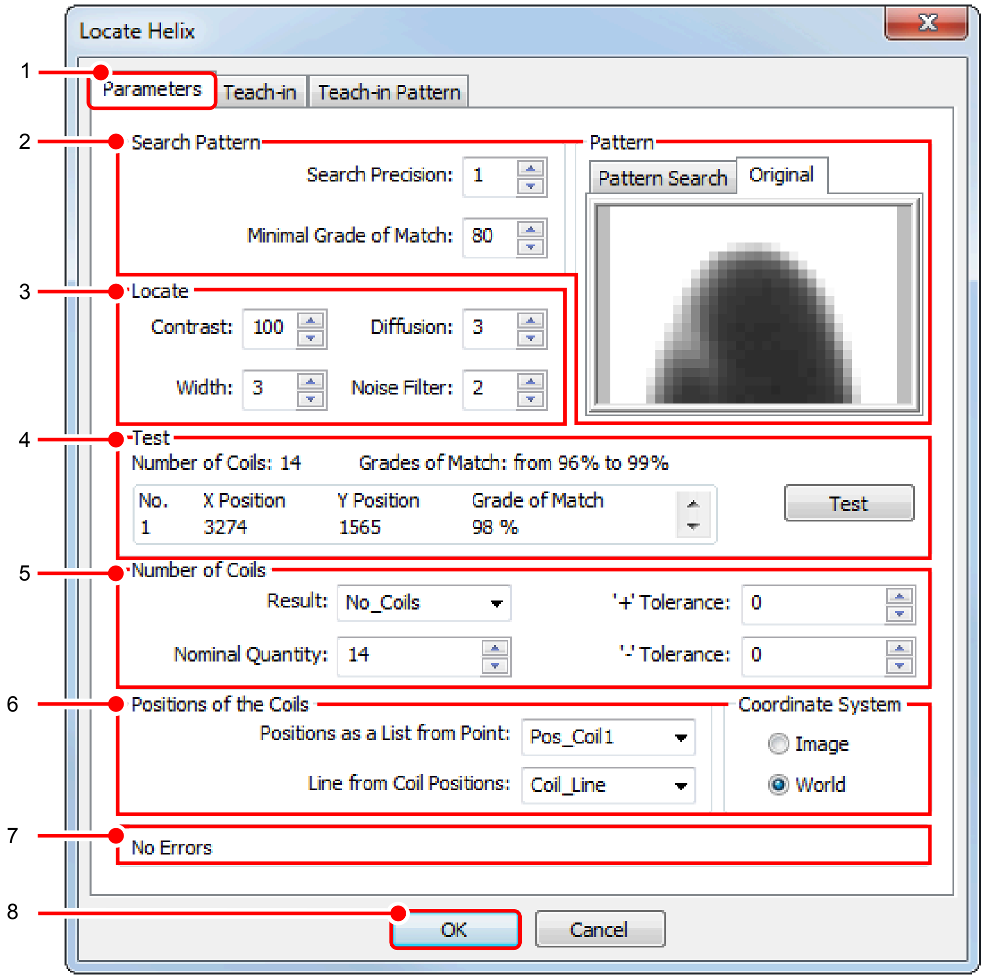


Image 129: Dialogue Locate Helix, tab control Parameters

1. Switch to the tab control "***Parameters***".
2. In the area "**Search Pattern**" determine the search precision method and the minimal grade of match.

Parameter	Description
Search Preci-sion	Factor used for image and pattern for the pattern search. **Factor for Search Precision** • 0 = original resolution • 1 = half resolution • 2 = quarter resolution • etc. The effect of the search accuracy on the taught pattern is shown in the area "**Pattern**" in the tab control "**Pattern Search**".
Minimal Grade of Match	Similarity that must be achieved in an image area with the pattern, so that the pattern is marked as coil cusp. • If further image contents, except the actual coil cusps, are detected as coil cusp, increase the Grade of Match. • If single coil cusps are not detected, decrease the Grade of Match.

3. In the area "**Locate**" set parameters for finding the edge transitions on the edge of a coil cusp.

Parameter	Description
Contrast	Criterion for edge detection. Specifies the minimum grey value difference, from which a transition is considered as an edge. The value should be slightly smaller than the grey value difference between background and helix.
Diffusion	Indicates the number of pixels in which the grey value has to change by a certain amount (contrast) so that the transition is counted as an edge. The value should correspond to the number of transition pixels at the edge between the background and helix. For an optimal determination of a subpixel edge, enter a diffusion of at least 3.
Width	Number of pixels that is averaged transversely over the edges. Individual pixels and noise can be compensated.
Noise Filter	Number of pixels that are averaged over the front and rear of the edge. Using the noise filter pseudo edges and noise are compensated.

Work Flow: Testing and Saving

4. Test the settings here

   After activating the button [Test], all coils found with the search parameters are list in the area "**Test**". In parallel, the coils found are highlighted in the video image with a red cross. Use the arrow keys to scroll through the respective matching coils.

Parameter	Description
Number of Coils	Indication of coils found.
Grades of Match	Indication of the minimal and maximal grade of match.

5. In the area "**Number of Coils**" enter the requirements for the assessment of the command and to store the results.

Parameter	Description
Result	Enter the number or the name of the result under which the quantity of coils found is to be stored in the result structure.
Nominal Quantity	Specify here how many coils must be found so that the command is considered successful.
'+' Tolerance	Permitted upward deviations from this target value.
'-' Tolerance	Permitted downward deviations from this target value.

6. In the area "**Positions of the Coils**" and "**Coordinate System**" enter the saving options of the coils centres and the reference line from coil points.

Parameter	Description
Positions as a List from Point	Enter the number or name in the point structure, from which the positions (points) of the coils found are to be stored. This can be used as an input for the command Check Point Distances.
Line from Coil Positions	Enter the number or name in the line structure, under which the line determined from the coil points is to be stored. This can be used as an input for the command Check Point Distances.
Image / World	Indication whether the points are to be stored in image or in world coordinates.

7. Evaluation Area

   Here you can see the good/bad evaluation of the command, notes for parameterization (tolerance range exceedance) as well as error messages.

8. Use the [OK] button to insert the command into the test program.

Error messages

Possible error messages

Number of coils outside the tolerance range

- The command has found more or less coils than the Nominal Quantity (incl. tolerances).

Tracking variable not defined

- The specified point of the X or Y tracking is not defined.

Command does not work on color image

- The command was executed on a color image. Convert the image into a grayscale image *see "Color Conversion", page 153*.

Pattern not yet taught

- Before executing the command, the pattern of a single coil cusp must be thought in the "***Teach-in Pattern***" area.

Not enough edges in the pattern

- At a coil cusp no sufficient number of edge points could be determined. This can have two causes:
  - The parameters for the edge detection are unsuitable (for example to high contrast).
  - A coil cusp was found at a point where no coil cusp exists. In this case, the Minimal Grade of Match should be increased.

6.2.28 Locate Angle

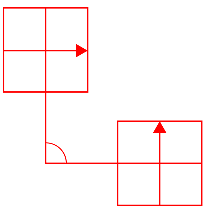


Image 130: Locate Angle - Detection Windows

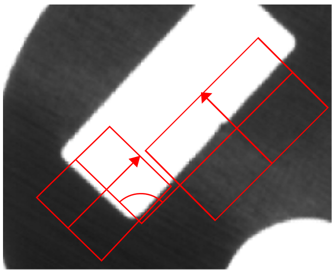


Image 131: Locate Angle - Positioning of the Detection Windows

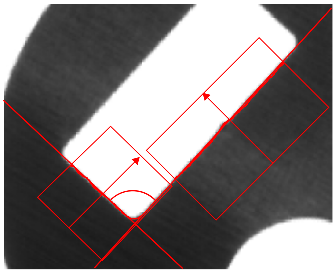


Image 132: Locate Angle - Display: lines found in the Detection Windows

With **Locate > Locate Angle** you determine the angle between two object edges. You select the edges via two detection windows, in which the individual edge points are detected. A line, which represents the respective object edge, is created from each of the edge points.

Workflow Teach-in

1. Switch to the tab control "***Teach-in***".
2. If necessary, set the parameters (X Point, Y Point, phi Line) for position tracking (see also *"Position Tracking of Objects", page 35*).
3. Set here the position of the respective detection windows (see also *"Teaching in Test, Detection and Search windows", page 26*).

ADVICE

Optimal positioning of the windows

- Position the windows so that the detection arrow is perpendicular to the edge to be probed.
- Position the windows so that the edge is completely covered.
- The values for the rotation are selectable in the range of -180.00° to +179.99°, and are in 1/100°. The result for the value 9000 is 90°, 17999 is 179.99°.

4. Use the [Test] button to check whether the edges are found in the detection window with the default parameters.

Workflow
Define Parameters

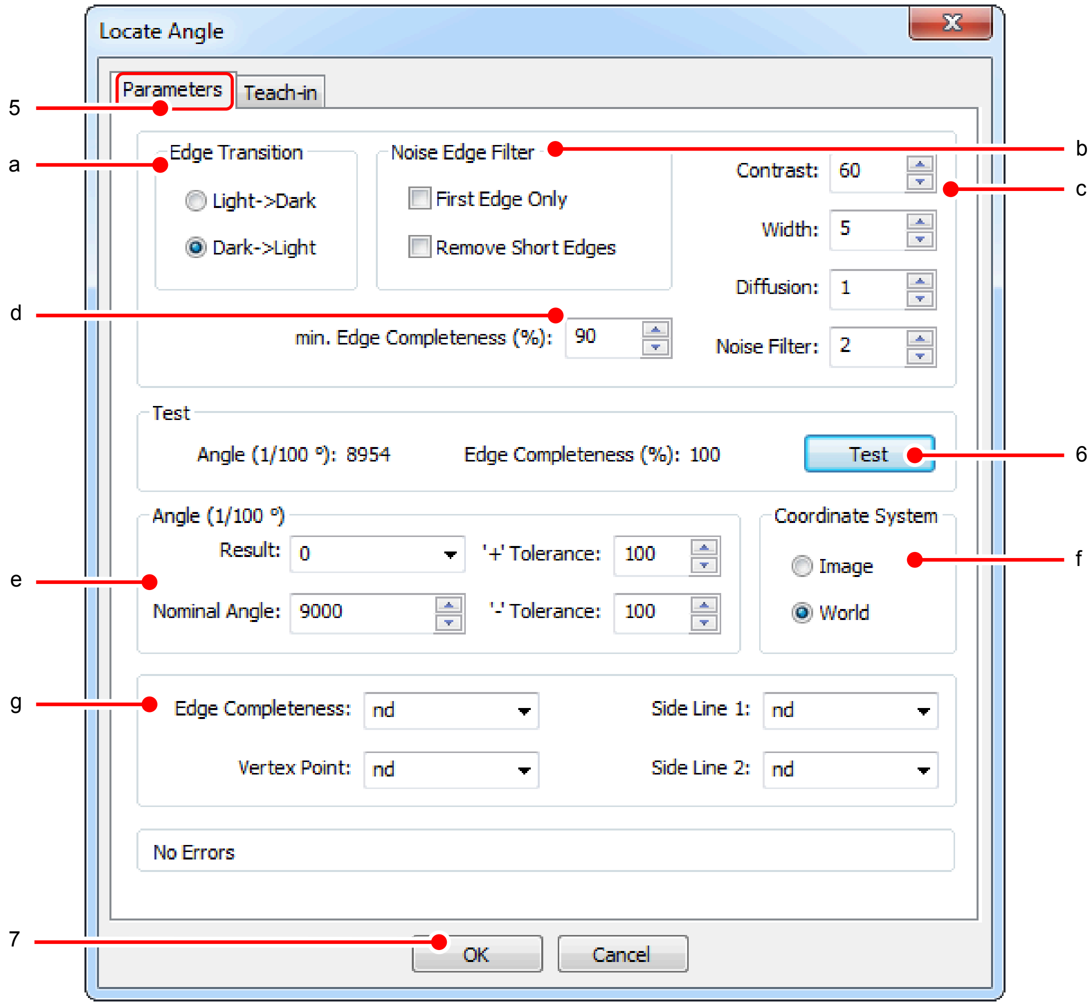


Image 133: Dialogue Locate Angle, tab control Parameters

5. Switch to the tab control "***Parameters***".

   Set here the parameters:

   a) Edge Transition
   b) If necessary: Noise Edge Filter
   c) Parameters for edge detection
   d) Minimum edge completeness in per cent

6. Test the command with the [Test] button:

   ◦ In the area "Test" the determined data are shown.
   ◦ Modify the parameters in the points (a) to (d) until the result is error free.

   Set further parameters:

   e) Result number/name under which the angle is to be stored, as well as the nominal value and tolerances for the evaluation. The values are given in 1/100°. A value of 27000 corresponds to an angle of 270°.
   f) Coordinate System
   g) Save settings of the values determined for Edge Completeness, Vertex Point and Side Line.

7. Use the [OK] button to insert the command into the test program.

Parameters

Edge Transition

Here you can specify whether the edge transition is from light to dark or from dark to light.

Parameter	Description
Light->Dark	Only edge transitions from light to dark are considered.
Dark->Light	Only edge transitions from dark to light are considered.

The detection direction is an important factor for these parameters. You can find any object edge with both possible settings (light->dark, dark->light) if you change the detection direction accordingly. Note, however, that the edge intersections cannot be individually adjusted for each detection area.

Parameters

Noise Edge Filters

These parameters are used to improve the creation of the line from the edges found. If there are numerous interfering edges in the detection area, this filter reduces the processing time significantly.

Parameter	Description
First Edge Only	If you activate this checkbox the algorithm searches through every line of the detection area until it finds the first object edge that corresponds with the set parameters. If you deactivate the checkbox, all edge points that apply for the existing parameters are detected and processed.
Remove Short Edges	If you activate this checkbox, individual edge points or smaller edges are automatically ignored and filtered out. This interference filter is active by default.

Parameters

Edge Filter

These parameters are used to improve the edge detection. When using the arrow keys a test is performed and the result displayed immediately.

Parameter	Description
Contrast	You enter the differential amount here, which must be at least in the image with a grey value transition so that an edge point can be found.
Width	Number of pixels over which the sub-pixel algorithm is applied.
Diffusion	Length of the edge increase.
Noise Filter	The noise filter is the number of pixels in a row that meet the require-ments so that the edge is considered as an edge. Example: a noise filter of 2 hides all edges that meet the conditions, but are shorter than 2 pixels.

Parameters

Minimal Edge Completeness

You can consequently specify how closed and therefore clear an edge must be in the image to be detected as an applicable object edge. Pixel gaps, which can be caused by possible image interferences between individual edge points, can also be compensated and replaced with a certain tolerance. The edge completeness always refers to the complete detection window. A measurement is also performed if the detection window is partially outside the image. In this case however, the edge completeness decreases according to how much of the window is outside the image.

Parameter	Description
min. Edge Completeness (%)	Here you enter the minimum percentage of image lines, that must have at least one edge point, so that an object edge is detected.

Parameters

Angle

Parameter	Description
Result	Result number/name in which the angle is to be stored.
Nominal Angle	Nominal value of the determined angle so that the command is considered to be successful. The values are given in 1/100°. A value of 27000 corresponds to an angle of 270°.
+/- Tolerance	Allowed upper and lower deviation from the nominal angle. The values are given in 1/100°. The values of +100 and -100 correspond to a tolerance of +1° and -1°.

Parameters

Coordinate System

Parameter	Description
Image	Storing of the points and lines is in image coordinates.
World	Storing of the points and lines is in world coordinates.

Parameters

Save Settings

Parameter	Description
Edge Completeness	Result number/name in which the determined edge completeness should be saved.
Vertex Point	Point number/name in which the determined vertex should be saved.
Side Line 1	Line number/name in which the determined side line 1 should be saved.
Side Line 2	Line number/name in which the determined side line 2 should be saved.

Results

Determined Angle and Edge Completeness

After activating the [Test] button, the determined angle and the edge completeness are shown in the area "Test". The determined angle is given in 1/100° and the completeness in per cent.

In parallel, the determined lines and the angle are shown in the video image.

Geometry Variables

In the docking windows geometry lists the following values are given:

Docking Window	Description
Results	Determined angle and good/bad rating according to nominal angle and tolerance. Edges completeness (if declared) and good/bad rating according to min. edges completeness.
Points	X and Y value of the vertex (if declared) in the chosen coordinate system type.
Lines	Side Line 1 (if declared) in the chosen coordinate system type. Side Line 2 (if declared) in the chosen coordinate system type.

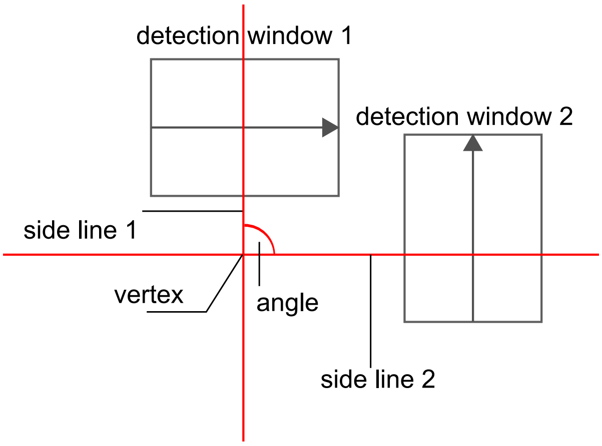


Image 134: Locate Angle, display of the result

6.2.29 Read Character

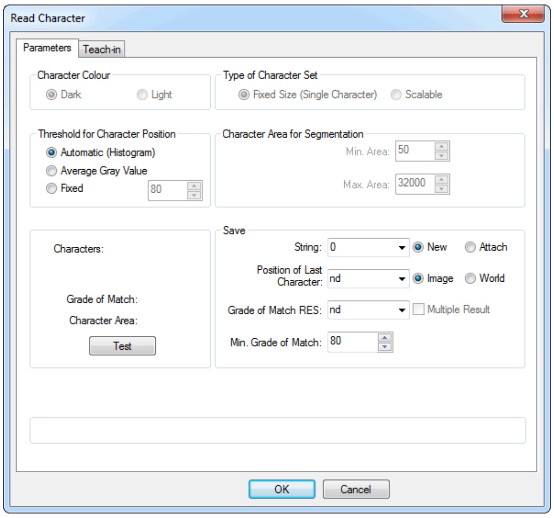


Image 135: Dialog Read Character, tab control Parameters

Use **Locate > Read Character** to teach-in a test command for reading clear text characters on the basis of a previously taught character set. When executing the command, position the search window around the character or characters to be read.

Within the same program, you can read several character sets. All of them are saved in the program. The last character set read is set as the current character set for all subsequent read functions in the test program. When reading, the characters are recognized in the same way as during the teach-in, and compared with the current character set. Characters will be recognized if the correspondence is greater than or equal to the specified minimum value.

Prerequisites for the Teach-in

- Communication with the vision system must be initialized.
- Character set must have been taught. >> *"Set Characters Set", page 269*
- Character type must correspond to the character type taught-in (fixed size or scalable character set).

The parameters of the taught character set appear in the edit dialog box.

For the evaluation, the character read (as string) or the result of the correspondence, together with the taught-in characters are stored or sent.

More Information:

- *"Send Measuring Values", page 394*
- *"Superimpose", page 279*

You can store the character detected in a string or appended it to an existing string with **Evaluation > Evaluate String** command. You can store the gravity center of the first character in a point in order to read several strings at equal spacing. This point is then used as a reference point for reading the next character.

Brief Guide

The programming of **Read Character** takes place in the following order:

1. Define character type: fixed size (single characters) or scalable character set.
2. Define threshold, character color and segmentation (in the case of a scalable character set).
3. Define the detection window.
4. Test and/or teach character recognition in the detection window.
5. Set the evaluation parameters.

Teaching-in the Detection Window

Position and Appearance of the Search Window
After double-clicking with the right-hand mouse button, a teach-in dialog box and a rectangular detection window will appear on the video screen. The search window is characterized as follows for both character types:

Fixed Size (single characters)	Scalable Character Set
• Each character requires its own command in the program. • The search window encloses only one single character. • The search window is the same size for all characters within an edited program. The size specified for the first character applies to all subsequent characters.	• The search window encloses all characters (even several lines). • The individual characters are detected automatically by the segmentation of the search window.

>> *"Teaching in Test, Detection and Search windows", page 26*

Tracking the Position

The detection window can either be at the same image location (no tracking) or it can follow a reference object. >> *"Position Tracking of Objects", page 35*

With the aid of position tracking, arrange several read commands for individual characters advantageously in a loop. See **Strategy for Defining a Loop for Reading Single Characters** later on in this section.

Input Parameters

The character type must correspond to the taught-in character type. For information regarding the advantages and disadvantages of individual character types with **Set Character Set**: >> *"Table 0", page 270*

Threshold for Character Position	Description
You can use this threshold for defining the character position within the search window. The gravity center point of the character is used as the reference for the character position, this is calculated internally.	
Automatic (Histogram)	Automatic threshold determination with the gray scale value histogram
Average Gray Value	Determination of the threshold with the median gray scale value in the window.
Fixed	Manual determination of a fixed threshold in gray values. Initial values between 80 and 120 are advantageous. • You must change this if the reading with the [Test] button is unsuccessful. • The reading quality is affected by the brightness in the test window.

Character Color	Description
Dark	The character to be read is a dark color on light background
Light	The character to be read is a light color on dark background.

Character Area for Segmentation	Description
All characters in the video image are automatically segmented (separated) for the character type **scalable character set**. In so doing, the center of gravity of each segment is determined. The permissible size range of each character has to be defined.	
Min. Area	Minimum area of a single character.
Max. Area	Maximum area of a single character.

Evaluation Parameters

Save	Description
String	String number for storing the character.
New/Attach	Option for defining whether the string is newly written or a character is to be appended to the string.
Image/World	Saving in image coordinates or world coordinates.
Position of Last Character	Point number for storing the gravity center point of the last character read (use this point for the character type “single character [fixed size]” as reference point for the position tracking before reading the next character.)
Grade of Match RES	Result number for storing the degree of correspondence (consistency rate).
Min. Grade of Match	Required gray value correspondence (correlation) in per cent of the character found with the assigned taught-in character for the test procedure.

Test

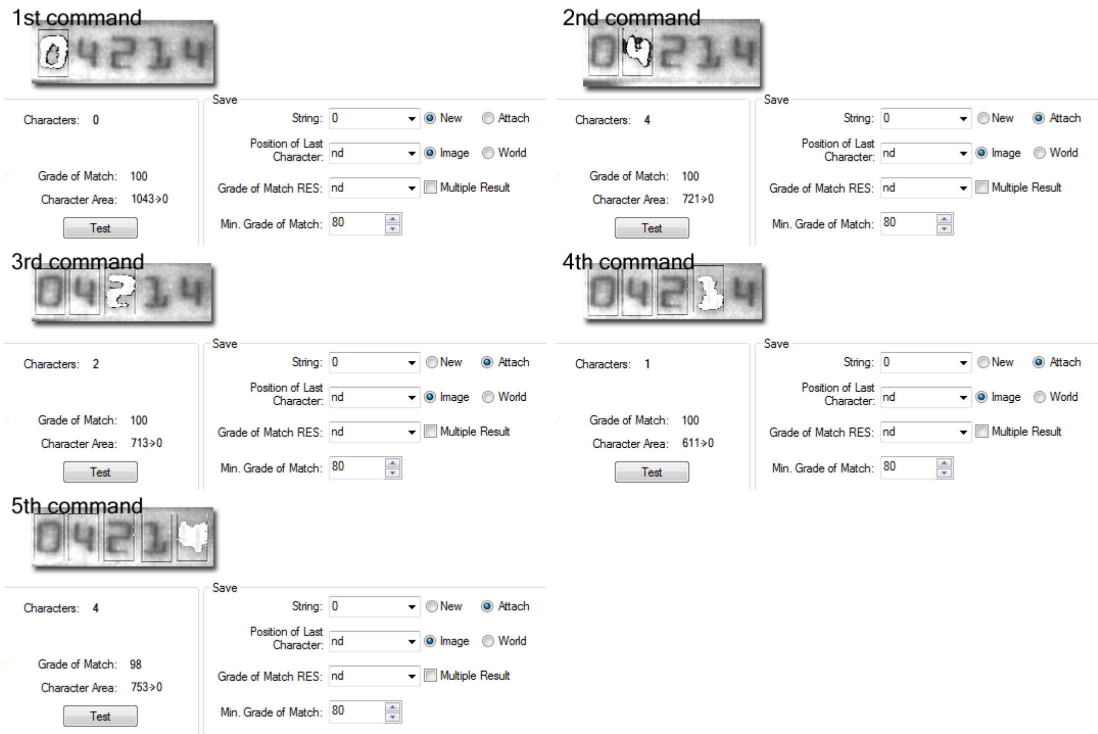
The test displays the following:

Parameters	Description
Characters	Quantity of characters determined (for the character type fixed size: 0 or 1).
Grade of Match	Gray value correspondence (correlation) in per cent of the pattern found with the taught-in pattern.
Character Area	Size of the character area in pixels × pixels.

Recognizing Characters with Fixed Sizes

```
00001 Recognize character in window [101,203/48,82]
      with fixed size, black; save in string no. 2;
      min. Grade of Match 75
00002 Recognize character in window [257,203/48,82]
      with fixed size, black; save in string no. 2;
      min. Grade of Match 80
00003 Recognize character in window [315,202/48,82]
      with fixed size, black; save in string no. 2;
      min. Grade of Match 80
00004 Recognize character in window [377,204/48,82]
      with fixed size, black; save in string no. 2;
      min. Grade of Match 80
00005 Recognize character in window [435,202/48,82]
      with fixed size, black; save in string no. 2;
      min. Grade of Match 80
```

Recognize Character • Program script example for 5 characters: fixed size (single characters)



Prerequisite: The character set has been taught-in as **fixed size (single character)**

1. Open the **Read Character** dialog box.
2. Enter the evaluation parameters (string, grade of match, coordinate system).
3. Position the search window around the first character allowing some clearance. (The largest character in the character set should fit into the search window.)
4. Click on the [Test] button.
   The character is read in and displayed in clear text together with the consistency rate (Grade of Match) in comparison to the taught-in nominal character. If a question mark "?" is displayed, you should change the detection parameters or teach this character in again.
5. Click on the [OK] button.
   The Read Character dialog box closes. A program step is added which can read a character in the corresponding position in run mode.
6. Repeat steps 1 to 6 for each further character: See example picture **Recognize Character • Program script example for 5 characters: fixed size (single characters)**.

Strategy for Defining a Loop for Reading Single Characters

ADVICE

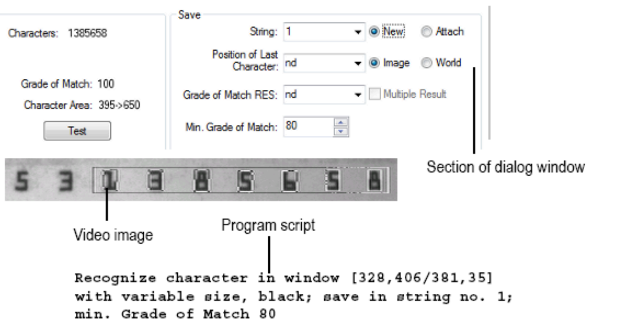
As the gravity center points of the individual characters vary as a result of the differing geometries of the characters, you should select a search window in the video image which is larger than the actual character (especially at the top and bottom).

Prerequisite: The read command for a single character can run in a loop if the spacing between all the characters is equal and they all have similar sizes.

1. Save the gravity center point of the first character as Position of Last Character in a point number, e.g. “2”. See example picture **Recognize Character • Program script example for 5 characters: fixed size (single characters)**.
2. For teaching-in the next character:
   - Select the previously saved point (“2” in the example) for position tracking.
   - Save the determined gravity center point again under the same number (“2” in the example) as Position of Last Character.
3. Nest the command taught-in with pos. 2 in a loop with the aid of program control commands.

Result: The position at which each new character is determined is related to the gravity center point of the previously recognized character in each case. The search windows of successive characters are shifted relative to one another by the same amount.

Recognizing Characters with Scalable Sizes



Recognize Character • Example with 7 characters: variable size (scalable character set)

Prerequisite: The character set was taught as **Scalable character set**.

1. Open the **Read Character** dialog box.
2. Enter the evaluation parameters (string, grade of match, coordinate system).
3. Position the search window around all the characters allowing some clearance.
4. Click on the [Test] button.
   The characters are read in, automatically segmented, and displayed in clear text together with the degree of correspondence in comparison to the taught-in nominal character. If a “?” is displayed, the detection parameters must be changed, or this character must be taught-in again.
5. Click on the [OK] button.
   The **Read Character** dialog box closes. A program step is added which can read all characters in the corresponding positions in run mode.

6.2.30 Set Characters Set

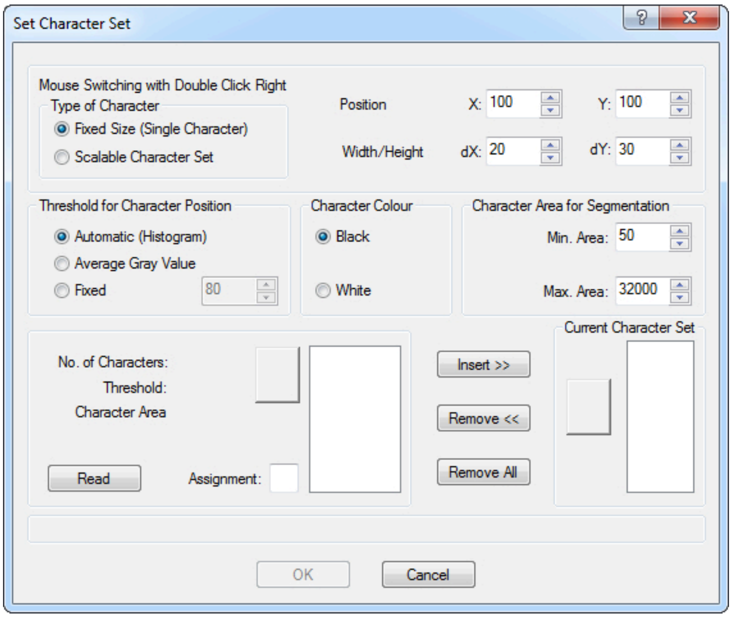


Image 136: Dialogue Set Character Set

Teach-in character sets or individual characters interactively with **Locate > Set Character Set**. When using this command, the detection window is positioned around one or more characters for teach-in.

Use the command for preparing the automated reading of clear text characters, such as those described in the **Read Character** section. >> *"Read Character", page 263* Preferably, embed this command in a teaching program which runs before the actual test program.

Character Set with Fixed Size

You can determine the gravity center point of all pixels of the character color using the **Fixed Size (Single Character)** option. The data of the character is read in relative to this gravity center point. The taught-in characters are inserted into a character set. In the current character set, the individual characters can be leafed through, deleted or replaced by new characters. In the case of a character set with fixed sizes, the window size cannot be changed after the first character has been inserted.

Scalable Character Set

You can segment all characters present in the window corresponding to color and surface area using the **Scalable Character Set** option. The data of the characters are read one after the other and displayed in the dialog box. An individual character can be selected and inserted into the current character set.

Brief Guide

The programming of **Set Character Set** takes place in the following order:

1. Define character type: fixed size (single character) or scalable character set.
2. Define threshold, character color and segmentation (in the case of a scalable character set).
3. Define the detection window. Read the image of the character(s) in the search window.
4. Assign a nominal character to the image of the character(s) read and transfer it into a character set.
5. Click on the [OK] button to transfer the character set to the vision system.

Teaching-in the Detection Window

Position and Appearance of the Search Window

After double-clicking with the right mouse button on the left corner of the dialog, you can position and adapt the detection window on the video screen. This search window is characterized as follows for both character types:

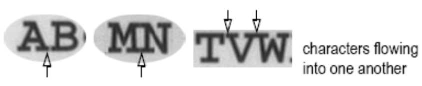
fixed size (single characters)	scalable character set
• The search window encloses only one single character. • The search window is the same size for all characters within an open dialog box. The size specified for the first character applies to all subsequent characters.	• The search window encloses all characters (even several lines). • The individual characters are detected automatically by the segmentation of the search window.

>> *"Teaching in Test, Detection and Search windows", page 26*

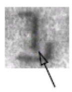
Input Parameters

Character Type	Fixed size (single characters)	scalable character set
You can switch between the character types **fixed size (single characters)** and **scalable character set** until a character has not been taught.		
Advantages	• Characters which flow into one another or are broken can be recognized. • It does not need to be segmented (better recognition). • Individual characters can be inserted into the character set later.	• All characters can be taught in a single program step. • The characters to be recognized (read characters) may be a different size to the taught-in characters.
Disadvantages	• An individual detection is required for each character. • Only one character can be taught with one detection. • The characters to be recognized (read characters) must be the same size as the taught-in characters.	• Characters which flow into one another or are broken cannot be recognized. Good character quality is required. • See later below: **Problematic characters with scalable character set**.

Problematic characters with scalable character set



Difficult characters in scalable character sets


Broken characters

Threshold for Character Position	Description
You can define a binary threshold for determining the gravity center point of the character (fixed size/single character) or for segmenting the character (scalable character set). The gravity center point of the character is used as the reference for the character position, this is calculated internally.	
Automatic (Histogram)	Automatic threshold determination with the gray scale value histogram.
Average Gray Value	Determination of the threshold with the median gray scale value in the window.
Fixed	Manual determination of a fixed threshold in gray values. Initial values between 80 and 120 are advantageous. • You must change this if the reading with the [Read] button is unsuccessful. • The reading quality is affected by the brightness in the test window.

Character Color	Description
Black	The character to be taught is a dark script on light background.
White	The character is a light script on dark background.

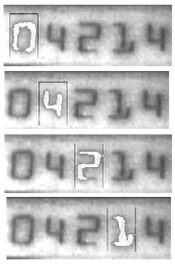
Character Area for Segmentation	Description
All characters in the video image are automatically segmented (separated) for the character type scalable character set. In so doing, the center of gravity of each segment is determined. The permissible size range of each character has to be defined.	
Min. Area	Minimum area of a single character.
Max. Area	Maximum area of a single character.

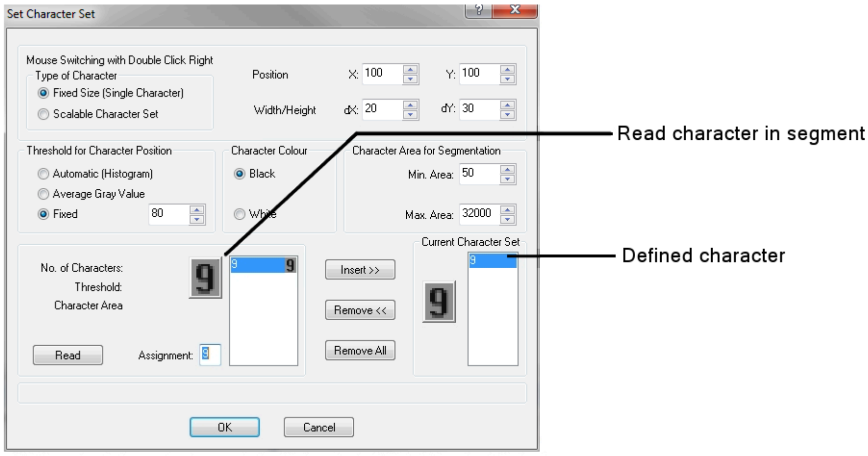
Setting Parameters for Reading and Defining »Characters«

Characters Found	Description
After reading the characters, a segment is displayed for each character detected. Scroll between the segments with the cursor keys.	
No. of Characters	Quantity of characters found.
Threshold	Automatic or manually defined gray value threshold.
Character Area	Size of the character area in pixels × pixels.
Assignment	Nominal character entered which is to be assigned to the segment displayed in the window. You can enter with the keyboard all those characters and symbols which can be displayed in the input field. It is possible to separate them into upper and lower case letters.

Current Character Set	Description
A segment is displayed for each character defined. Scroll between the segments with the cursor keys.	
Characters	Nominal character assigned to the segment displayed in the window.
Number	Number of characters in the character set.

Teaching-in Single Characters (Fixed Size)


Example (above): Teaching four characters with successive detections for the character type Fixed Size (Single Character)



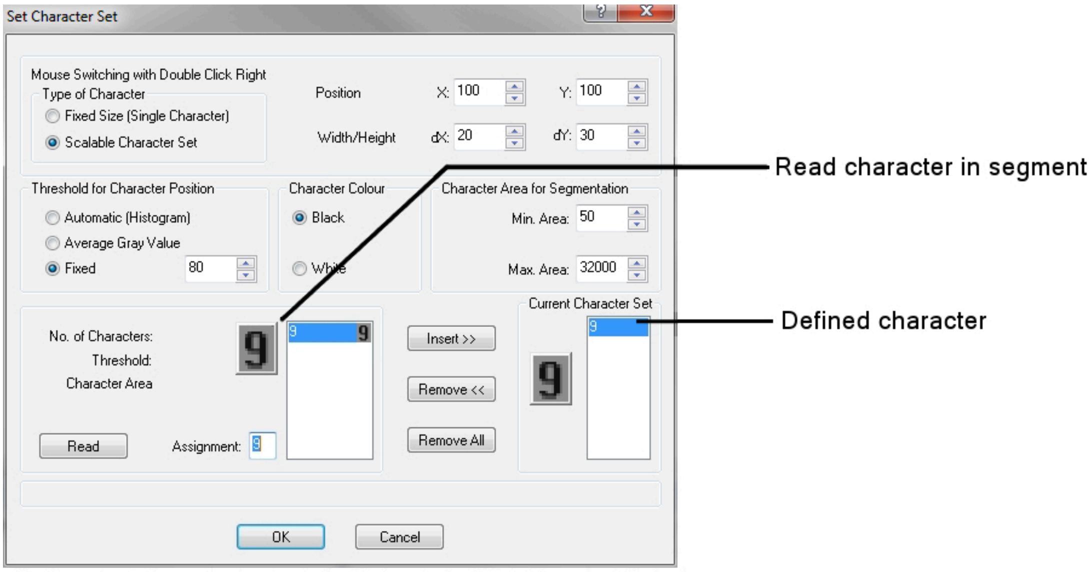
Prerequisite: The **Set Character Set** dialog box is opened and the character type **Fixed Size (Single Character)** has been selected.

1. Position the search window around the first character allowing some clearance. (The largest character in the character set should fit into the search window.)
2. Click on the [Read] button.
   The character is read in. The character read is displayed in the segment.
3. Enter the associated nominal character on the keyboard.
4. Click on the [Insert >>] button.
   The defined character is transferred into the current character set. Repeat steps 1 to 4 for each further character.
5. Click on the [OK] button.
   The characters included in the current character set are sent to the vision system.

Example: Program entry for the taught characters teached by Set Character Set/ fixed size (single character)

```
Font with fixed size [48,82], black
Number of characters 4:0421
```

Teaching-in a Scalable Character Set



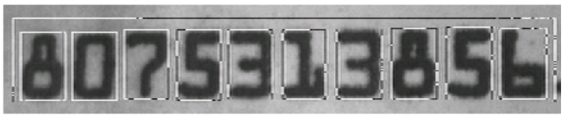


Image 137: Examble of search window: Each character detected is a separate segment.

Prerequisite: The **Set Character Set** dialog box is open and the character type **Scalable Character Size** has been selected.

1. Position the search window around all the characters of the character set.
2. Click on the [Read] button.
   The characters are read in and automatically segmented.
   The segments are displayed in the image section of the dialog box together with the characters which have been read in. Scroll between the characters with the cursor keys.
3. Enter on the keyboard the nominal character associated with the character displayed (segment).
4. Click on the [Insert >>] button.
   The defined character is transferred into the current character set.
5. Repeat steps 3 and 4 for each further character (also see Fig. above).
6. Click on the [OK] button.
   The processed character set is sent to the vision system together with the defined characters.
   The **Set Character Set** dialog box closes. The character set has been taught.

Example: Program entry for the taught characters for search window (relates to example of search window below point 2)

```
Scalable Font, black
Number of characters 8:80753136
```

6.3 Evaluation Commands

6.3.1 Best Fit Line

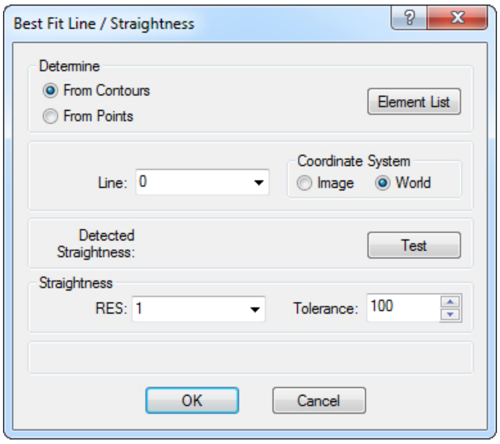


Image 138: Dialogue Best Fit Line / Straightness

You can define a command for determining the best fit line from the contour buffer or points from the point structure using **Evaluation > Best Fit Line**.

You can also test the selected contours and points for straightness. Straightness is defined as a cylinder around the best fit line which contains all the points. So, the straightness is the maximum distance between the calculated straight and any given point.

Select the points and contours which you want to include from the list of geometry variables using the mouse and confirm the selection by clicking on the [OK] button. Defined elements in the list appear dark and undefined ones appear gray.

Input Parameters

Parameters	Description
Coordinate system > Image/ World	Calculation in image coordinates or world coordinates (may only be specified from contours, not from points).
Determine > From Contours/ From Points	Selects whether straightness is defined from contours or from points.

Evaluation Parameters

Evaluation	Description
Line	Number under which the coordinates of the best fit line will be stored in the line structure.

Straightness	Description
RES	Result number used for storing the calculated straightness. If you select {-1} or {nd}, the point will not be stored.
Tolerance	Tolerance allowed during the calculation. It can be set interactively by using the [Test] button.

Test

The test displays the calculated **Actual Straightness**.

6.3.2 Best Fit Circle / Roundness

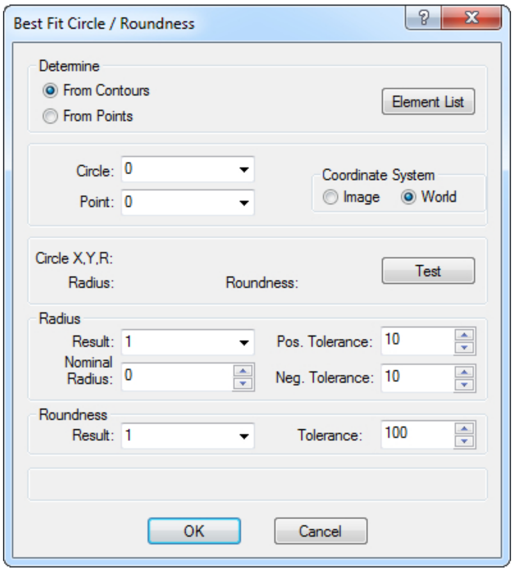


Image 139: Dialogue Best Fit Circle / Roundness

You can define a command for determining the best fit circle from the contour buffer or points from the point structure using **Evaluation > Best Fit Circle**.

You can also test the selected contours and points for roundness. Roundness is defined as a cylinder around the Best Fit Circle. So, the roundness is always defined in world units. The dimensions of this cylinder are defined by the minimum and maximum radii between which all selected contours and points lie. The central point of the Best Fit Circle can be stored in the point structure. The radius of the Best Fit Circle and the corresponding radius can be compared with a predefined radius. The points and contours to be included are selected from a list of elements using the mouse. The selection is confirmed by clicking on the [OK] button. Defined elements in the list appear dark and undefined ones appear gray.

Input Parameters

Parameters	Description
Coordinate system > Image/ World	Calculation in image coordinates or world coordinates (may only be specified from contours, not from points).
Determine > From Contours/ From Points	Selects whether roundness is defined from contours or from points.

Evaluation Parameters

Parameters	Description
Circle	Enter the number in the circle structure under which the best fit circle will be stored.
Point	Enter the number in the point structure under which the center of the best fit circle will be stored.

Radius	Description
Result	Result number under which the calculated radius will be stored: If you select {-1} or {nd}, the data will not be stored.
Nominal Radius and Tolerances	Nominal radius and the allowed ± tolerances in pixels: Set suitable parameters interactively with the aid of the [Test] button.

Roundness	Description
Result	Result number for saving the resulting Roundness. If you select {-1} or {nd}, the point will not be stored.
Tolerance	Acceptable deviations from the nominal value for the roundness. It can be set interactively by using the [Test] button.

Test

The test displays:

Parameters	Description
Circle X,Y,R	Calculated coordinates X, Y, and radius of the circle.
Radius	Calculated radius of the circle.
Roundness	Calculated roundness of the circle.

6.3.3 Distance

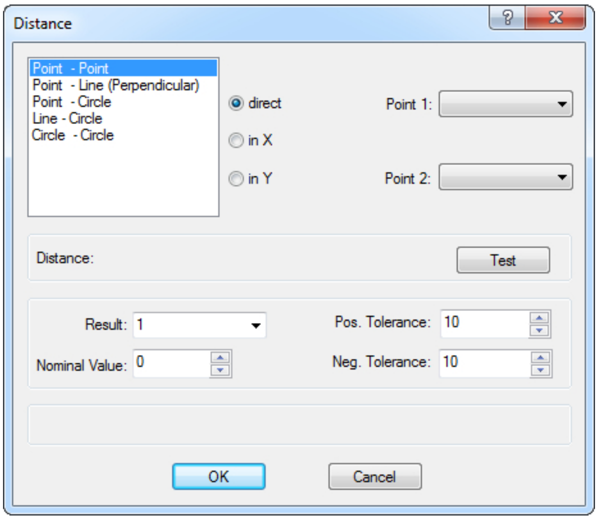


Image 140: Dialogue Distance

Write a command for determining the distance between geometrical variables in the test program using **Evaluation > Distance**. To do this use any previously defined points, lines, planes or circles. Distance describes the shortest euclidean distance between these geometric variables.

Selecting a Method

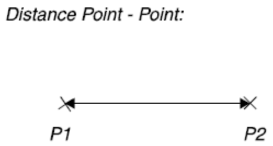
The following methods are possible:

- Distance between two points
- Length of a perpendicular (plumb) between a point and a line
- Distance between a point and the center of a circle. Also the minimum and maximum distances between a point and a circle (i.e. to nearest and farthest points on the circle).

- Minimum and maximum distance between a line and a circle (i.e. to nearest and farthest points on the circle).
- Distance between the centers of two circles

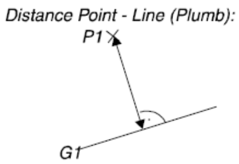
The result (distance) is stored in the result structure. Tolerances can also be checked.

Distance Point - Point



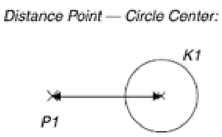
1. Enter the first point number.
2. Enter the second point number.
3. Select the distance determined either directly or only the X or Y distances.
4. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
5. Enter the result number.

Distance Point - Line (Plumb)



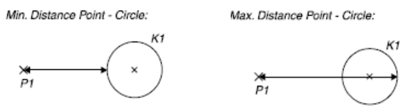
1. Enter the point number.
2. Enter the line number.
3. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
4. Enter the result number.

Distance Point - Circle Center



1. Enter the point number.
2. Enter the circle number.
3. Select the **to Center** option.
4. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
5. Enter the result number.

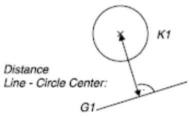
Min. / Max. Distance Point - Circle



1. Enter the point number.
2. Enter the circle number.
3. Activate the **to Circumference min./max.** option.

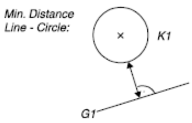
4. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
5. Enter the result number.

Distance Line - Circle Center



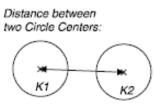
1. Enter the line number.
2. Enter the circle number.
3. Activate the **to Center** option.
4. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
5. Enter the result number.

Min. / Max. Distance Line - Circle



1. Enter the point number.
2. Enter the circle number.
3. Activate the **to Circumference min./max.** option.
4. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
5. Enter the result number.

Distance Between two Circle Centers



1. Enter the first circle number.
2. Enter the second circle number.
3. Select whether the distance should be measured directly or only the X or Y component.
4. Enter the nominal distance and +/– tolerances in world units (default 1/10 Pixel).
5. Enter the result number.

Evaluation Parameters

Evaluation	Description
Result	Result number for storing the calculated distance.
Nominal Value and Tolerances	Nominal distance and the allowed +/– tolerances in pixels. Set suitable parameters interactively with the aid of the [Test] button.

Test

The test displays the calculated **distance**.

6.3.4 Superimpose

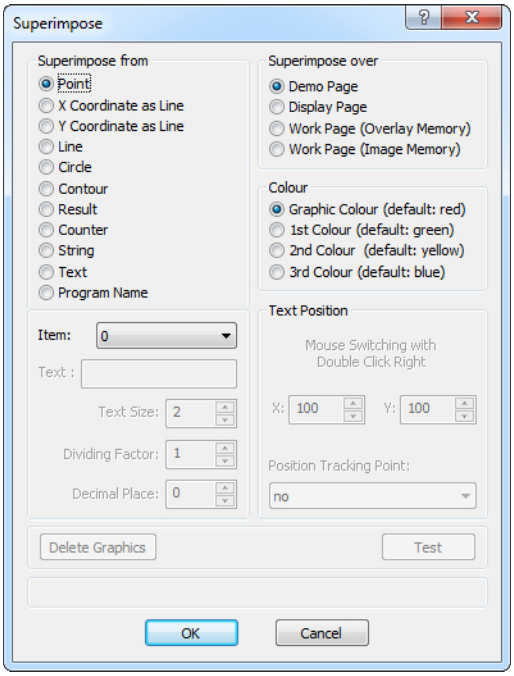


Image 141: Dialogue Superimpose

With **Evaluation > Superimpose** you create a command for superimposing the value of a variable of the geometry structure, auxiliary texts, measured values, text and other information into the video image.

Superimpose Geometry Variables and Information

You can insert a superimposition command into the test program for the following geometry variables:

- Point (Shown as a cross)
- X Coordinate of a point as a Line
- Y Coordinate of a point as a Line
- Line
- Circle
- Contour
- Result with additional text
- Counter with additional text
- String
- Text
- Program Name

Enter the number or the name of the element in the corresponding element structure. Enter the desired additional text for superimposing Result or Counter in the text box.

The program name must not be entered, it will automatically taken over. Thus, the program name is displayed correctly, the program must be transfered to the vision system.

Memory Image for Superimpose

As an image memory for superimpose one of the following pages is to select:

- Display Page: Superimpose on control monitor or in the monitor window
- Demo Page: Superimpose as overlay in demo mode to illustrate the processing
- Work Page (Overlay Memory): Superimpose as overlay in the image, image content is not overwritten
- Work Page (Image Memory): Superimpose directly in the image, image content is overwritten

The assignment of the image memory pages can be made via the command **Image>Display** *see "Display", page 147*.

Superimpose Color

The following options can be selected:

- Graphic Color (red)
- 1st Color (green)
- 2nd Color (yellow)
- 3rd Color (blue)

You can change the colors for the display in the monitor window and on an external monitor by using the command **Control > Direct Code Input** and a vicorem code.

	Default	Change	Examble
Graphic	red (FF 00 00)	02 40 xx xx xx	02 40 FF FF FF (white)
1st color	green (00 FF 00)	02 41 xx xx xx	02 41 00 00 FF (blue)
2nd color	yellow (FF FF 00)	02 42 xx xx xx	02 42 A0 20 F0 (purple)
3rd color	blue (00 00 FF)	02 43 xx xx xx	02 43 00 00 00 (black)

ADVICE

More information about vicorem and its use can be found in the vicorem manual. Please contact our support.

ADVICE

There is no takeover of the change color values after reconnecting to a vision system of the type pictor M. The representation in the operating software is done in the standard colors.

Settings Text Position

These settings are for result, counter, string, text, and program name.

Determining Text Position

Here you define the position of the superimposed information and optionally a reference point in the image.

Enter the x- and y-coordinate of the superimpose position in image coordinates, or double-click with the right-hand mouse button in the field Text Position for positioning the frame in the monitor window /control monitor.

Position Tracking Point

Superimposition can be static or can follow a point.

If you enter a point number, the superimposed information can follow the point movement. Choose {no} for the reference point creates static overlays.

Settings Text

Depending on the types of "**Superimpose from**", you can configure the required parameters. Parameters that have no influence on the respective superimposition will be grayed out.

Input Field Text:

Enter here the text that is to be superimposed, as well as any additional text (at option Result or Counter).

ADVICE

A maximum of 39 characters can be entered.

A line break will not occur. Characters which are beyond the image will be cut off.

Text Size:

Here, the text size can be selected.

Dividing Factor:

Conversion factor for the display. Only integers can be entered, with exception of zero.

Decimal Place:

Sets the number of decimal places. Results may be displayed in Monitor Window or on the control monitor in floating point format, showing the decimal places. A dividing factor and the number of decimal places must be entered in order to achieve this.

Example:

Using the command "angle", an angle of 26.57 degrees was calculated and written to the result. The value of the angle is in 1/100 degrees so the angel has a value of 2657.

superimpose without dividing factor: angle: 2657

superimpose with dividing factor '100': angle: 27

superimpose with dividing factor '100' and decimal place '2': angle:26.57

Delete Graphics

By clicking the button [Delete Graphics] the Overlay-Graphic will be deleted.

Test Command

By clicking the button [Test] overlays in the monitor window or on the video screen are displayed.

6.3.5 Evaluate Result

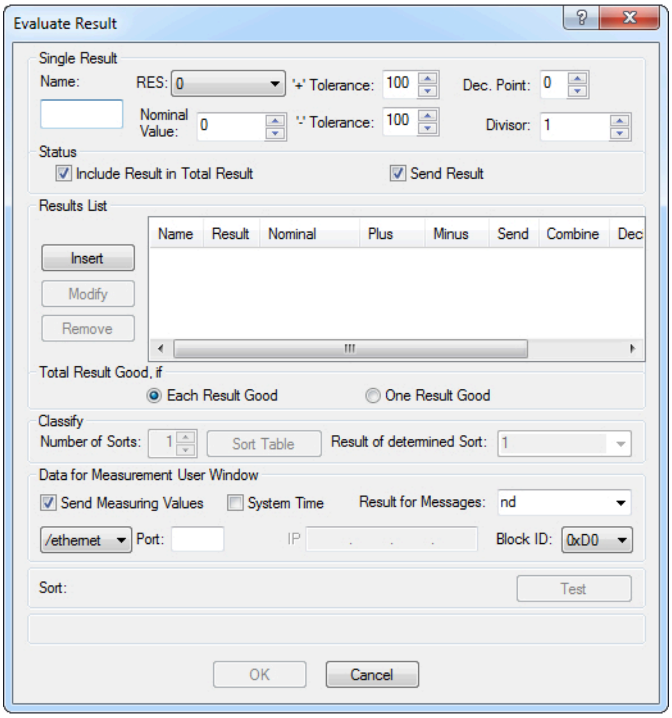


Image 142: Dialogue Evaluate Result

Define a command for the logical operation, classifying and sending of results from the result structure using **Evaluation > Evaluate Result**. Determine the following for each individual result individually:

- Nominal value and tolerances of the result.
- Should the result be sent to vcwin for visualization?
- Should the result be considered when determining the total result?

Independently of this, all the results for determining the sort are taken into account. The classification is made with a so-called sort table. This defines which individual results (good, bad or no influence) are assigned to a sort. Depending upon the individual results, you can assign an overall result (good, bad) to the sorts. Store the sort determined in the result structure as required.

ADVICE

Do not use this command together with the Send Measuring Values command.

ADVICE

If this command is used on the system pictor M, special rules need to be observed. *see "Usage at vision systems of the type pictor M"*

Input Parameters

Single Result	Description
Name	Identifier for checking an individual result.
Nominal Value and Tolerances	Nominal value and the permitted +/– tolerances of the individual result which is to be checked.
RES	Number of the individual result from the result structure which is to be checked.
Dec. Point & Divisor	The settings "Dec. Point" and "Divisor" concern only the measurement user window and will be ignored by pictor / vicosys for sending data.
Dec. Point	Sets the number of decimal places. Example: 2 means that there are two decimal places after the dot.
Divisor	Sets the value by which the result is divided. Example: divisor 2 means that the result is divided by 2.

Status	Description
All the individual results which are to be entered in the result table. Depending on how the status boxes are marked, the result can be sent and/or be added to a total result.	
Include Result in Total Result	Single results are included or not included in determining a total result.
Send Result	Single results are sent or not sent.

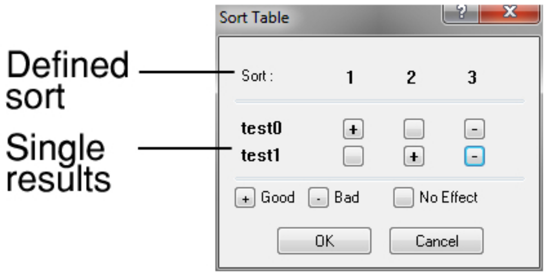
Buttons in the Results List	Description
Insert	Transfers the parameters of a individual result into the result table.
Edit	Changes the parameters of the marked individual result in the result table
Clear	Deletes the marked individual result from the result table.

Evaluation Parameters

Total result Good, when	Description
Each Result Good	Total result is good if all individual results are good.
One Result Good	Total result is good if one individual result is good.

Classify	Description
Quantity	Quantity of possible sorts in the sort table.
Result	Result number for saving the calculated sort. If you select {-1} or {nd}, the data will not be stored.

Sort Table



Use the [Sort Table] button to open the sort table. You can assign the individual results to the sorts in the sort table.

The columns are sorted as follows:

- The sort appears at the top (designated by a number).
- The operators for the individual results are lined up underneath

Assign the following logical operands to an individual result per mouse click:
[+] … Result must be good.
[–] … Result must be bad.
[ ] … Result can be ignored.

The following rules apply for determining the sort:

- If all individual results within the first column correspond to the assigned operands, then sort number 1 is output.
- If the first column does not comply, the columns to the right are searched for compliance and – if there is compliance – the corresponding sort is output.
- If no column complies with the individual results, then the sort “0” is output into the result structure.

Data for measuring User Window	Description
Send Measuring Values	Activates the transfer of measured values marked with **Send Result** and the total result via the data interface to vcwin . These values can be visually displayed in the Measurement User Window in vcwin . This visualization is only possible if vcwin has started with the measurement user window. >> *"User Window", page 62*
System Time	System time of the vision system will be sent or not.
Result for Messages	Result number for saving the system result calculated. If you select {-1} or {nd}, the data will not be stored. How this is set has no influence on the sending of results.
Block ID	Send block label for sending values. Different result send blocks that are labeled with different recognition IDs can be sent over the data interface.

Interface	Description
Select one of the available data interfaces, to which the result is to be sent. The interfaces that are available depend on the type of vision system used.	
/com	Serial interface
/ethernet	Ethernet interface
In order to guarantee proper functioning, do not change the entries behind /com and / ethernet unless you are a network technician.	

Test

The sort determined is displayed during the test.

Usage at vision systems of the type pictor M

ADVICE

At the beginning of the test cycle, set back the **Result for Messages** value from **Data for Measured User Window**! Use the command **Combine Results > Set** = 0.

The result of the command **Evaluate Result** is either good (error equal 0) or bad (error unequal 0). If the current value of **Result for Message** is bad (unequal 0) and the previous value of **Result for Message** was good, then the value for **Result for Message** is set to bad (unequal 0). Even if the value turns to good, the value for **Result for Message** still remains bad until is is reset to 0.

If an error occurs in a test cycle, the **Result for Message** value is bad until it is explicitly set to 0.

ADVICE

As opposed to vicosys and pictor T, the vision system pictor M takes only the evaluation in the **Result for Message**, not whether errors have occurred during transmission.

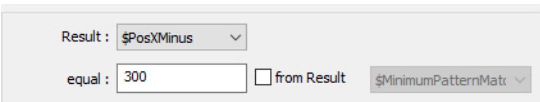
6.3.6 Result equal

With **Evaluation > Result equal** you write a command into the test program, which compares a result with a fixed value or with another result.

The result is evaluated as good if its value is equal to the comparative value.

Setting parameters

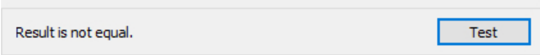
Enter the parameters in the dialog:



Parameter	Description
Result	Select the result to be compared from the drop-down list.
equal	Enter the comparative value in the text field. If the value is to be taken from a result, select the check box *from Result* and select the corresponding result in the drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameters.



In the area to the left of the [Test] button the evaluation of the command (No Errors/ not equal) or an error message (for example: an involved result is undefined) is displayed.

In addition, the result to be compared is marked as "Good" or "Bad" according to the evaluation in the docking window *Results*.

After completing the configuration, add the command to the test program with the [OK] button.

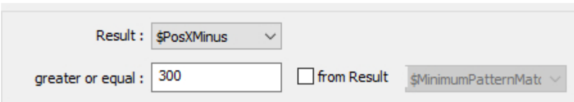
6.3.7 Result greater or equal

With **Evaluation > Result greater or equal** you write a command into the test program, which compares a result with a fixed value or with another result.

The result is evaluated as good if its value is greater than or equal to the comparative value.

Setting parameters

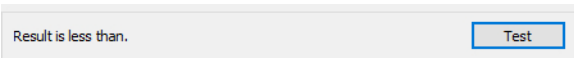
Enter the parameters in the dialog:



Parameter	Description
Result	Select the result to be compared from the drop-down list.
greater or equal	Enter the comparative value in the text field. If the value is to be taken from a result, select the check box *from Result* and select the corresponding result in the drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameters.



In the area to the left of the [Test] button the evaluation of the command (No Errors / less than) or an error message (for example: an involved result is undefined) is displayed.

In addition, the result to be compared is marked as "Good" or "Bad" according to the evaluation in the docking window *Results*.

After completing the configuration, add the command to the test program with the [OK] button.

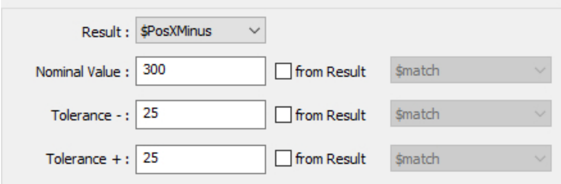
6.3.8 Result in Tolerance

With **Evaluation > Result in Tolerance** you write a command into the test program, which compares a result with a nominal value and associated tolerances. The nominal value and the tolerances can each be entered fixed or can be taken from different results.

The result is evaluated as good if its value is within the range of the nominal value and its tolerances.

Setting parameters

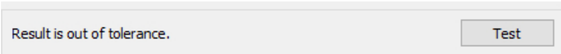
Enter the parameters in the dialog:



Parameter	Description
Result	Select the result to be compared from the drop-down list.
Nominal Value Tolerance - Tolerance +	Enter the nominal value and the permissible minimum and maximum deviation from the nominal value in the text field. If the value is to be taken from a result, select the check box *from Result* and select the corresponding result in the drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameters.



In the area to the left of the [Test] button the evaluation of the command (No Errors / out of tolerance) or an error message (for example: an involved result is undefined) is displayed.

In addition, the result to be compared is marked as "Good" or "Bad" according to the evaluation in the docking window *Results*.

After completing the configuration, add the command to the test program with the [OK] button.

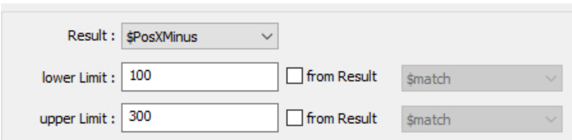
6.3.9 Result in Range

With **Evaluation > Result in Range** you write a command into the test program, which compares a result with a value range. The values of the value range can each be entered fixed or can be taken from different results.

The result is evaluated as good if its value is in the value range including its limits.

Setting parameters

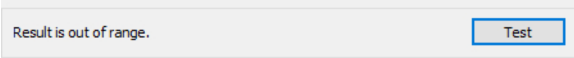
Enter the parameters in the dialog:



Parameter	Description
Result	Select the result to be compared from the drop-down list.
lower Limit upper Limit	Enter the lower and upper limit of the value range in the text fields. If the value is to be taken from a result, select the check box *from Result* and select the corresponding result in the drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameters.



In the area to the left of the [Test] button the evaluation of the command (No Errors / out of range) or an error message (for example: an involved result is undefined) is displayed.

In addition, the result to be compared is marked as "Good" or "Bad" according to the evaluation in the docking window *Results*.

After completing the configuration, add the command to the test program with the [OK] button.

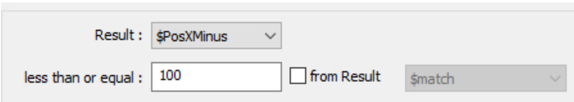
6.3.10 Result less than or equal

With **Evaluation > Result less than or equal** you write a command into the test program, which compares a result with a fixed value or with another result.

The result is evaluated as good if its value is less than or equal to the comparative value.

Setting parameters

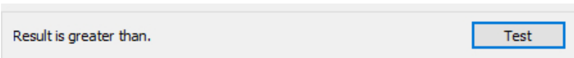
Enter the parameters in the dialog:



Parameter	Description
Result	Select the result to be compared from the drop-down list.
less than or equal	Enter the comparative value in the text field. If the value is to be taken from a result, select the check box *from Result* and select the corresponding result in the drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameters.



In the area to the left of the [Test] button the evaluation of the command (No Errors / greater than) or an error message (for example: an involved result is undefined) is displayed.

In addition, the result to be compared is marked as "Good" or "Bad" according to the evaluation in the docking window *Results*.

After completing the configuration, add the command to the test program with the [OK] button.

6.3.11 Combine Results

ADVICE

The number of entries possible in the results structure is displayed when connecting the vision system.

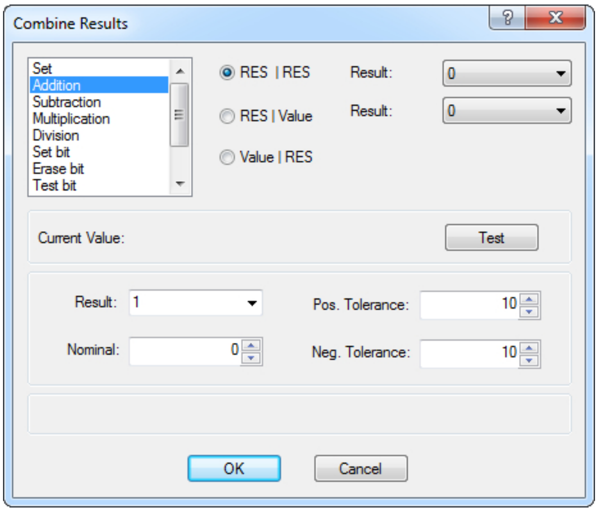


Image 143: Dialogue Combine Results

Use **Evaluation > Combine Results** to insert a command into the test program that performs mathematical operations on results and saves the outcome in a new result field, or sets one bit of a specified result.

Set

Sets an element in the result structure to any particular value.

1. Enter the value.
2. Enter the result number.

Add

Two elements in the result structure are added together or a value is added to a result.

Two Results

1. Enter the first result number.
2. Enter the second result number.
3. Enter the nominal value and +/– tolerances.
4. Enter the number of the calculated result.

Result and Value

ADVICE

The addition of 1 to a result can be used as a loop counter.

1. Enter the result number.
2. Enter the value to be added.
3. Enter the nominal value and +/– tolerances.
4. Enter the number of the calculated result.

Subtract

Subtracts either one result from another, a result from a value or a value from a result.

1. Enter the result number or value for the subtrahend.
2. Enter the result number or value for the minuend.
3. Enter the nominal value and +/– tolerances.
4. Enter the number of the calculated result.

Multiply

ADVICE

Values must be in the range between -32767 and 32767. The maximum accuracy for real numbers is 5 digits.

Multiply two elements in the result structure or multiply a result with a value.

Two Results

1. Enter the number of the first result.
2. Enter the number of the second result.
3. Enter the nominal value and +/– tolerances.
4. Enter the number of the calculated result.

Result and Value

1. Enter the result number.
2. Enter the value to multiply with.
3. Enter the nominal value and positive/negative tolerances.
4. Enter the number of the calculated result.

Divide

ADVICE

Values must be in the range between -32767 and 32767. The maximum accuracy for real numbers is 5 digits.

Divide either one result by another, a result by a value or a value by a result

1. Enter the first result number or the value of the dividend.
2. Enter the second result number or the value of the divisor.
3. Enter the nominal value and positive/negative tolerances.
4. Enter the number of the calculated result.

Set bit

One **bit** from a selected result in the **Result** field is set to 1. The result is saved in the **Result** field at the bottom left.

Example Setting		The following takes place when executing – Set bit
Bit	**Result**	
0	3	Bit 0 from result 3 is set to 1. The value in the result memory is set to 1.
2	3	Bit 2 from result 3 is set to 1. The value in the result memory is set to 4.

Erase bit

A **bit** from a selected result in the **Result** field is set to zero. The result is saved in the **Result** field at the bottom left.

Example Setting		Contents of result before command is executed	The following takes place when executing – Erase bit
Bit	**Result**		
0	3	128 (contents of result no. 3 is 128)	Bit 0 from result no. 3 is set to zero. The value in the selected memory position in the result memory remains at 128.
7	3	128 (contents of result no. 3 is 128)	Bit 7 from result no. 3 is set to zero. The value in the selected memory position in the result memory changes to zero.

Test bit

ADVICE

When using the Test bit command, set the tolerances to zero.

A bit from a selected result in the **Result** field is tested. The result (one or zero) is set in the **Result** field at the bottom left.

Example Setting		Contents of result before command is executed	The following takes place with the execution of the command – Test bit
Bit	**Result**		
0	4	128 (contents of result no. 4 is 128)	Bit 0 from result no. 4 is tested. The value in the selected memory position in the result memory changes to zero (false).
7	4	128 (contents of result no. 4 is 128)	Bit 7 from result no. 4 is tested. The value in the selected memory position in the result memory changes to one (true).

Trigonometric Functions

Trigonometric functions apply the operations sin, cos, tan, arcsin, arccos or arctan to existing results.

1. Select whether the calculation should take place in radians or degrees.
2. Combine the results according to the following principle:

$$current\ value = multiplication\ factor * operation\left[\frac{result}{division\ factor}\right]$$

Example:

$$current\ value = 1000 * \cos\left[\frac{250}{1}\right] = -\ 342\ (corresponds\ to - 0.342)$$

The result is in whole numbers. Choose the appropriate multiplication factor for the accuracy required.

Power

Power raises the result to the exponent given.

Principle:

$$current\ value = multiplication\ factor * \left[\frac{result}{division\ factor}\right]^{exponenet}$$

The result is in whole numbers! Choose the appropriate multiplication factor for the accuracy required.

Root

Root extracts the root given from the result.

Principle:

$$current\ value = multiplication\ factor * \sqrt[n]{\left[\frac{result}{division\ factor}\right]}$$

The result is an integer! Choose the appropriate multiplication factor for the accuracy required.

Evaluation Parameters

Parameters	Description
Result	Result number for saving the number of results calculated.
Nominal Value and Tolerances	Nominal value and pos./neg. tolerances allowed during the calculation. You can set suitable parameters interactively with the aid of the [Test] button.

Test

The calculated **actual value** is displayed during the test.

6.3.12 Define Line

Overview

The command **Define Line** is in the Command Selection under **Evaluation**.

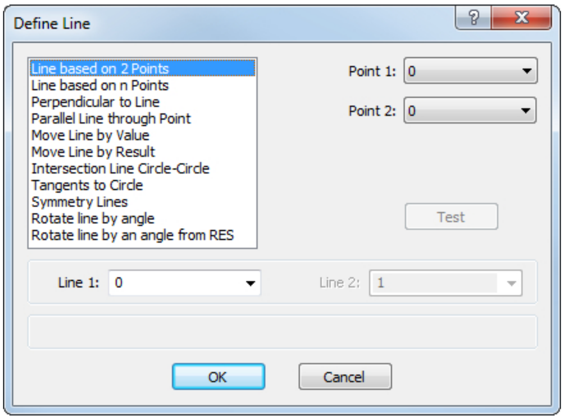


Image 144: Dialogue Define Line

Characteristics

- Calculation of straight lines from defined values.
- The calculation is made from pre-determined values (these must be present in the geometry lists).
- The calculated straight lines are stored in the line structure.
- The coordinate system used corresponds to the used variables.
- One process per command.

Line based on 2 Points

Calculating a straight line from two points.

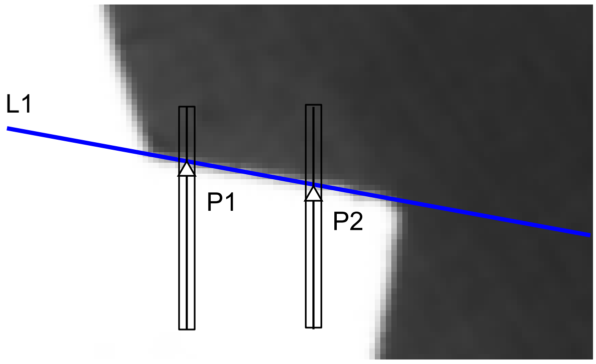


Image 145: Define Line: Line from 2 points

1. Point 1: Select the first point (P1) from the drop-down list.
2. Point 2: Select the second point (P2) from the drop-down list.
3. Line 1: Enter the number / name to store the calculated straight line (L1).

Line based on n Points

Calculate a straight line that is optimally approximated to a set of points.

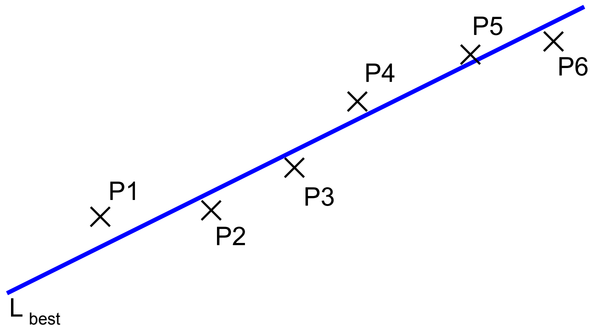


Image 146: Best fit line through a number of points

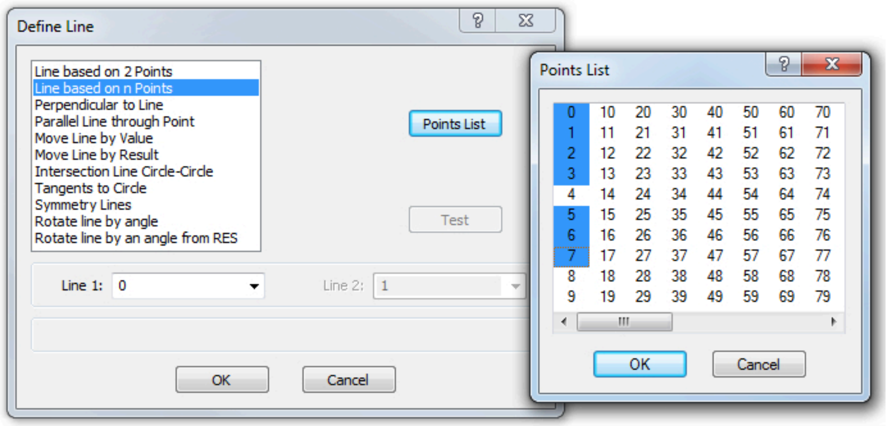


Image 147: Dialogue Define Line: Point List

1. Click on the [Point List] button to open the list of available points.
2. Select points by clicking on them. Selected points in the list appear highlighted. Click on [**OK**] to confirm selection.
3. Enter the number/name to store the calculated best fit line (L_{best}).

Perpendicular to Line

Calculating a straight line by dropping a perpendicular on a straight line through a point.

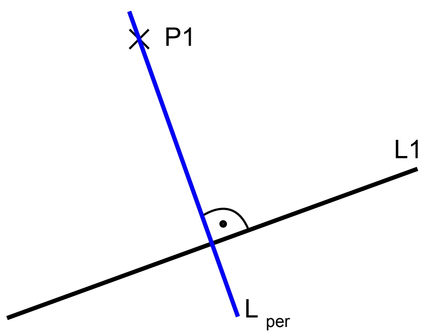


Image 148: Perpendicular to Line

1. Point: Select the point (P1) through which the perpendicular line goes from drop-down list.
2. Line: Select the start line (L1) from drop-down list.
3. Line 1: Enter the number / name to store the calculated perpendicular line (L_{per}).

Parallel Line through Point

Calculating a straight line through parallel shifting of a line through a given point.

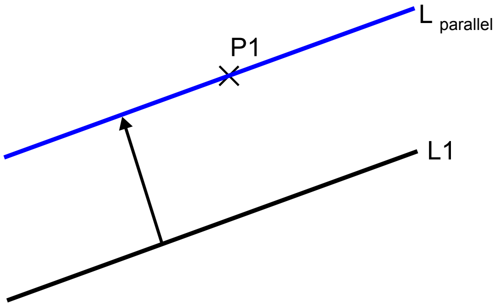


Image 149: Parallel Line through Point

1. Point: Select the point (P1) from drop-down list. The straight line is shifted to this point.
2. Line: Select the start line (L1) from drop-down list.
3. Line 1: Enter the number / name to store the calculated parallel line ($L_{parallel}$).

Move Line by Value

Calculation of a straight line through shifting a start line by a numerical value in a certain direction.

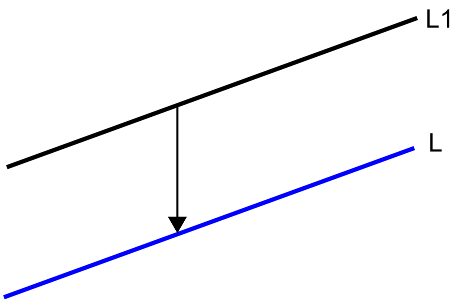
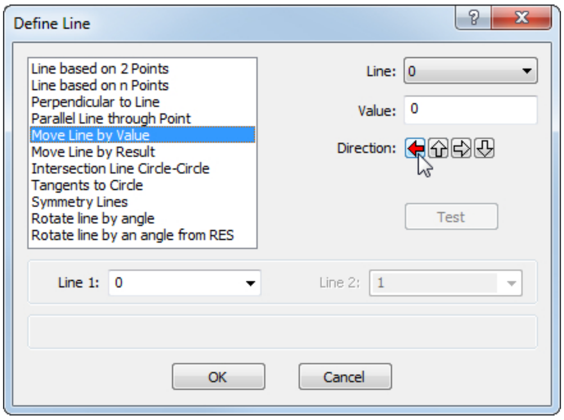


Image 150: Moving Line by Value



1. Line: Select the start line (L1) from the drop-down list.
2. Value: Enter the numeric value for the displacement.
3. Direction: Enter the movement direction for parallel movement. Use the buttons with directional arrows for this.
4. Line 1: Enter the number / name to store the calculated displacement line (L).

Move Line by Result

Calculation of a straight line through shifting a start line by a result from the result structure in a certain direction.

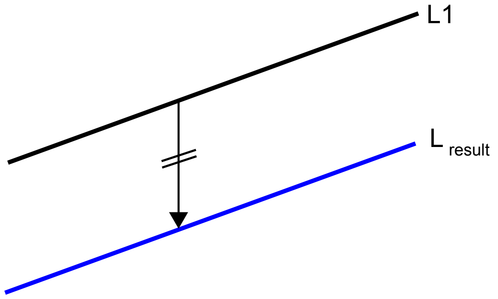
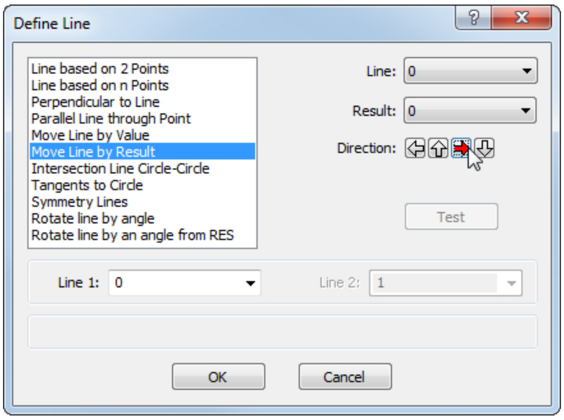


Image 151: Move Line by Result



1. Line: Select the start line (L1) from drop-down list.
2. Result: Select the result for the displacement from drop-down list.
3. Direction: Enter the movement direction for parallel movement. Use the buttons with directional arrows for this.
4. Line 1: Enter the number / name to store the calculated displacement line (L_{result}).

Intersection Line Circle-Circle

Calculating a straight line through the two points of intersection circles.

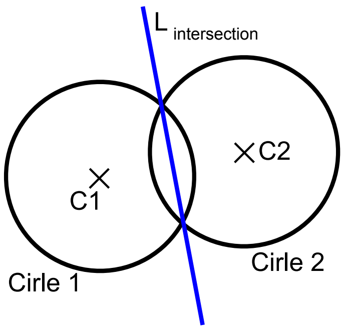


Image 152: Intersection Line Circle-Circle

1. Circle 1: Select the first circle (Circle1) from drop-down list.
2. Circle 2: Select the second circle (Circle2) from drop-down list.
3. Line 1: Enter the number / name to store the calculated intersection line ($L_{intersection}$).

Tangents to Circle

Calculation of two circle tangents passing through a given point.

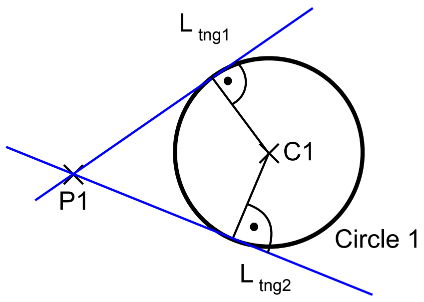


Image 153: Tangents to Circle

1. Point: Select the point (P1) through which the tangents will pass from drop-down list.
2. Circle: Select the circle (Circle1) to which the tangents are formed from drop-down list.
3. Line 1: Enter the number / name to store the first calculated circle tangent (L_{tng1}).
4. Line 2: Enter the number / name to store the second calculated circle tangent (L_{tng2}).

Symmetry Lines

Calculation of two symmetry line (bisecting lines) to two intersection lines.

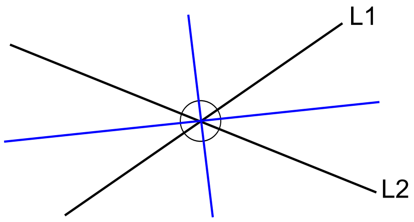


Image 154: Symmetry Lines

1. Line 1: Select the start line (L1) from drop-down list.
2. Line 2: Select the second start line (L2) from drop-down list.
3. Line 1: Enter the number / name to store the first calculated symmetry line.
4. Line 2: Enter the number / name to store the second calculated symmetry line.

Rotate Line by Angle

Calculating a rotational line from a starting line and a given angle.

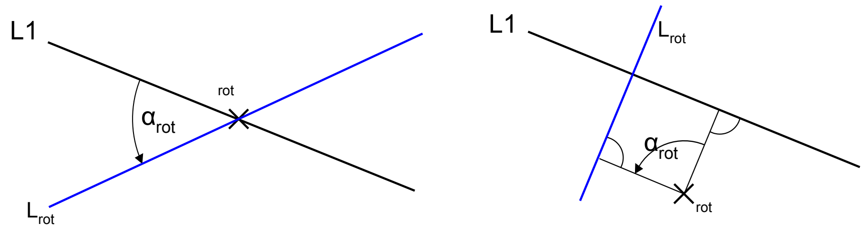


Image 155: Rotate Line by Angle

1. Line: Select the first start line (L1) from drop-down list.
2. Angle in 1/100°: Enter the angle of rotation (α_{rot}) in 1/100 degrees as an integer.
3. Apex Point: Select the rotation (P_{rot}) point from drop-down list.
4. Line 1: Enter the number / name to store the calculated rotational line (L_{rot}).

Rotate Line by an Angle from RES

Calculating a rotational line from a starting line and a result.

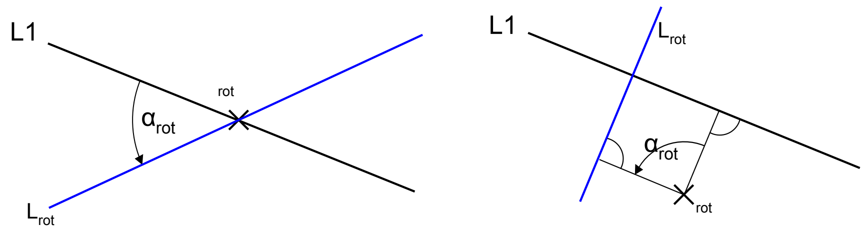


Image 156: Rotate Line by an Angle from RES

1. Line: Select the first start line (L1) from drop-down list.
2. Result: Select the result with the angle of rotation (α_{rot}) from drop-down list. The angle must be given in 1/100 degrees in the result.
3. Apex Point: Select the rotation (P_{rot}) point from drop-down list.
4. Line 1: Enter the number / name to store the calculated rotational line (L_{rot}).

6.3.13 Contour Distance

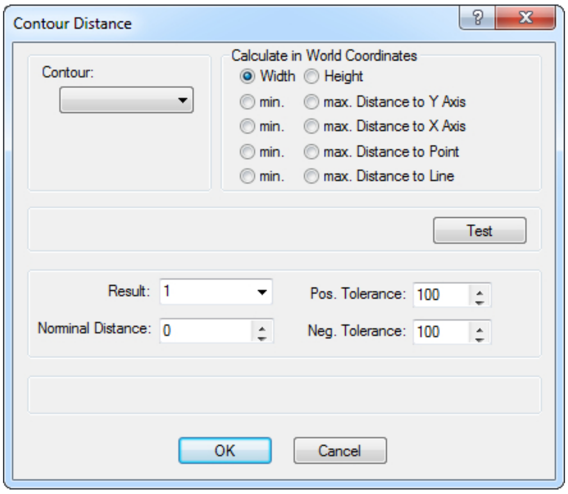


Image 157: Dialogue Contour Distance

Define a command for estimating the contour distance using **Evaluation > Contour Distance**. For this, the distance value is calculated and compared with a preset tolerance range.

Define the distance corresponding to requirements, either using the width and height of the contour or by entering the minimum or maximum distance of a given contour to the X axis, to the Y axis, to a point or to a line. Use the previously determined contours. The distances are calculated in world coordinates.

Input Parameters

Parameters	Description
Contour	Number in the contour structure in which the contour to be evaluated is stored.
Point / Line	Number of the point / line in the point / line structure for which the distance should be calculated.

Calculating in World Coordi-nates	Description
Width /Height	Specifies if the width or height of the contour should be calculated.
Min./Max. Distance to the Axis, Point or Line	Geometric element to which the minimum or maximum distance of the contour should be calculated.

Evaluation Parameters

Parameters	Description
Result	Result number for saving the calculated distance. If you select {-1} or {nd}, the data will not be stored.
Nominal Value and Tolerances	Nominal distance and permitted deviations: Set suitable parameters interactively with the aid of the [Test] button.

6.3.14 Extreme Points of Contour

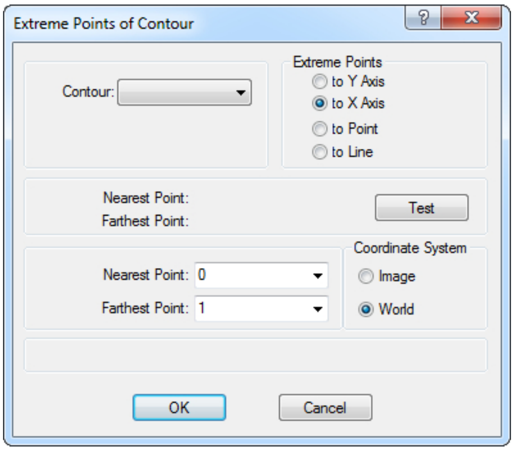


Image 158: Dialogue Extreme Points of Contour

With **Evaluation > Extreme Points of Contour** you can define a command for the calculation of the extreme points on an existing contour. Extreme points are characterized by a maximum or minimum distance to the X axis, Y axis, to a point, or to a line.

All the previously determined contours can be used for this. You can store the calculated extreme points in the point structure in image coordinates or world coordinates.

Input Parameters

Parameters	Description
Contours	Number in the contour structure in which the contour to be evaluated is stored.
Point / Line	Number of the point / line in the point / line structure for which the extreme points are to be calculated.
Extreme Points	Axis, point or line for which the extreme points are to be calculated.

Evaluation Parameters

Co-ordinate system	Description
Image/World	Calculating the extreme points in image coordinates or world coordi-nates.

The following type combinations are possible between the output geometry variables and the variables to be defined:

Points to be Defined	Contour	Point or Line
Image	Image	Image
World	Image	Image
World	Image	World

Extreme Points	Description
Nearest Point/ Farthest Point	Number for saving the coordinates of the nearest/farthest extreme point in the point structure. It is not saved if {1} is selected.

Test

The coordinates of the calculated extreme points are displayed in the test.

6.3.15 Contour Curvature Analysis

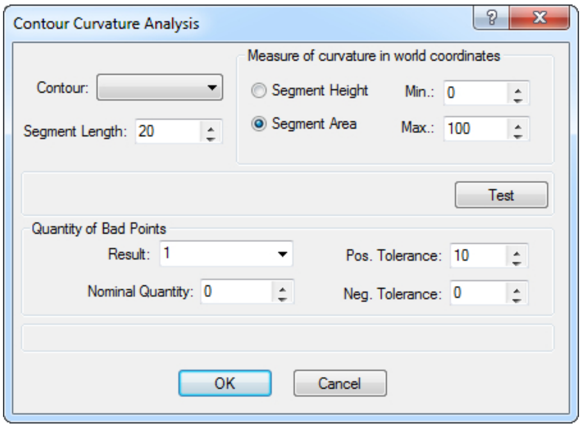
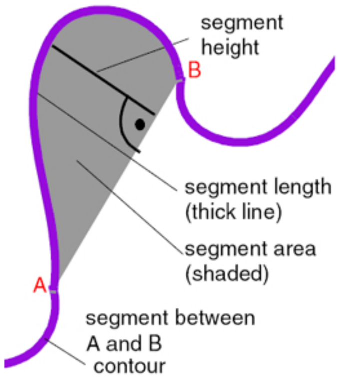


Image 159: Dialogue Contour Curvature Analysis

Write a command for estimating the curvature of a contour /contour segment into the test program using **Evaluation > Contour Curvature Analysis**. For this, the curvature along the contour is calculated and then compared with a preset tolerance range. Either the segment area or the segment height can be selected as the measure of curvature.

Every individual point of the contour is tested. Then, the number of points outside the tolerance range – referred to as bad points here – are estimated and stored in the result structure.



All the previously determined contours can be used.

Input Parameters

Parameters	Description
Contour	Number in the contour structure in which the contour to be evaluated is stored.
Segment Length	Length of the contour segment in pixels to be evaluated.

Measure of Curvature World Coordinates	Description
Segment Height	Uses the segment height as the measure of curvature in world coordinate.
Segment Area	Uses the segment area (in pixels or world units) as the measure of curvature.
Min./Max.	Permitted minimum and maximum measures of curvature in pixels (valid for either segment height or segment area). The suitable parameters for the measure of curvature (Min./Max.) are defined interactively with the [Test] button. In which, the bad points are displayed on the video screen and their number is displayed in the curvature analysis teaching dialog.

Evaluation Parameters

Quantity of Bad Points	Description
Result	Result number for saving the number of bad points.
Nominal Quantity and Tolerances	(Permitted) nominal number of bad points and tolerances. Use the [Test] button to establish that the parameters that are suitable.

6.3.16 Center of Gravity of Contour

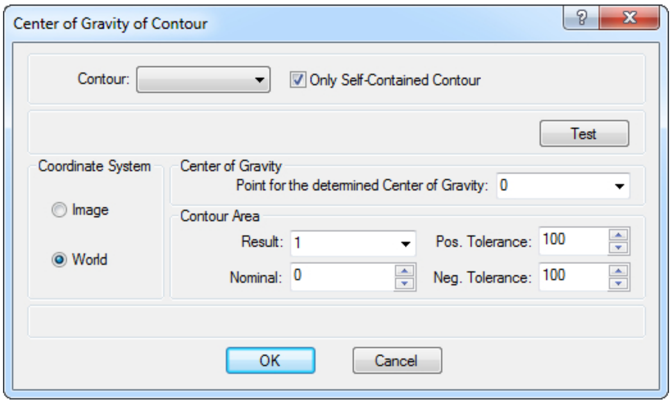


Image 160: Dialogue Center of Gravity of Contour

You can write a command for calculating the center point and the area of an existing contour into the test program using **Evaluation > Center of Gravity of Contour**.

To do this, use the open or closed contours which have already been determined (see Locate Contour command). The calculated center point can be stored in the point structure in image coordinates or world coordinates.

Test the contour area by setting a nominal value and the tolerance range. The unit for area depends upon the coordinate type of the center point defined and can be pixels or in world units.

Input Parameters

Parameters	Description
Contour	Number/name in the contour structure in which the contour to be evaluated is stored.

Only Self-Contained Contour	Description
Option marked	Only closed contours can be used. Otherwise the command fails.
Option unmarked	All contours can be used.

Evaluation Parameters

Coordinate system	Description
Image/World	Defines whether the Center of Gravity or the Contour Area are calculating in image coordinates or world coordinates.

Parameters	Description
Point for the determined Center of Gravity	Number/name for saving the coordinates of the calculated center point in the point structure. If you select {-1} or {nd}, the data will not be stored.

Contour Area	Description
Result	Result number/name for saving the area enclosed by the contour: If you select {-1} or {nd}, the data will not be stored.
Nominal Value and Tolerances	Nominal area and permitted area deviation: Set suitable parameters interactively with the aid of the [Test] button.

Test

The test displays:

Parameters	Description
Center of Gravity Point	Calculated coordinates of the gravity center point.
Area	Calculated contour area. Hint: You can use the results for the setting of the nominal value and its tolerances.

6.3.17 Contour Comparison / Contour Angle

ADVICE

Use the **Locate > Locate Contour** command to store contours in the contour buffer.

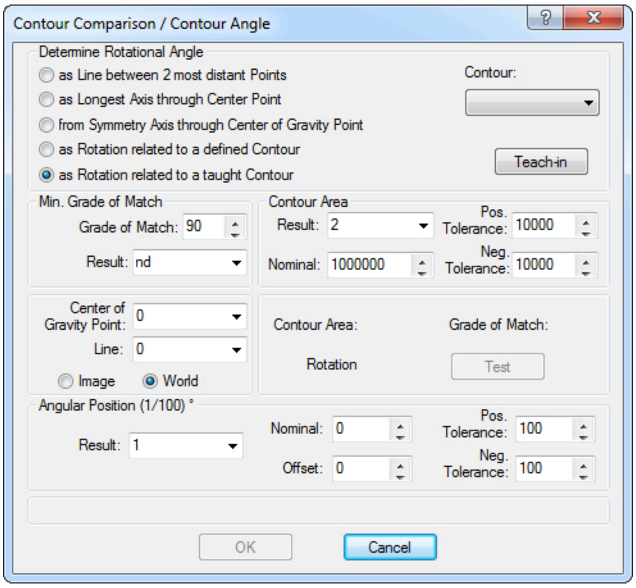


Image 161: Dialogue Contour Comparison / Contour Angle

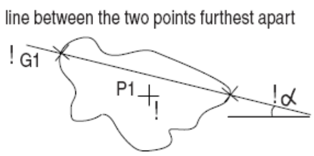
Define a command for assessing the contour angular position using **Evaluation > Contour Comparison / Contour Angle**. For this, you need a closed contour which is loaded from the contour buffer.

To determine the contour angular position, a reference line is determined and either set in relation to the X axis or compared with the position of a defined nominal contour. The gravity center point of the contour and the contour area are automatically determined and can be assessed and stored.

The contour angular position can be calculated in image or world coordinates.

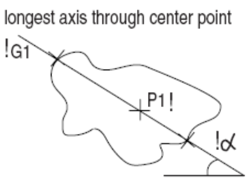
Selecting a Method

From a Line Between the Two Points Furthest Apart



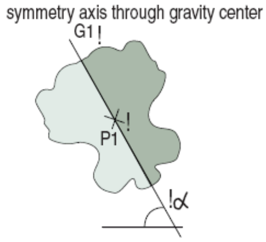
The reference line is defined by the two points furthest apart in the contour.

From the Longest Axis Through the Center Point



The reference line is defined by those two points of the contour which are furthest away from each other where the line between them also passes through the center point.

From a Symmetry Axis Through the Gravity Center Point



The reference line is determined by a symmetry line that divides the contour in two halves, according to a certain algorithm. Use this option for symmetric contours.

As a Rotation from a Previously Defined Contour

The rotation position is determined by comparison with a nominal contour. The nominal contour is loaded from the contour buffer as is the contour to be tested.

As a Rotation From a Taught-in Contour

The angular position is determined by comparison with a taught-in contour. The nominal contour must have previously been taught-in with the [Teach-in] button and stored in the test program.

Input Parameters

Parameters	Description
Contour	Number in the contour structure in which the contour to be evaluated is stored.
Nominal Contour	Number in the contour structure in which the desired contour is stored. Only required for the »as Rotation related to a defined Contour« method.

Co-ordinate system	Description
Image/World	Calculates in image coordinates or world coordinates.

Evaluation Parameters

Min. Grade of Match	Description
This command is to configure a test of the correspondence between the contour found and the nominal contour.	
Covering Rate	The minimum degree of coverage in per cent when the contour parts are reflected around the symmetry axis. Only required for the »Symmetry Axis through Center of Gravity Point« method.
Grade of Match	Correlation with the nominal contour. Only required for the two »Rotation related to a Contour« methods.
Result	Result number for the result of the comparison in the correspondence test: If you select {-1} or {nd}, the data will not be stored.

Contour Area	Description
Result	Result number for saving the area enclosed by the contour: If you select {-1} or {nd}, the data will not be stored.
Nominal Value and Tolerances	Permissible deviation of the area from the nominal area: Set suitable parameters interactively with the aid of the [Test] button.

Center of Gravity Point and Line	Description
Point / Line	Number in the point or line structure under which the determined center point of contour and the reference line determined by the relevant process are stored.

Angular Position (1/100 degree)	Description
Result	Result number for saving the determined angular position (1/100 degree).
Nominal and Tolerances	Nominal angular position and the permitted ± tolerances. Set suitable parameters interactively with the aid of the [Test] button. For technical reasons, tolerances cannot exceed 32768 in value (corresponds to 215). The tolerance range is from +32767 to –32767. The angular position in the first half circle is from 0° to 180° clockwise and further from -179° to 0° clockwise. The value given by the program should be divided by 100 to maintain the correct angle in degrees (°). This division factor should also be used when superimposing. Examples: 9000 corresponds to +90° / –9000 corresponds to –90°. See figure »Angle Definition« under the table.
Offset	Offset for the angular position (1/100 degree). If an initial rotation is already present in the loaded contour, the offset may be added in order to relate the angular position determined to the X or Y coordinate.

Angle Definition

ADVICE

The line is saved in a way that 0° corresponds to a parallel to the X axis. The angle of rotation is relative to the taught contour in a proportion of 1/100, counterclockwise.

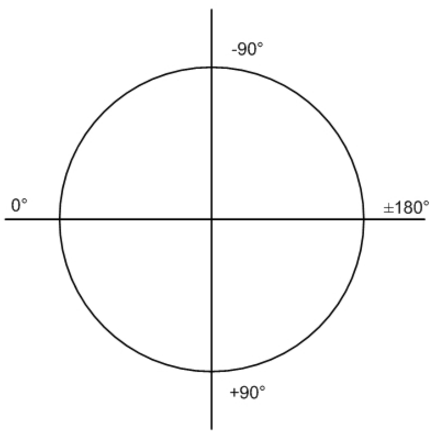


Image 162: Angular Position Line Output

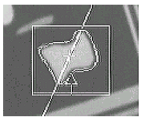
Test

The test displays:

Parameters	Description
Contour Area	Calculated contour area.
Rotation Angle	Calculated value of angular contour position.
Grade of Match	Actual correspondence when testing the min. grade of match.

Example

The reference line determined for determining the angular position of the contour is displayed during the test.



Example: Reference line for the rotation position of the contour during the test with »Longest axis through the center point«

6.3.18 Define Circle

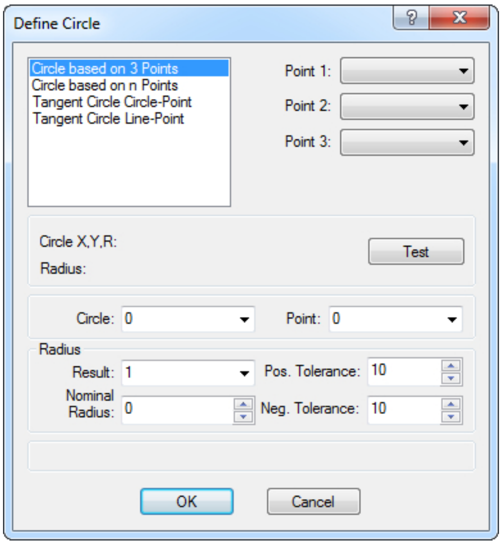


Image 163: Dialogue Define Circle

Use **Evaluation > Define Circle** for creating new circles from existing points, lines and circles in the test program. You can use previously determined points, lines, and circles for this.

The generated circles are saved in image or measurement units in the circle structure under a circle number. The coordinate system of a resulting circle corresponds to the coordinate system of the original variables.

Selecting a Method

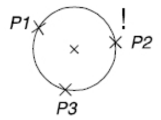
The following methods are possible:

- Circle based on 3 points
- Circle based on n points
- Tangential circle line-point
- Tangential circle circle-point

Circle Based on 3 Points

This defines a circle using three points.

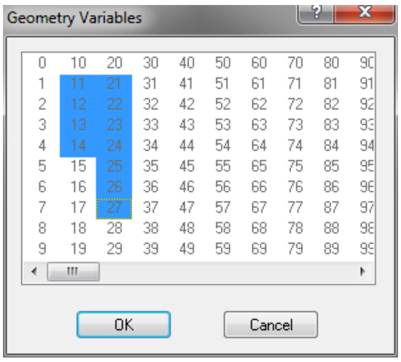
Circle through three Points:



1. Enter the numbers of the points P1, P2 and P3.
2. Enter the nominal radius and +/- tolerances.
3. Enter the result number for the radius if required (otherwise {nd}).
4. Enter the number of the new circle.
5. If the circle center should be saved, enter die point number, otherwise enter or select {–1} or {nd}.

Circle Based on n Points

This option calculates the optimum circle from a list of points.

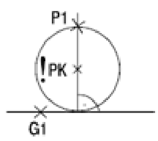


1. Click on the [Point List] button to open the list of available points (see fig.)
2. Select points by clicking on them. Selected points in the list appear highlighted. Click on [OK] to confirm selection.
3. Enter the circle number.
4. Enter the nominal radius and +/- tolerances.
5. Enter the result number for the radius if required (otherwise {nd}).
6. Enter the number of the new circle.
7. If the circle center should be saved, enter die point number, otherwise enter or select {–1} or {nd}.

Tangential Circle Line-Point

A command for the computation of a tangential circle from a point and a line is written into the test program.

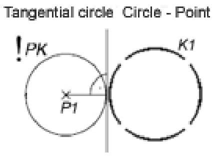
Tangential Circle with Line



1. Enter the point number.
2. Enter the line number.
3. Enter the nominal radius and +/- tolerances.

4. Enter the result number for the radius if required (otherwise {nd}).
5. Enter the number of the new circle.
6. If the circle center should be saved (see point PK in the illustration), enter die point number, otherwise enter or select {–1} or {nd}.

Tangential Circle Circle-Point
A command for the computation of a tangential circle around a specified point (functions as the circle's center point) to the closest tangent on a circle is written into the test program.



1. Enter the point number.
2. Enter the circle number.
3. Enter the nominal radius and +/- tolerances.
4. Enter the result number for the radius if required (otherwise {nd}).
5. Enter the number of the new circle.
6. If the circle center should be saved (see point PK in the illustration), enter die point number, otherwise enter or select {–1} or {nd}.

Evaluation Parameters

Parameters	Description
Circle	Number under which the circle found (after the detection) will be stored in the circle structure.
Point	Number for storing the coordinates of the center of the circle in the point structure. If you select {-1} or {nd}, the data will not be stored.

Radius	Description
Result	Result number under which the calculated radius will be stored: If you select {-1} or {nd}, the data will not be stored.
Nominal Radius and Tolerances	Nominal radius and the allowed ± tolerances in pixels: Set suitable parameters interactively with the aid of the [Test] button.

Test

The test displays:

Parameters	Description
Circle X,Y,R	Calculated coordinates X, Y, and radius of the circle.
Radius	Calculated circle radius.

6.3.19 Write back Parameter

With **Evaluation > Write back Parameter** you write a command into the test program, which writes the current value of the assigned geometry variable into the respective parameter of the active Parameter-Set-Definition. Use the command for interactions with the webHMI (*see "Section inspection - actions", page 407*).

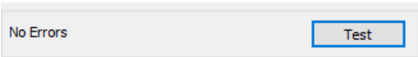
Setting Parameters

Enter the parameters in the dialog:

Parameter	Description
Parameter	Select the parameter from the drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameter.



In the area to the left of the [Test] button the evaluation of the command (No Errors) or an error message is displayed.

After completing the configuration, add the command to the test program with the [OK] button.

6.3.20 Define Point

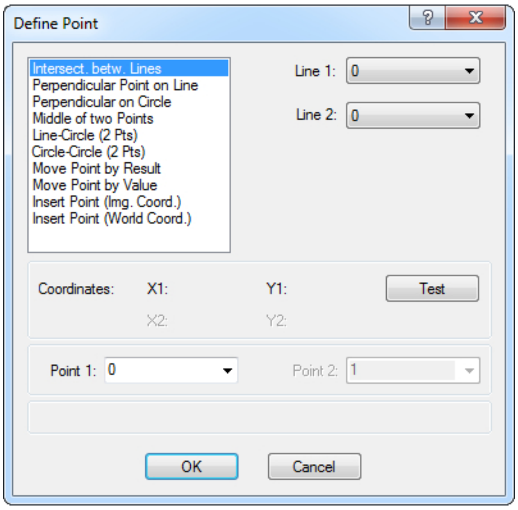


Image 164: Dialogue Define Point

Use **Evaluation > Define Point** to enter the command for creating new points in the test program. For this, use points, lines, planes, and circles that have already been detected or enter the coordinates directly.

The calculated points are stored as image or world coordinates under a user-defined number within the point structure. The coordinate system used corresponds to the coordinate system of the source elements.

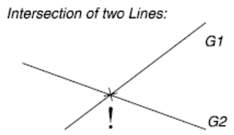
Selecting a Method

The following methods are possible:

- Intersection of two lines
- Intersection of a perpendicular from a point to a line
- Intersection from a point to the center of a circle
- Center of a line connecting two points (i.e. average)
- Two intersections between a line and a circle
- Two intersections between two circles
- Addition of a result to a point (i.e. shift)

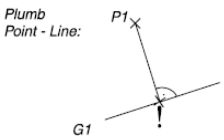
- Addition of a fixed value to a point (i.e. shift
- Direct entry in image coordinates
- Direct entry in world coordinates

Intersection of Two Lines



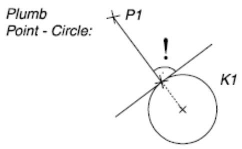
1. Enter the first line number.
2. Enter the second line number.
3. Enter the number of the new point.

Plumb Point - Line



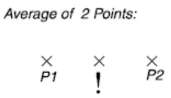
1. Enter the point number.
2. Enter the line number.
3. Enter the number of the new point.

Plumb Point - Circle



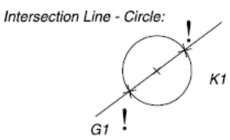
1. Enter the point number.
2. Enter the circle number.
3. Enter the number of the new point.

Average of Two Points



1. Enter the first point number.
2. Enter the second point number.
3. Enter the number of the new point.

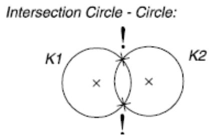
Intersection Line - Circle



1. Enter the line number.
2. Enter the circle number.

3. Enter the number of the first intersection point.
4. Enter the number of the second intersection point.

Intersection Circle - Circle



1. Enter the first circle number.
2. Enter the second circle number.
3. Enter the number of the first intersection point.
4. Enter the number of the second intersection point.

Shift Point
This option copies a point to a new position, creating a new point. The original point is not deleted.

1. Enter the number of the point which is to be shifted.
   - Enter the offsets (either numeric input in image points or enter the result from which the value should come).
   - Enter the value or result number for the Y offset
   - Enter the value or result number for the Y offset
   - Enter the value or result number for the XY offset
2. Enter the number of the new point.

Entering Coordinates Directly

1. Set the coordinate system: image coordinates or world coordinates
2. A: Enter image coordinates:
   - X coordinate
   - Y coordinate
   - Camera number
3. B: world coordinates
   - X coordinate
   - Y coordinate
4. Enter the number of the new point.

Evaluation Parameters

Evaluation	Description
Point	Number(s) or names under which the point(s) found will be stored in the point structure.

Test

The coordinates of the calculated point(s) are displayed during the test.

6.3.21 Check Point Distances

Overview

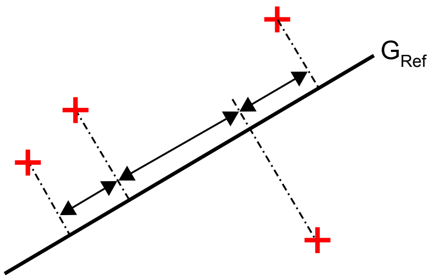


Image 165: Point Distances parallel to Reference Line

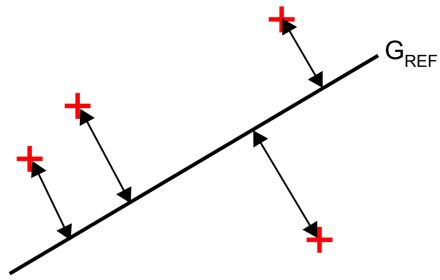


Image 166: Distances to Reference Line

The command **Evaluation > Check Point Distances** calculates the distance between nearby points to each other and the distances from points to a reference line.

Function

1. If no reference line is specified, a reference line is calculated from the given points.
2. The points are sorted along the reference line.
3. Testing of the distances between nearby points.
4. Testing of the distances of the points to the reference line.
5. Display of the determined values.
   - Independent from the chosen coordinate system the output is in world coordinates.
   - Only the absolute value is determined. Whether a point is above or below the reference line will not be calculate.

Dialogue structure

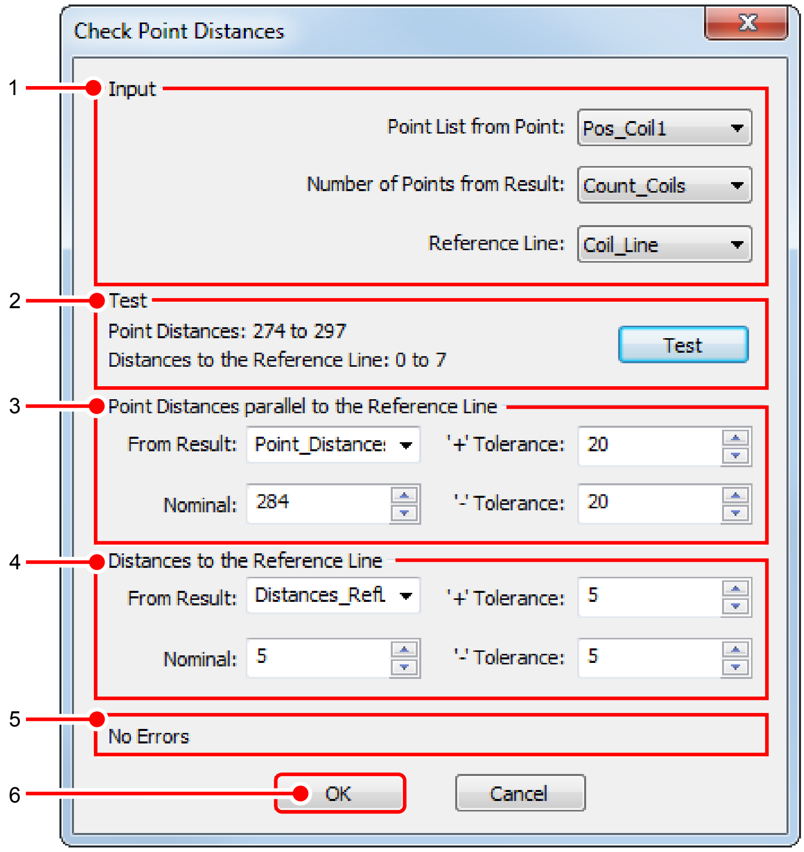


Image 167: Dialogue Check Point Distances

Work Flow: Parametrize

1. Enter here the variables to be checked.

Parameter	Description
Point List from Point	Specify here the point from which the calculation starts.
Number of Points from Result	Enter the result in which the amount of coils is saved. This corresponds to the number of points in the point list.
Reference Line	Enter here the reference line. If no reference line is specified, a reference line is calculated from the given points.

2. Test the settings here

   After activating the [Test] button the variance of the point distances from one another and the distances to the reference line are displayed in this area.

3. In the area "**Point Distances parallel to the Reference Line**" enter the requirements for the assessment of the command and to store the results.

Parameter	Description
From Result	Number / name for storing the determined distances. If you select {nd} no storage takes place.
Nominal	Nominal distance between the points.
'+' Tolerance	Allowable upper deviation from the target value.
'-' Tolerance	Allowable lower deviation from the target value.

4. In the area "**Distances to the Reference Line**" enter the requirements for the assessment of the command and to store the results.

Parameter	Description
From Result	Number / name for storing the determined distances of each point on the reference line. If you select {nd} no storage takes place.
Nominal	Nominal distance of points to the reference line.
'+' Tolerance	Allowable upper deviation from the target value.
'-' Tolerance	Allowable lower deviation from the target value.

5. Evaluation Area

   Here you can see the good/bad evaluation of the command, notes for parametrization (tolerance range exceedance) as well as error messages.

6. Use the [OK] button to insert the command into the test program.

Error messages

Possible error messages:

Distances outside the tolerance range

- The determined distances are outside of the specified nominal range (incl. tolerances).
- Adjust the tolerances according to the determined values. Check the set value of good parts and bad parts.

Input variable not defined

- At least one input variable is not defined.
- Make sure all variables are defined in previous commands. If they are not defined, execute the commands again. The variables must be stored in the Geometry List.

Reference line can not be calculated

- No reference line was given and less than 4 points were assigned.
- No reference line was given and the variance of the points around the calculated line is too large.
- Use more points to determine the reference line.

Insufficient number of points

- Less than two points were assigned for the distance calculation.
- Assign more points for the determination of the distances.

6.3.22 Calculation Script (Ruby)

The Calculation Script command offers the option of writing test commands in the form of plain text instructions with the Ruby 1.9 programming language. For this purpose, the calculation script can access saved geometry variables of the vision system or parts of the current image. The advantage over other evaluation commands of vcwin is the high complexity which can be achieved using the many integrated functions.

ADVICE

Licence information: Ruby and its source code can be downloaded from the website *http://www.ruby-lang.org*. This is not a prerequisite for the calculation script command in vcwin, however.

ADVICE

When using this command, a basic knowledge of script languages like Ruby or JavaScript is advantageous. For comprehensive information on Ruby and object-oriented programming, visit *http://www.ruby-lang.org/en/documentation/*.

ADVICE

A list and description of the attributes and methods of the Calculation script can be found in the Appendix.

The basics

The object-oriented Ruby language represents the basis of the command. It can be used to compose instructions for evaluating geometry variables or image information in plain text. Object-oriented means that all values or numbers used in Ruby are objects. These objects are defined by classes which give the individual objects certain properties and so-called methods which can be applied to the objects. Objects which originate from a certain class are instances of this class (class instances) and can be defined with different properties. They feature the same applicable methods, however. The instances, which contain the status of the object, are referenced by variables. The following example summarises the use of the Integer class, which defines a data type: A class (Integer) creates an instance whose value (e.g. "50") and properties (whole number) are referenced by a variable (e.g. blob number) and to which the methods can be applied in order to manipulate the object or create new ones.

The calculation script command provides you with classes with their own attributes and methods specifically for vcwin. A majority feature the well-known geometry variables Result, Point, Line, Circle, Contour, Counter and String. Another is the Image class. Standard classes of Ruby, such as Math, Integer, Float and String can be used. Each class defines methods which can only in some cases be applied to the class itself (e.g. to create a new instance) or only on its instances (e.g. to manipulate the value). Attributes are features of an object which are also passed to an instance of the class, e.g. the co-ordinate system being used for a point.

Method call syntax:

Instance.Method(Parameters)

Instance describes the object or the variable to which the method is applied. Method is the designation of the method, which is always preceded by a dot [.]. The parentheses after the method contain parameters to be passed to the object if they were defined in the class. If parameters do not need to be passed, they remain empty or are left out.

The access to attributes is a class-based method call and therefore uses the same syntax.

Access to geometry elements

To access saved geometry elements with the calculation scripts, the following character sequence is used:

VC::GeometryElement[No.]

For *GeometryElement*, either Result, Point, Line, Circle, Contour or String is used depending on the desired element, and the corresponding number of the element is used for *No.*. Assignment of the geometry variable to a variable is then carried out using an equal sign, which then references the value of the geometry element. To save the co-ordinates of point 15 to variable a as an instance, for example, the following code must be used:

a = VC::Point[15]

Variable a then contains the co-ordinates of point 15 and is a reference to the instance of class VC::Point. Since a point is comprised of two co-ordinates and operation can only be applied to individual numerical values, the individual numerical values must be individually accessible. For points, the following values can be output using methods .x and .y:

a = VC::Point[15].x
b = VC::Point[15].y

Now the value of the x-co-ordinate is saved in variable a, and the value of the y-co-ordinate is saved in variable b. Since concrete numerical values were now saved in both variables, they are instances of the Float class, not of the VC::Point class.

The other geometry elements also possess methods so that they can output individual numerical values or strings.

Geometry element	Output value(s)	Methods
Results	Result value	.value
Points	Co-ordinates of the point (x, y)	.x, .y
Lines	Co-ordinates of the start and end points (x1, y1, x2, y2)	.x1, .y1, .x2, .y2
Circles	Co-ordinates of the centre point and radius (x, y, r)	.x, .y, .r
Strings	Character string	.string
Contours	Point	[pointnumber]
Counter	Counter value	.value

Methods for outputing numerical values

Variables and references

ADVICE

All defined variables are deleted again after the calculation script is exited and cannot be used by other calculation script commands.

To handle geometry values in a better way, we recommend always creating variables with meaningful names. This enables you to access them at any time and have full control of the objects available to you. If variables are declared but do not describe an object, they have the status nil. This also applies for undefined geometry elements.

With variables, calculation operations can also be carried out or conditions implemented.

The variables in Ruby are not static storage locations for instances, rather they function as references to a respective object and its content. This enables you to create a relationship between two objects very easily. For example:

a = VC::Result[2]
b = VC::Result[4]

a = b

With this code, the result numbers 2 and 4 were first assigned to variables a and b, respectively. The result values are 10 (result 2) and 40 (result 4) in this example. As a result of the equation a = b, not only does instance b no longer refer to the output object VC::Result[4], but instance a does as well. Both instances now have the value 40 and reference the same object, which means that a change to one instance changes the other as well. To continue the example:

b.value += 4

This line changes the value of the result value of variable b by +4. Since a and b now reference the same object as a result of line a = b, the in the variable a referenced result 4 in the vision system VC::Result[4] increases by +4. If you then output the values of the variables via puts a.value and puts b.value to the output window, the value 44 is yielded for both.

References can also be broken under certain conditions. If, for example, you assign a new instance to variable c (which references the same instance as variable d) using a class method such as new, variable d retains its old value, but variable references the new instance. Now the variables do not reference the same instance anymore and will display different values in the output window. When an instance is not referenced by a variable any longer, it gets deleated.

To avoid undesired referencing errors, you can use the .clone method. Here is the example from above, except this time with the .clone method:

a = VC::Result[2]
b = VC::Result[4]

a = b.clone

b.value += 4

.clone causes the VC::Result object instantiated by b on the third line to be copied and transferred to a. Instance a now references a copied object and no longer the same thing as b . If b is then increased by +4, a retains the old value of b. Upon output, a has the value 40 and b has 44.

Working with the dialog

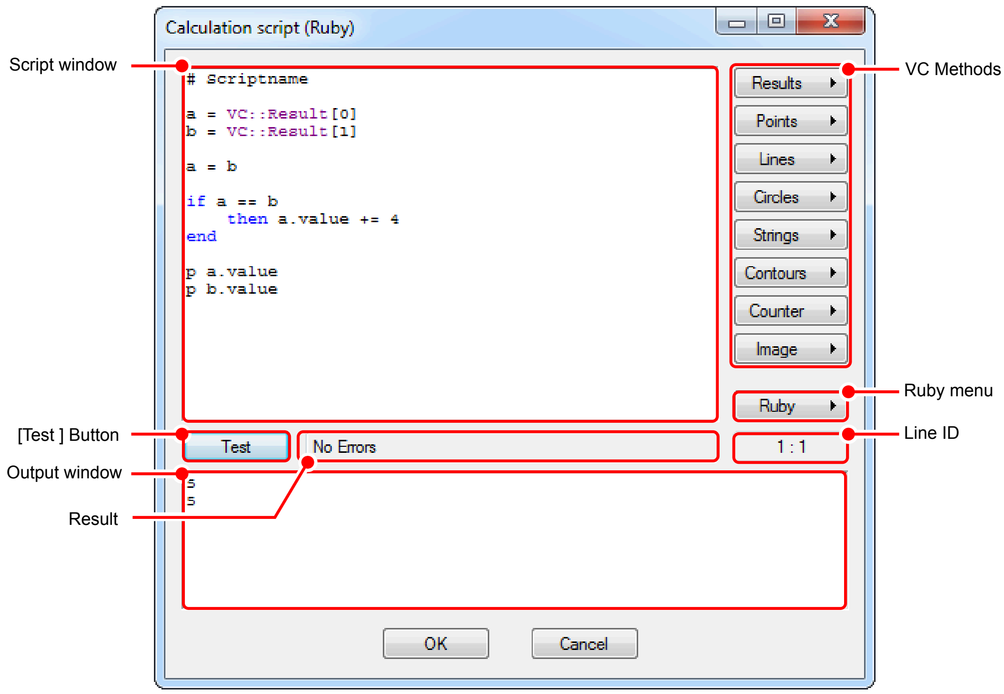


Image 168: Dialog Calculation Script

Script window

When opening a new calculation script command, the comment "# Script Name" is located in the first line. Enter the name of your script here. The first line is written as description in the vcwin programlisting. So that several calculation script commands in the vcwin program can be differentiante. If the first line contains no comment, instead the first 100 characters of the first line is written as description in the vcwin programlisting.

You enter the code for your calculation script in the script window. Pressing [ENTER] adds line wraps, and the Tab key allows you to indent lines of code for clearer code formatting. Normal code, such as variables, operators and methods are displayed in black, whereas namespaces for accessing geometry variables or image content intrinsic to vcwin in red and system terms intrinsic to Ruby, such as loop expressions, in blue.

Test button and test result

With the Test button, you can check the current code for syntactic correctness and display the result of the command to the right of the button. The result can be true or false here, but the conditions for it must be defined by the user using the return command. If, for example, a certain condition in the calculation script is to be decisive for the result of the command, whether or not the command is evaluated as good or bad after the condition arises must be specified in the code with return true or return false. If the code cannot be executed due to syntax errors, the error type with the corresponding line is displayed in the output window and the message "Ruby interpreter error" appears when the test results are provided.

Output window

In addition to error descriptions (which are output in red), the output window displays the output generated with the calculation script and can thus be used as a log for debugging. Using the puts command, variable contents or character strings can be displayed in the output window to check, for example, whether loops or conditions are running as intended or to test individual variables. The puts command can also display manipulations via methods or attributes. A version better-suited for debugging is the short form p, as it outputs the object as a string and thus also displays empty variables such as nil.

Output is provided after successful processing of the line in which the instruction is located if the written script is free of syntax errors. A few examples:

Calculation script	Output
puts "condition fulfilled"	Conditions fulfilled
a = 5 + 4 puts a	9
VC::Result[5].value = 15 p VC::Result[5].value	15

Line ID

Using the line ID, you can read where you are in the script with the cursor at any time. The first number indicates the line, and the second the position of the line. Since the line of the error is specified with error messages in the selection window, you can quickly and easily locate the sought line and remedy the error using the line ID.

Ruby menu

The Ruby menu contains methods of standard classes which can be applied to the classes intrinsic to vcwin or their instances. Methods of the Math class for advanced calculation operations, methods for whole numbers (Integer) and floating-point numbers (Float) and methods for strings are available.

The methods can very easily be inserted in the calculation script at the current cursor position with a mouse click. A comprehensive list and description of the methods can be found in the appendix.

Methods

Here you can find all methods and additional templates intrinsic to vcwin sorted by the respective classes. Using these menus, you can very easily insert the desired method into the calculation script at the current cursor position with a mouse click. Geometry elements saved in vcwin or the vision system are displayed in the menu of the corresponding class and can also be inserted into the code with a mouse click if it reduces the writing expenditure. If you would like to use an element in your script which has not yet been defined, you can use the template, in which you need only add the element number.

A comprehensive list and description of the methods can be found in the Appendix.

Advanced operation

Define geometry elements

To define new geometry elements or set a value to a undefined geometry element, you have to create a new instance using a class method, which automatically sets a new geometry element in the Vision System and assigns the specified value. Example:

VC::Result[5] = VC::Result.new(10)

With this code, you assign the value 10 to the new result geometry element number 5. The method call differs depending on the class you want to create, for example the point class method to create a new instance in world coordinates is VC::Point.world(x, y). A comprehensive list and description of all the methods can be found in the Appendix.

If you want to add a new value to an already existing element, you can do this as described above or by manipulating the values directly. This works by using special class methods, such as .value of the VC::Result class which accesses the result value, or the .x method of the VC::Point class which accesses the x coordinate of a point:

VC::Result[5].value = 20

VC::Point[1].x = 5

A comprehensive list and description of the methods can be found in the Appendix (*see page 424*).

To set a defined geometry element to undefined, you have to assign the nil expression to the element. When assigned, the geometry element does not store any values and is not defined. When the calculation script wants to process this element, it throws an exception error, cancels the operation and returns "false" as result of the command. Example:

VC::Point[1] = nil

Now the point element number 1 is not defined anymore and the instance stores the nil value. Using the puts command to output the VC::Point[1] instance, no value will be shown in the output window. When you use the command p VC::Point[1], the output will display nil (value as a string). All undefined geometry elements are nil by default.

Calculation operations

Using so-called operators, you can execute calculations with the values from geometry elements in the calculation script. The following table contains an overview of the individual operations:

Calculation operation	Operator (symbol)	Syntax
Addition	+	b = a + 5
Subtraction	-	a = b - 3
Multiplication	*	c = 3 * b
Division	/	a = b / 3
Power	**	c = 3**3
Modulo (division with remainder)	%	c = a%b
Increase/decrease variable	+= / -=	a += 4

The operations + and += can be applied to strings in the same way, e.g. to concatenate individual character strings to one another.

Lists and arrays

You can use arrays to save several elements within an object. In arrays, several elements are stored in a type of list which you can access using an index. You can thus, for example, save consecutive result values within a variable and evaluate them using loops and conditions. Arrays can be implemented in two ways:

Range: a = VC::Result[2..5] (saves the defined VC::Result objects of element numbers 2 through 5 in variable a)

Start and count: a = VC::Result[2,4] (saves four result values starting with element number 2, i.e. 2, 3, 4, 5)

You can also use arrays to define several elements at once, as shown in the following example:

a = VC::Result.new(10)
VC::Result[2..5] = a

Results 2–5 thus all refer to the same VC::Result object and are all given the value 10.

Loops and conditions

Loops are important control structures in programming languages. They repeat an instruction or an instruction block until a certain condition is valid or until a break condition arises. They are especially suitable for running through multiple values or for repeating certain program parts. Ruby provides a number of loops (e.g. for loop, while loop, etc.), but they will not be discussed in more detail at this point.

Conditions enable branching in the program flow and are a common component of loops. It executes code depending on a certain value. Example: If a condition is fulfilled, execute code part A. If not, execute code part B. Conditions can be implemented in the same way as loops in various ways (e.g. if-then-else, unless).

Logical operators which place a certain requirement on one or more values are used to implement conditions. Common logical operators include == (equal to), != (not equal to), > (greater than), < (less than), >= (greater than or equal to) and <= (less than or equal to).

The following is a practical example for implementing an each do loop which runs through an array of references to point elements of the vision system and changes the values of the array using an if-then-else condition.

```
pointlist = VC::Point [10,20]
pointlist.each do
   |point|
   if point != nil then
      point.x = point.x+100
   end
end
```

In the first line, the point elements with numbers 10–29 are saved in an array with the designation pointlist. The loop each do is applied to this array. It runs through all elements of the array and designates it internally as a point, which was defined in the third line. The condition starts in the fourth line with keyword if. This is followed by the condition point != nil. This expression describes the requirement that the elements of the array may not be empty (i.e. all point elements must be defined and contain co-ordinates). If this condition is fulfilled, then the following instruction block is executed, which in that case contains only the instruction point.x = point.x+100. This shifts the respective point by the value 100 in the x-direction. The end in the penultimate line ends the condition block, whereas the end in the last line ends the each do loop.

In this case, there was no else block for the so-called if-then-else condition. This means that if the if condition was not fulfilled, no instructions were executed (i.e. no shifting of the "empty" point), and the next element of the array was subsequently tested.

To break a loop when a condition has arisen, use the command break inside this condition. If a result for the test command is obtained through the fulfillment of a condition or the complete run-through of a loop (e.g. if all points, without exception, were shifted in direction x), insert the instruction return true or return false into the respective instruction block to obtain a positive or negative result of the command from the test.

Defining methods

You can also define your own methods in your calculation script to utilise individually adjusted functions. In contrast to variables, they can also be used by other calculation script commands as soon as they have been executed at least once. The basic structure for the definition of methods is as follows:

def new_method
   (instruction block)
end

You can define variables, create loops, etc. within the instruction block. The designation (in this case, new_method) is the name of the method and must then be applied to the desired object in the calculation script with .new_method. The example code above will create a global method which is not linked to a specific class. To enhance an instance of a class with a new method, you have to add the the referenced variable to the syntax (def variable.neue_methode)

Image class

The Image class is different from the other classes intrinsic to vcwin in that it does not define or access geometry elements. Instead, this class enables access to the image contents of a certain image memory page and enables the user to evaluate image information, e.g. greyscale values, with his/her calculation script.

Since accessing image data via the calculation script is very computationally intensive, we recommend only accessing smaller sections of the image and not the entire image. For this reason, the number of pixel values has been limited to 1,000,000 per access.

Accessing image information in an image section is carried out using the createArea method:

image = VC::Image.createArea(-1, 200,200, 100,100)

Here, the image information of a square, 100 x 100 pixel image section is written to variable image at position 200, 200. The parentheses contain the following parameters: (image page number, x-position, y-position, x-stretch, y-stretch). If, as in the example, the image page number is set to -1 or if no parameter at all is used at this point, the calculation script accesses the current image page.

The image variable then contains the following information: Image format (greyscale value, RGB), width, height and an array with all greyscale or colour values of the pixels being viewed. These values can be queried using special attributes:

Information	Attribute	Value
Image format	.format	1 for greyscale images (constant VC::Image::Grey8), 3 for RGB images (constant VC::Image::Rgbx)
Width	.width	Width in pixels
Height	.height	Height in pixels
Greyscale value	.grey[y][x]	Greyscale value(s) at position [y] [x]
Colour value red	.red[y][x]	Colour value(s) red at position [y] [x]
Colour value green	.green[y][x]	Colour value(s) green at position [y] [x]
Colour value blue	.blue[y][x]	Colour value(s) blue at position [y] [x]

To run through several image sections in succession, as with the line scan camera, you should use a loop which accesses image sections step by step and processes the information further. Ensure, however, that you do not query image information until the loop is run through and that you do not reference them further, as otherwise the upper pixel limit will be reached after too many run-throughs and the command will be aborted. The following example shows how you could create a functional loop for querying image information:

```
(0..1000).step(1)do
    |y|
    image = VC::Image.createArea(-1,0,y,2500,1)
      for x in 0..2500
        ifimage.grey[0][x] > 80then
          return false
        end
      end
  end
return true
```

The first line contains a step loop which specifies a certain range (0..1000) run through in a certain number of steps (1). The affected variable is defined in the second line |y|. This is followed by the createArea method, which specifies an image section of 2,500:1 pixels at position 0, y. Since y is incremented from 0 to 1,000 by the loop, the entire image can be run through step by step. The if-then-else condition then queries the individual greyscale values in the image section and checks whether they are greater than 80. If this is the case for a greyscale value, the instruction block is exited and the command is terminated as having failed.

6.3.23 Evaluate String

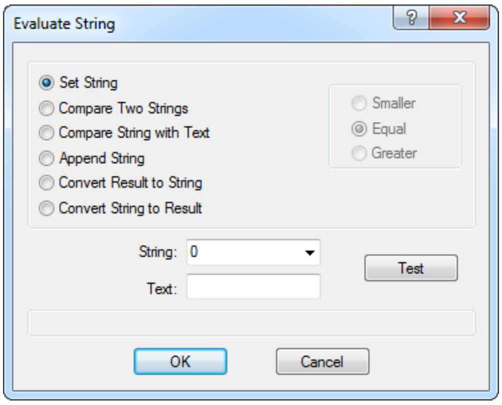


Image 169: Dialogue Evaluate String

String variables are variables for storing characters which have been read. Up to 20 string variables can be stored in the pictor or vicosys. The maximum length of the string variables is 54 characters.

Use **Evaluation > Evaluate String** to compare the strings previously read with one another or converted into numerical values for other uses. The following options are available:

The Evaluate String Option	Description
Set String	Setting a string variable with a text.
Compare Two Strings	Two strings are compared alphabetically with one another. The result of the comparison (smaller, equal or greater) is specified. The function is evaluated as unsuccessful if the result deviates.
Compare String with Text	The string is compared alphabetically with a defined text.
Append String	The first string is appended to the second string and saved.
Convert String to Result	A string of numerical characters is converted into a number. The number may be stored in a result variable for further evaluation.
Convert Result to String	The value of a result variable is converted into a string and stored in a string variable.

6.3.24 Angle

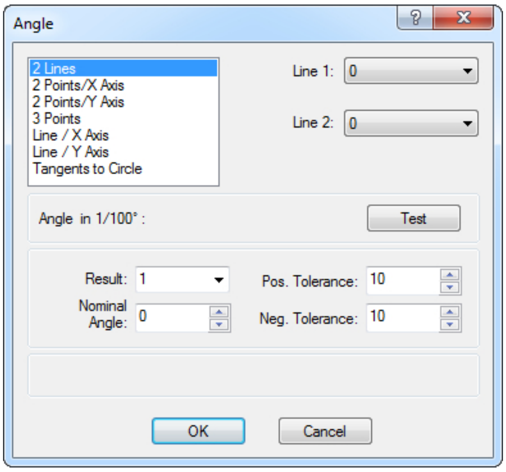


Image 170: Dialogue Angle

Write a command for determining the angle between geometrical variables into the test program using **Evaluation > Angle**. To do this, use any previously defined lines, circles, planes or even image coordinate axes as operands.

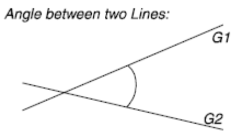
Selecting a Method

The following methods are possible:

- Angle between two lines
- Angle between two points and the X axis
- Angle between two points and the Y axis
- Angle between line and the X axis
- Angle between line and the Y axis
- Angle between three points
- Angle between two tangents from a point to a circle

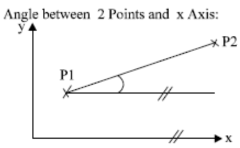
Angles are stored in the result structure with 1/100 of a degree precision. Tolerances can also be checked.

Angle between 2 Lines (-90° to 90°)



1. Enter the first line number.
2. Enter the second line number.
3. Enter the nominal angle and +/– tolerances.
4. Enter the result number.

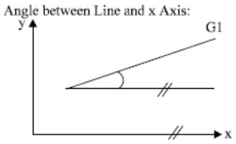
Angle between 2 Points and X or Y Axis (-180° to 180°)



1. Enter the first point number.
2. Enter the second point number.

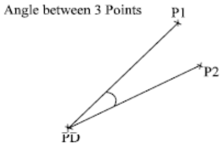
3. Enter the nominal angle and +/– tolerances.
4. Enter the result number.

Angle between a Line and X or Y Axis (-90° to 90°)



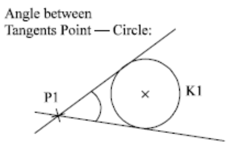
1. Enter the line number.
2. Enter the nominal angle and +/– tolerances.
3. Enter the result number.

Angle between 3 Points (-180° to 180°)



1. Enter the first point number.
2. Enter the second point number.
3. Enter the pivot point number (PD).
4. Enter the nominal angle and +/– tolerances.
5. Enter the result number.

Angle between Tangents Point - Circle



1. Enter the point number.
2. Enter the circle number.
3. Enter the nominal angle and +/– tolerances.
4. Enter the result number.

Evaluation Parameters

Evaluation	Description
Result	Result number for storing the calculated angle (in 1/100 of a degree).
Nominal Value and Tolerances	Nominal angle and pos./neg. tolerances allowed during the calculation. You can set suitable parameters interactively with the aid of the [Test] button.

Test

The test displays the calculated **angle in 1/100 of a degree**.

6.4 Control Commands

6.4.1 Asynchronous Processes

ADVICE
Use of the Asynchronous Processes feature is based on a licence model.

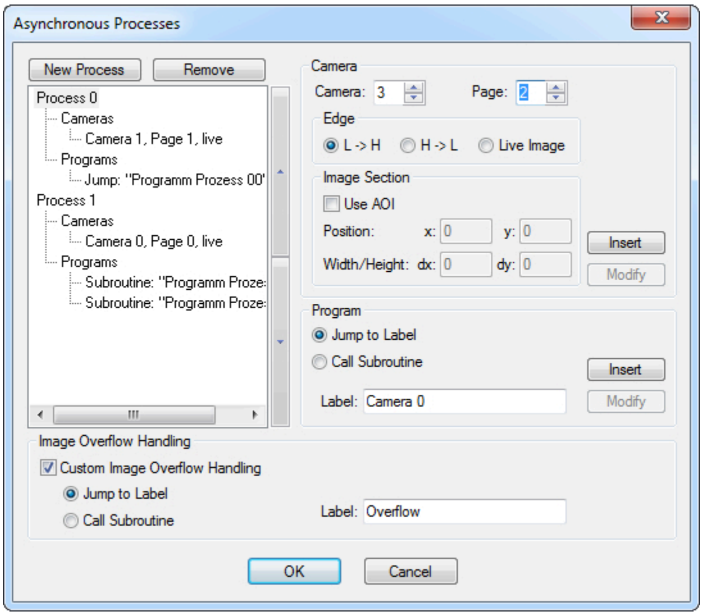


Image 171: Dialogue Asynchronous Processes

Control > Asynchronous Processes allows you to wait for new images from a camera, while other pictures of other cameras are also being processed.

Parameters Camera

ADVICE
Hint: The image acquisition is started by the digital input of the camera. This only applies to the edges Low to High and High to Low.

Parameter	Description
Camera	Designation of the camera.
Page	Designation of the image memory page, on which the image should be saved to.
Edge	Status or edge transition of the trigger input which releases the image acquisition. Possible states: Low -> High, High ->Low, Live Image (continuous image acquisition).

Parameters Image Section

Parameter	Description
Use AOI	Activate the checkbox if you want to make an image acquisition of a smaller, limited area (Region/Area of Interest).
Position	Starting point (x-, y- coordinates) of the rectangle which forms the AOI. The origin (0,0) of the image-coordinate system is located in the top left corner.
Width /Height	Edge lengths of the rectangle in relation to the starting point.

Parameters Program

ADVICE

Here, for every program choose separately, whether you want to jump to a subroutine or a label. When using labels, it is necessary to ensure that there is a jump-back to the command "Asynchronous Processes" after the processing.

Parameter	Description
Jump to Label	Jump to a label.
Call Subroutine	Jump to a subroutine.
Label:	Enter the name of the label here. Make sure that the label already exist in the program.

Parameters Image Overflow Handling

ADVICE

There is no time schedule for subroutines, so it is possible that a camera captures several images, when returning to the "Asynchronous Processes" command. In this case, an error treatment is necessary. You can configure the error treatment under "Image Overflow Handling".

The standard error treatment stops the vicosys and provides a warning notice with the error description „image overflow on camera xx“.

Parameter	Description
Custom Image Overflow Handling	Activate the checkbox if you want to aberrate from the standard error treatment.
Jump to Label	As soon as an image overflow is detected the program immediately jump to that label.
Call Subroutine	As soon as an image overflow is detected the program immediately calls this subroutine.
Label:	Enter the name of the label here. Make sure that the label already exist in the program.

Short description

Using "Asynchronous Processes" you can define different processes. Each process shall have to be assigned at least one camera with the appropriate image memory page and at least one program. The priority of the process depends on the position of the process. Process 0 has the highest priority.

The parameters of a process are handled synchronous. This means the process is waiting until until the acquisition of all captured images is completed and the images are available on the image memory pages of the vicosys. The processes, in relation to each other, run asynchronously.

The command "Asynchronous Processes" checks every process if a new image (using more than one camera in an single process, every image has to be new) exists. In this case the processing routine of the corresponding process is called.

Depending on the configuration of the process a jump to a label or a call of a subroutine will be executed. Within this block, the captured images are processed.

When using a subroutine, it is necessary to ensure that there is a "RETURN" at the end of the section. Automatically there is a jump to the next command behind the command "Asynchronous Processes".

When using a label, it is necessary to ensure that there is a jump directly behind or previous to the command "Asynchronous Processes".

ADVICE

Use a endless loop!

If all processes are checked and if applicable processed, the command "Asynchronous Processes" ends itself and returns to the main program. Therefore it is necessary to ensure that the command is embedded in an endless loop. This loop will be left only for the labels and subroutines.

ADVICE

Pay attention to the priorities of your processes!

For each cycle of "Asynchronous Processes", only one program will be processed. Even if there is more then one program in a process, only one of them will be processed.

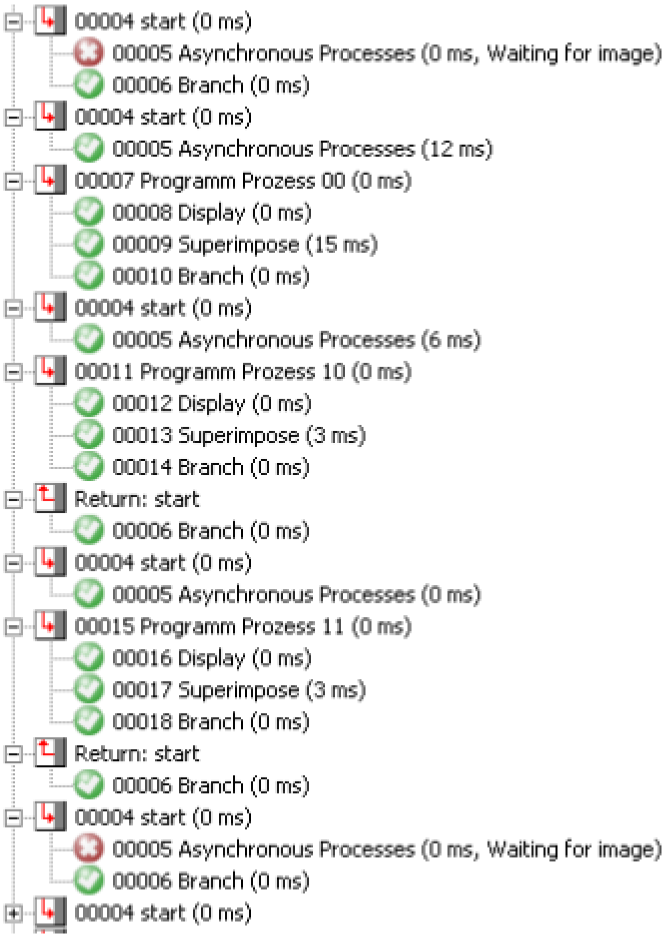
The processing of the other programs of the process take place the next cycles. In the meantime, if a process with a higher priority provides an image for processing, this process is preferred.

Short Overview "Asynchronous Processes"

The following program represents a short overview of the embedding of "Asynchronous Processes".

No.	Label	Command
******		Last Change at vcwin pro 2.14.131; Vision System: All commands
00000		Turn on Multiscreen Mode with Pages: 0, 1, 2, 3
00001		RES[0] = 0; Nominal Value 0 (+10/-10)
00002		Exposure Time Camera 0 80000 µs
00003		Exposure Time Camera 1 40000 µs
00004	start	
00005		Asynchronous Processes Cameras: 1 (1) Labels: Programm Process 00 Cameras: 0 (0) Labels: Programm Process 10, Programm Process 11
00006	-->	Jump to Label "start" (unconditional)
******		
******		
******		
00007	Programm Process 00	
00008		Image memory pages: Acquisition Page 1, Display 1, Work page 1, Demo Page 1
00009		Superimpose "Programm Process 00"; over Demo Page; Graphic Colour Text Size 7; Position [100, 100]
——		Wait 1500 ms
00010	-->	Jump to Label "start" (unconditional)
******		
******		
******		
00011	Programm Process 10	
00012		Image memory pages: Acquisition Page 0, Display 0, Work page 0, Demo Page 0
00013		Superimpose "Programm Process 10"; over Demo Page; Graphic Colour Text Size 3; Position [100, 100]
00014	<<<<	RETURN (Leave Subroutine)
00015	Programm Process 11	
00016		Image memory pages: Acquisition Page 0, Display 0, Work page 0, Demo Page 0
00017		Superimpose "Programm Process 11"; over Demo Page; Graphic Colour Text Size 3; Position [100, 250]
00018	<<<<	RETURN (Leave Subroutine)
******		
******		
******		
00019	Fehler	
00020		Image memory pages: Acquisition Page 2, Display 2, Work page 2, Demo Page 2
00021		Superimpose "Error"; over Demo Page; Graphic Colour Text Size 14; Position [100, 500]
00022	-->	Jump to Label "start" (unconditional)
******		

Using the Debug-Mode, the results are as follows:



6.4.2 Control Lighting

This command allows the user to change the parameters and settings of lighting from within a test program. The lighting must be controlled by a smart light-lighting controller (DLC) by Vision&Control.

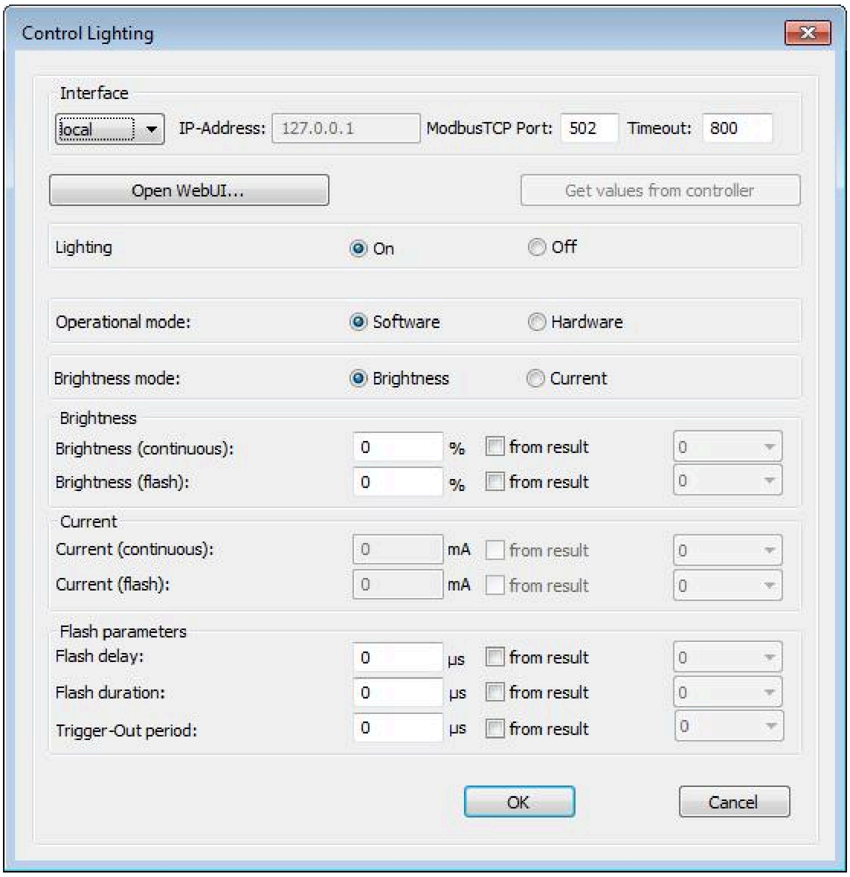


Image 172: Dialogue Control Lighting

Command options

1. Interface

In this section you can configure the connection to the DLC-server.

Parameter	Description
remote/local	Select "local" if you want to connect to a DLC-server on the local vision system. Select "remote" if you want to connect to a DLC-server on a remote system.
IP-Address	IP-address of the remote system running the DLC-server (Disabled if setting is "local").
ModbusTCP-Port	Port assigned to the DLC you want to control (*see "DLC-Server settings", page 92* or "DLC-Server Control Center").
Timeout	Timeout for the connection (in ms).

2. Open WebUI

With this button you can open a web browser, which connects to the DLC-WebUI of the chosen DLC-server. It uses the IP-address parameter (or the IP-address of the vision system if you selected "local"). The command always uses port 80 (port 591 if you selected "local"). If you configured your DLC-Webserver to use a different port (*see "DLC-Server settings",* *page 92* or "DLC-Server Control Center"), you have to change the port number in the address bar of the browser manually.

3. Get values from controller

Configure your DLC and lighting comfortably with the WebUI and then transfer the parameters into your test program by pressing this button. The current values of the DLC will be used as parameters for the command.

4. Lighting parameters

In this section you can set the parameters for the lighting.

Parameter	Description
Lighting on/off	Switches the lighting on/off.
Operational mode	**Hardware:** Configure the DLC with the rotary switches on the front panel. Parameters are LED-current and flash duration. **Software:** Configure the DLC by software parameters.
Brightness mode	**Brightness mode:** The brightness is set relatively to the brightness of the calibration lighting in percent. **Current mode:** The brightness is configured by setting the current to the lighting.
Brightness (continuous)	Brightness of the lighting during continuous operation. Brightness is relative to calibration lighting in percent.
Brightness (flash)	Brightness of the lighting during flash operation. Brightness is relative to calibration lighting in percent.
Current (continuous)	Current to the lighting during continuous operation (in mA).
Current (flash)	Current to the lighting during flash operation (in mA).
Flash delay	Delay betweensignal on TRIGGER-IN and flash (in µs).
Flash duration	Duration of the flash (in µs).
Trigger period	Trigger period is the minimum time (in µs), after which the lighting can be triggered again. It depends on the flash duration, the flash delay and the pulse load capacity (→ see lighting manual).

6.4.3 Copy Image into Buffer

Overview

The command **Copy Image into Buffer** is located in the Command Selection under **Control**.

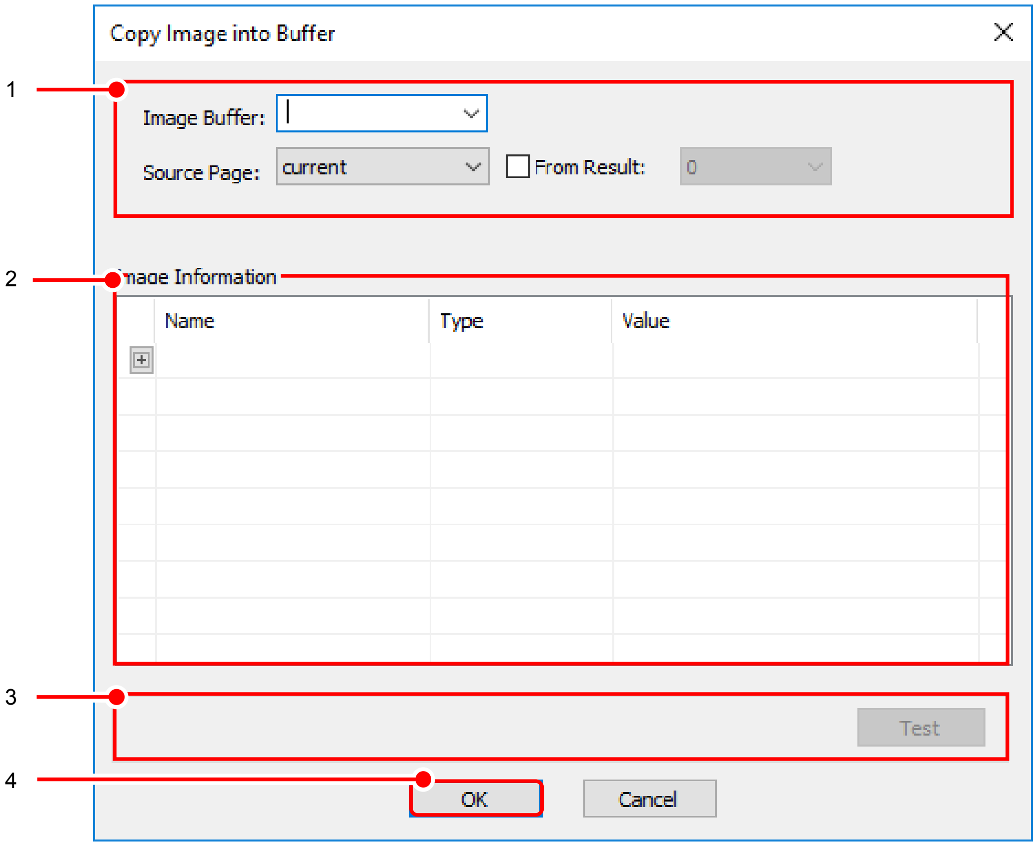


Image 173: Dialogue Copy Image into Buffer

Use this command to copy an image into an existing image buffer.

For more information, see *"Synchronized sending of images", page 398* and *"Initialize Image Buffer", page 342*.

Work flow

1. Define the image to be copied into the buffer.

Parameter	Description
Image Buffer	Select the buffer to which you want to copy the image
Source Page	Select the current image memory page from which you want to copy the image. If you want to use the current image memory page from a result, select *From result* and select the corresponding result from the drop-down list.

2. Specify the image information (metadata) here. These are copied with the image and can be retrieved from the WebHMI.

   Click the' +' button to insert a line. The button' -' deletes the respective line.

   Up to 16 entries are possible. The text of an entry (also from string) is limited to 64 characters.

   Name

- The name must be unique.
- All characters from "ISO/IEC 8859-15" (Latin-9, Western European) are allowed.

Type / Value

- Selection **Text**
  The text entered under **Value** is appended. All characters from "ISO/IEC 8859-15" (Latin-9, Western European) are allowed.
- Selection **Date/Time**
  The current system time (format: "YYYY-MM-DDTHH:mm:ss.sss") is appended. For custom time/date formats, use the command **System Time**.
- Selection **Result**
  The value of the specified result variable is appended.
- Selection **String**
  The value of the specified string variable is appended.

TIP

If the result or string for an image information is not defined, the image will still be stored in the buffer. The image information has an empty string as value.

3. Use the [Test] button to check the previous configurations of the dialogue.

Error messages

- Variable not defined
  Variables were not defined. The image is still copied into the buffer.
  If the variable for a page number from result is not defined, *"Source Page not available"* appears, in that case the image is not taken over.
- Image buffer not available
  Check the corresponding **Initialize Image Buffer** command, if the buffer has been configured for the camera from which the image was taken.
- Image in source page not from same camera as buffer is configured for.
  Check whether the buffer of the camera from which the image was taken is initialized.
- Source page not available
  The number of the source page is larger than the existing image memory pages or the result for the source page is not defined.
- Web Server not active
  The Web Server was not activated or no **Initialize Image Buffer** was called.

4. Use the [OK] button to insert the command into the test program.

6.4.4 Send Image

Use **Control > Send Image** to insert a command for sending images of the current or free electable image memory page from the vision system to the host computer or to automation equipment.

The images sent can also be displayed in the vcwin Measurement User Window.

You can also receive, further process or store the images with an additional external program. The protocol has been published and is in the format vicorem.

Use this command to send all images or only error images to the host or to display error images while the process is running.

The images sent to the host are saved in C:\temp as standard. The directory to be used can be changed under **Preferences** in the Measurement User Window.

Sending of data can take place, according to the connection, via the serial or Ethernet interface.

ADVICE

Saving images of pictor or vicosys can take place only by means of the measuring user window of vcwin or by a separately programmed tool. With the multi-camera system vicosys the user has additionally the possibility of saving images on an external storage device.

Brief Guide

To display an image in the top left of the Measurement User Window:

1. Include the **Send Image** command in the test program.
2. Transfer the program to the vision system using **Communication > Transfer > Load as File**.
3. Select **Options > User Window Selection > Measurement User Window**.
4. Restart **vcwin**.
5. Select **Preferences** in the Measurement User Window, then select the **Display** option.

Input Parameters

Condition for Sending
Sending can be linked to the result of the last command or command block.

- Unconditional
- After successful Command
- After failed Command
- After successful Block
- After failed Block

Image Contents	Description
Image Section Only	Only sends the image section produced in the Teach-in window. The transfer time is hence reduced due the smaller size of the image section.
With Graphic	The graphic information superimposed on the monitor (overlay) is included in the current image information. This option should be used, for example, for demonstration purposes to send the overlay graphics at the same time as the image. Note: Transferred images cannot be used anymore for image processing tests if you use the **With Graphic** option.
Send Current Page	When the **Send Current Page** option is used, the image page defined as the current work page in the system is selected. If **Send Current Page** is not selected, then the number of the image page must be given.

Image Format
Images in vcwin (R) can be saved in the JPG or BMP formats. For JPEGs, the compression can be set.

Synchronization
Before sending image data, the vision system sends a data block with image information (position, size and send format).

Options

For sending images there are 2 further options. (for pictor only)

- Send Quick
- In the background

Send Quick

With the option "Send Quick" image data are send without any protocol to the communication partner. This leads to an increase of transfer rate.

In the Background

With the option "In the Background" the image is saved on an internal memory page as ring buffer. If the pictor waits for external events like data transfer, I/Os, flash commands, waits or process module the image is sent block-by-block in the background. There is no jitter by using Ethernet. Using the RS232 interface you can get a jitter of 10 ms.

With the help of this option defect images can be sent without affecting or slowing down the process. The transmitting time can extend over several test cycles. The time for controlling one object can be thus smaller than the time for sending an image.

The measurement user window can be used with this option.

Numbers of the background pages

System	Numbers of pages
M16xx	2
M18xx	8 (but M1821E = 5)

Numbers of background pages

ADVICE

If all background pages with image data are occupied, the current image will be rejected.

Interface

Interface	Description
Select one of the available data interfaces, to which the result is to be sent. The interfaces that are available depend on the type of vision system used.	
/com	Serial interface
/ethernet	Ethernet interface
In order to guarantee proper functioning, do not change the entries behind /com and /ethernet unless you are a network technician.	

When using the **serial** interface, you can try to increase the transfer speed using **Communication > Interface** by setting the baud rate to a maximum of 115200.

- In order to preserve the increased speed during the next start, execute **System Preferences > Save File System In Flash**.
- In order to change the image transmission speed for pictor in the program, enter this as a remote control code into the test program (e.g. for conversion to 115200 baud: E9 01 02).
- In order to obtain the image transmission speed for vicosys , configure the interface setting via the parameters in the dialog box:
  Example: Interface: /com Port: 1 Parameter: 115200,n,8,1;rtson
  Possible settings: Baud rate … 115200 (9600 / 19200 / 38400/ 57600)
  Parity … n/e/o
  Data bits … 8 / 7
  Stop bits … 1 / 2
  RTS/CTS … rtson/rtsoff

To view an image in the Measurement User Window when using an **Ethernet** connection, the vision system must operate as a TCP server with port. As this corresponds to the presets in **Communication > Interface**, the fields for Port and IP address must be left free.

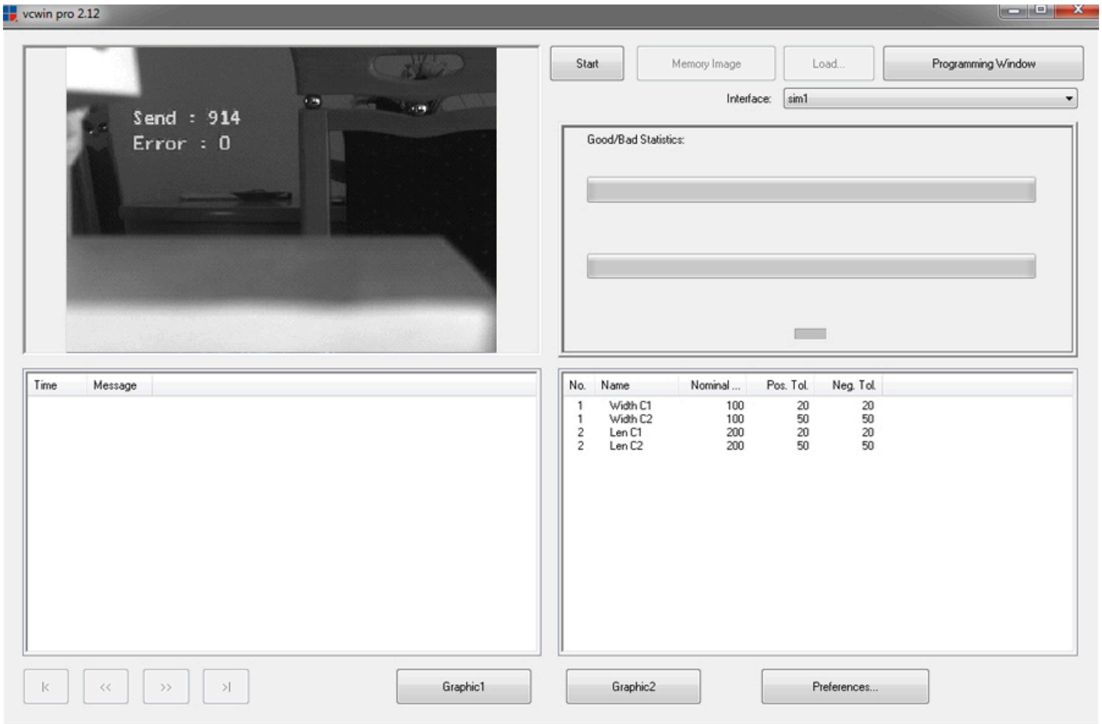
Background:

Interface	Meaning for Ethernet connection
/ethernet without port and without IP	All vision systems are TCP servers on port 8500.
/ethernet with port and IP	All vision systems are TCP clients on the set port.

Wait For Reception Readiness … When this option is selected, vcwin waits to receive confirmation of readiness to receive from the host computer before sending. If this is not received within the Timeout period, the vcwin command is aborted. If this option is not selected, vcwin sends the image data immediately after the send command.

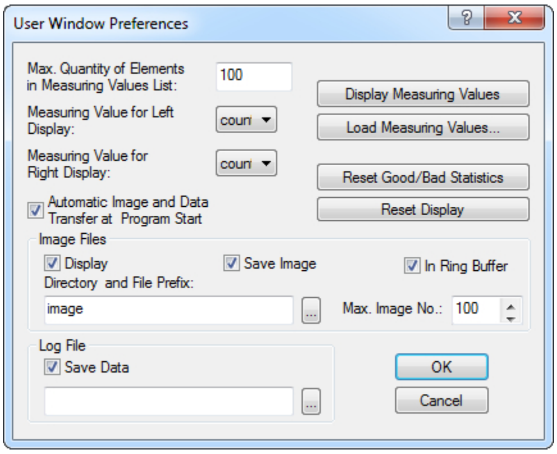
Receive and Display Images with vcwin

The images sent can be received, displayed and saved from within the vcwin Measurement User Window.



Parameterize Data Received in vcwin

Open the parameter dialog window in the Measurement User Window with the [Preferences] button. Use the following settings under the **Image Files** heading:



Parameters	Significance
Save Image	The image is saved in the file directory given. The file name contains the prefix given as well as the date received and the clock time. The image format (JPEG or BMP) cannot be selected in this dialog window - it is set by sending.
Display	vcwin displays the image received in the above left section of the Measurement User Window.

Example of the File Name for a Sent Image:
The file name is constructed from the prefix **B1** as follows: **B1-070528-142115.bmp** — Explanation: The image was received on the **May, 28th 2007** at **14:21:15**. If several images are received within a second, a consecutive numbering of the images results: B1-070528-142115-**2**.bmp.

6.4.5 Save / Load Image

Overview

The command **Save / Load Image** is located in the Command Selection under **Control**.

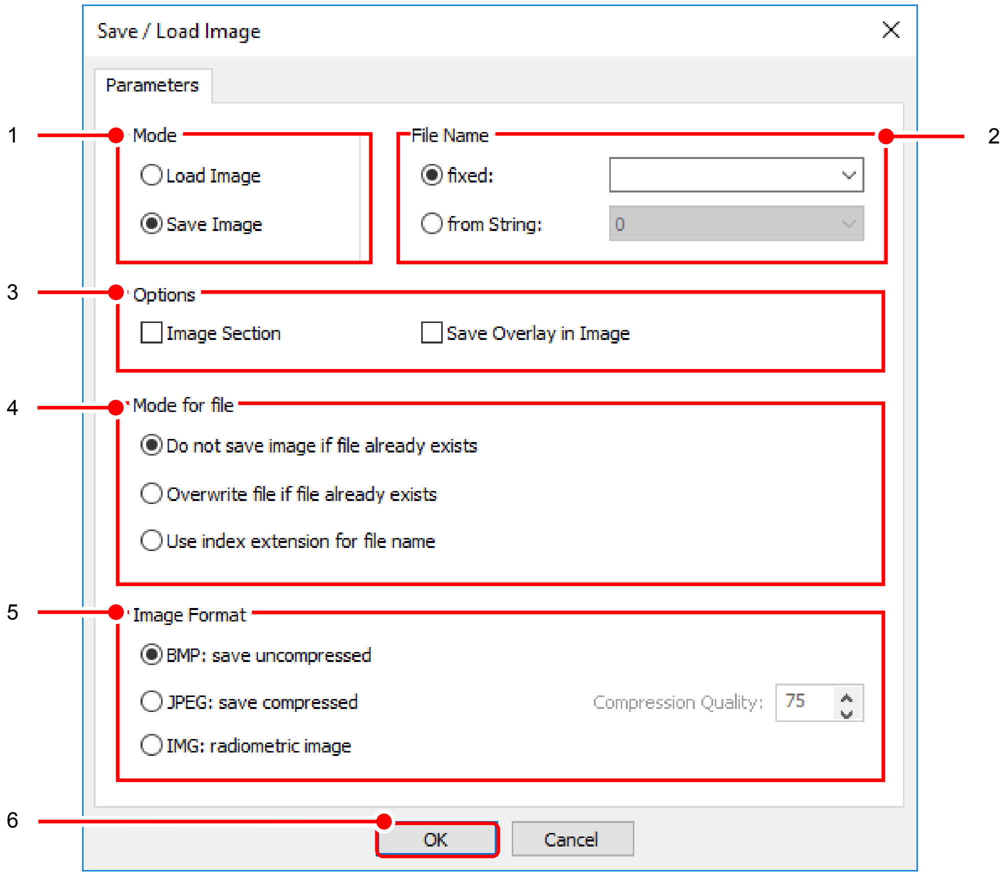


Image 174: Dialog Save / Load Image

Use this command to save or load images from or in the current image page.

The image is saved in the image directory. The image directory ist located on the vision system or on the external storage device selected using **Control > External Storage Device**.

With **System Preferences > System Resources / Initial Program** you can check which images are stored on the image directory.

Work flow Load Image

Use this option to load an image from the image directory to an image page (edit page).

1. Activate the mode *Load Image.*
2. Enter the name of the image file.

Parameter	Description
fixed	Select the name of the image file from the drop-down list. If the image is not in the list, enter the full name (image name and the file extension .bmp / .jpg). The image must be present in the images directory at the commands execution time.
from String	Select the name or number of the string from the drop-down list. The string must be present in the geometry list. The content of the string must contain the image name with the extension .bmp / .jpg. Note that, for an error-free loading, the content of the string must conform to the naming conventions of the respective vision system.

3. (6) Use the [OK] button to insert the command into the test program.

Work flow Save Image

Use this option to save an image in the image directory.

1. Activate the mode *Save Image*.
2. Enter the name of the image file.

Parameter	Description
fixed	Enter the name of the image file. The image is saved in the images directory with that name and the extension of the selected image format (.bmp / .jpg).
from String	Select the name or number of the string from the drop-down list. The string must be created in the geometry list. The image is saved in the images directory with the name which is in the string and the extension of the selected image format (.bmp / .jpg). Note that, for an error-free saving, the content of the string must conform to the naming conventions of the respective vision system.

3. Enter additional storage options.

Parameter	Description
Image Section	Enable this option if you want to save only a section of the image. When enabled, the dialog expands to the tab control **Teach-in**. Switch to the tab control and adjust the image section here (*see "Teaching in Test, Detection and Search windows", page 26*).
Save Overlay in Image	Enable this option if you want to save the overlay. The Overlay is written fixed in the image.

4. Enter here how the vision system should react if the specified file already exists.

Parameter	Description
Do not save image if file already exists	The image is not saved. A file system error is displayed.
Overwrite file if file already exists	The already existing file will be overwritten.
Use index extension for file name	With this option, the images are saved with the file name and a consecutive number (index extension).

5. Select the image format.

Parameter	Description
BMP	The image is saved in .bmp format without quality loss. Use this option if you want to apply inspection algorithms on the image.
JPEG	The image is lossy compressed and saved as a .jpeg. This allows smaller image files and faster image transfer, but at the expense of image quality. It is therefore recommended to leave this option disabled, because the image is changed in its gray values by a compression. Also for color images, individual color values do not match with the original image color.
IMG	Allows to store a thermographic image with all radiometric information. The image is stored in the FLIR FILE FORMAT and can be reopened with FLIR software (e.g., FLIR tools, FLIR ResearchIR). Only in combination with a thermography license.
Compression Quality (only JPEG)	Enter the quality with which the JPEG images are stored. The value can be set in increments of one from 10 to 100.

ADVICE

Use BMP, when it comes to accurate traceability.

Use JPG, if it is only a rough visual inspection.

6. Use the [OK] button to insert the command into the test program.

6.4.6 Initialize Image Buffer

Overview

ADVICE

This command must appear at the beginning of the test program.

ADVICE

When the command is executed, all existing buffers and the images contained in them are deleted.

The command **Initialize Image Buffer** is located in the Command Selection under **Control**.

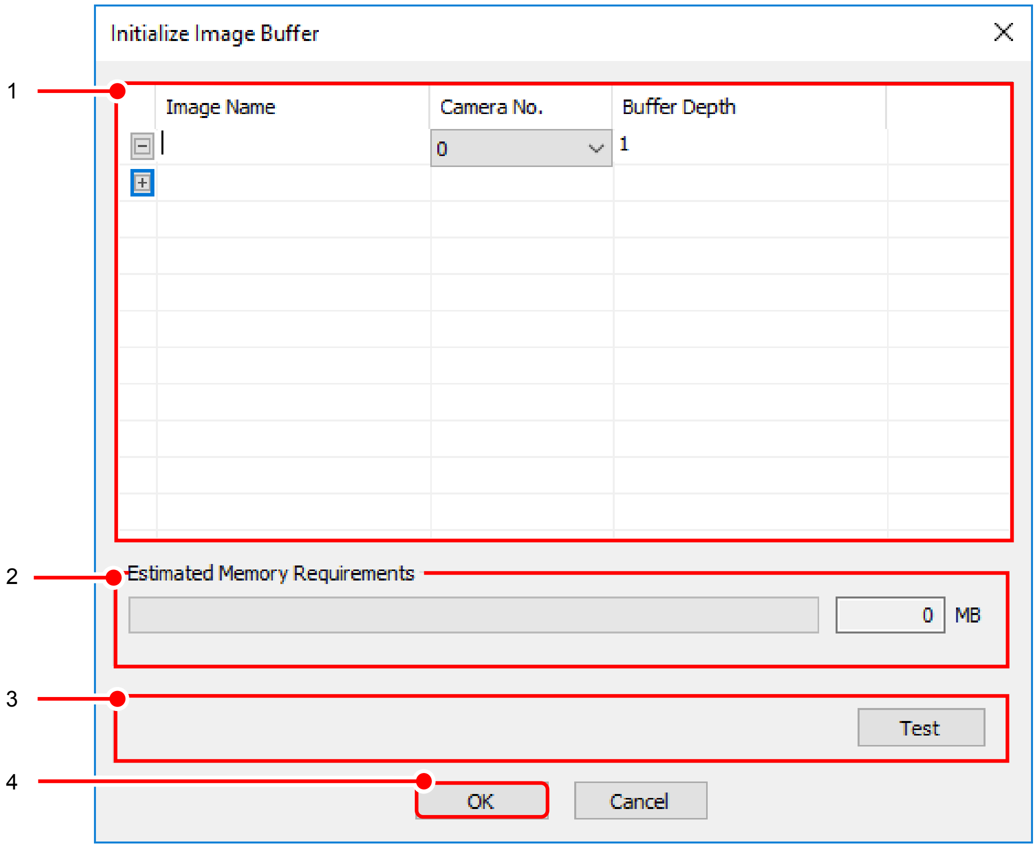


Image 175: Dialogue Initialize Image Buffer

Use this command to initialize all image buffers in which you want to save images for the web interface. The images stored in the buffers can be retrieved from the web server.

For more information, see *"Synchronized sending of images"* and *"Copy Image into Buffer"*.

Work flow

1. Click the' +' button to insert a line. Define here all image buffers to be used in the test program. The button' -' deletes the respective line.

Parameter	Description
Image Name	Enter the image name. The name must be unique. All characters from "ISO/IEC 8859-15" (Latin-9, Western European) are allowed.
Camera No.	Specify the camera whose images you want to store in the buffer. (Must be assigned to the camera from which the images to be buffered were taken.) • The camera determines the resolution and colour depth for the image buffers. • It is not possible to copy images with a different resolution into the buffer. • Colour images cannot be copied to the buffer of a monochrome (black/white) camera. • Black/white images can be copied into the buffer of a colour camera. (e.g.: result after a colour conversion or colour binarization)
Buffer Depth	Specify the number of images that should later be preserved as history in the buffer.

2. If there is a connection to the vision system, the memory consumption is automatically displayed here. When memory consumption is displayed in red, the buffer allocated to the web server is not sufficient for the buffer configuration. Increase the allocated memory in the **System Preferences > Web Server Settings** menu or decrease the buffer depth.
3. Use the [Test] button to check the previous configurations of the dialogue.

   Error messages

   ° not enough memory
   Too many image buffers with too much depth have been configured. The allocated memory is not sufficient.
   ° Camera not available
   A camera that is not connected to the vision system has been assigned to a buffer.
   ° Web Server not active
   The Web Server is not activated or has not been started.
4. Use the [OK] button to insert the command into the test program.

6.4.7 Data Transfer

Overview

The command **Data Transfer** is in the Command Selection under **Control**.

Usage

- Communication of the vision system with machines
- Sending and receiving elements
- Changing the sequences of the test program
- Configuring target sizes
- Generating statistics

Characteristics

- Several protocols
- Several elements per command
- Several data formats per command

Dialogue structure

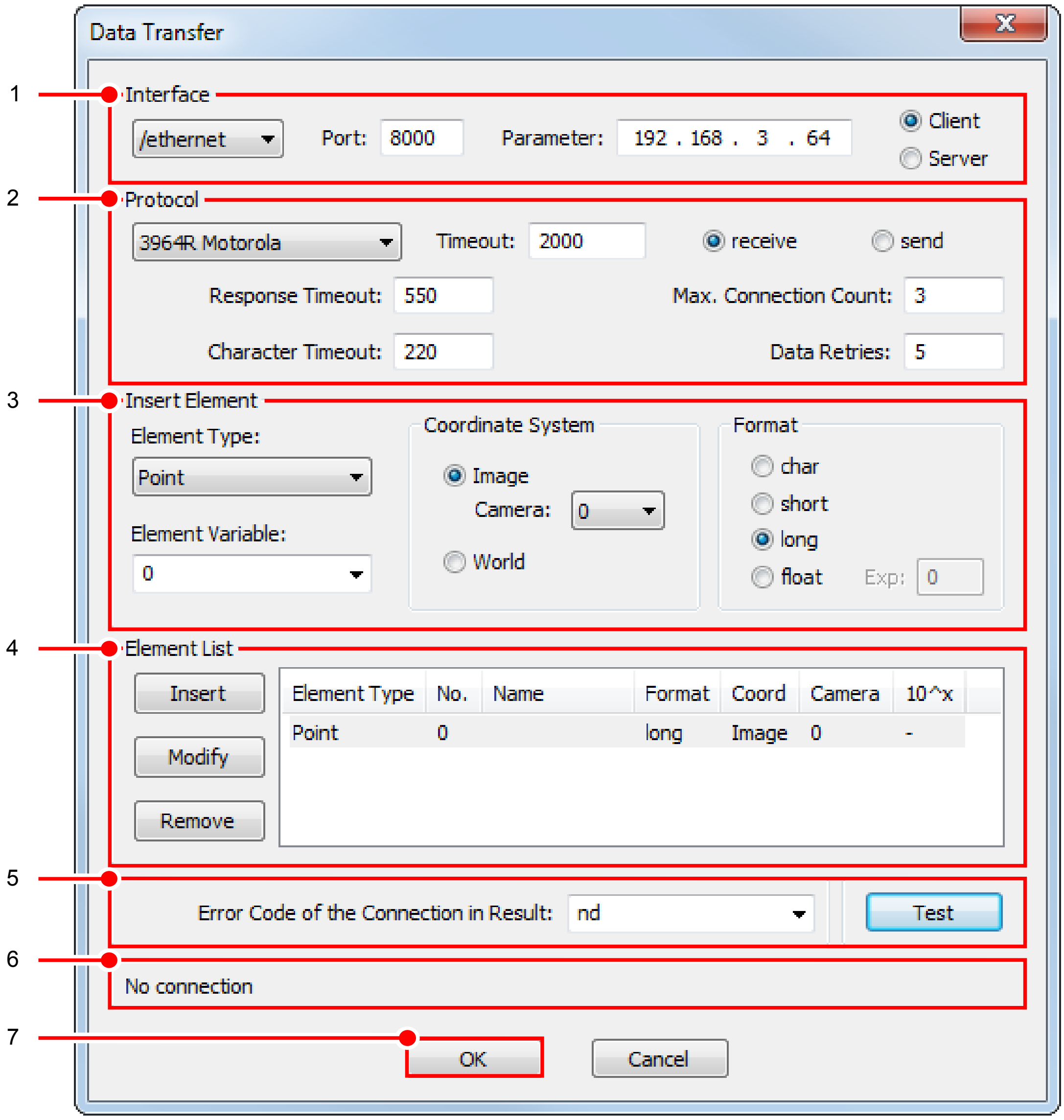


Image 176: Dialogue Data Transfer

Work Flow Parametrize Interface

1. Parametrize the interface

   Select the interface to be used */com* (serial) or */ethernet* from the drop-down list.

Serial Interface

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>/com</td><td>The serial interface /com is used.</td></tr>
<tr><td>Port</td><td>Port 1 is always used. Enter 1 to configure the interface.</td></tr>
<tr><td rowspan="2">Parameter</td><td>For the devices pictor T and vicosys following values can be set.
• Empty
° The values last used are assumed.
• 9600,n,8,1;rtson
° Baud rate: 9600, 19200, 38400, 57600, 115200
° Parity: n, e, o (none, even, odd)
° Data bits: 8 or 7
° Stop bits: 1 or 2
° Flow control (optional): rtson, rtsoff (rts/cts on, rts/cts off)</td></tr>
<tr><td>For the devices pictor M following values can be set.
• Empty
° The values last used are assumed.
• 9600
° Baud rate: 9600, 19200, 38400, 57600, 115200</td></tr>
</table>

Ethernet Interface:

ADVICE

If the vision system is used as a server, the server port is not opened with the start of the program, but only when the Data Transfer command is executed the first time.

Parameter	Description
/ethernet	The Ethernet interface is used.
Client / Server	Client: Setting up a connection to the IP port of the server. Server: Start a server on the specified port of the device.
Port	Device as Client: Port of the server to be connected to. Device as Server: Server port on the vision system.
Parameter	Device as Client: IP address of the server. Device as Server: Value is greyed out and will be ignored.

Available protocols

Robot protocols:

- Schunk
- ASCII
- Epson Basic

Protected protocols:

- 3964R Motorola: High-Byte first (Big-Endian)
- 3964R Intel: Low-Byte first (Little-Endian)
- HEX: remote control protocol vicorem with block header and checksum

Simple protocols:

- RAW Motorola: High-Byte first (Big-Endian), without block header and checksum
- RAW Intel: Low-Byte first (Little-Endian), without block header and checksum
- ASCII: Values as ASCII characters, separator and end of block

Work Flow Parametrize Protocol

2. Define the protocol

   Depending on the choice, the following parameters can be set.

Parameter	Description
for all protocols	
Timeout	If communication is not established within this time, the communication attempt is aborted by the vision system. A running server still persists after the timeout. The value is in milliseconds. • Use a short timeout (1 ms), if the image processing system is only to be activated sporadically. • Use a long or infinite timeout (timeout = 0), to reliably exchange data or when waiting for the readiness of another system.
receive / send	Specify whether you want to receive or send the data.
3964R (Motorola), 3964R (Intel), Epson Basic	
Response Timeout	Maximum waiting time of the transmitter for an answer to the call of a connection or the checksum. If exceeding the value the connection is terminated.
Character Timeout	Maximum time interval between the individual characters. An exceeding is classified as a transmission end.
Max. Connections Count	Maximum number of connection attempts before communication attempt is aborted.
Data Retries	Maximum number of transmission attempts before transfer is cancelled.
HEX (remote control protocol vicorem)	
Block ID	Indication of vicorem blocks.
ASCII	
Separator	Character to separate values.
Block End (ASCII Codes)	Terminating character for all values of a command. By default 0d 0a (line feed) is used.

Work Flow Configure Elements

3. Choose here the appropriate element and configure it.

Parameter	Description
Element Type	Enter the type here: Point, Line, Circle, Contour, Result, String or Counter. For option send: additional types Text and Byte List.
Element Variable	Select the variable from the geometry or result structure.
Coordinate System	Select here the appropriate coordinate system (image / world). For option "receive" and Coordinate System "Image": For elements of the type Point, Circle, Line or Contour enter a camera number. The calibration values of this camera are used for a possible later conversion into world coordinates.
Format	Enter the formate of the element: • char: 1 byte • short: 2 byte • long: 4 byte • float: 4 byte, floating-point In the float format, enter the exponent for the scaling factor of 10^x. The transmitted / received result is multiplied by this scaling factor. For example, 1234 has to be send as 1.234: exponent -3 The received values of the type Float are rounded to an integer.

ADVICE

For Results, Point, Circles and Lines, each individual component is sent in the format selected. For Contours, each component is send in the format specified, but the number is always send in short.

4. Element List

   In this area, all elements to be received or send are listed.

   ° With [Insert] you add an element from the area *Insert Element* to the Element List.
   ° With [Modify], you can configure an element from the Element List. To do this, select the element and change its values in the area *Insert Element*. Now apply the changed values by clicking the [Modify] button.
   ° With [Remove] you remove an element from the Element List. To do this, select the item and click the [Remove] button.

Work-Flow Testing and Evaluating

5. Test and Result area

   Here you can specify a result number / name to store the connection error codes returned by the command. Error codes are saved in the result structure.

   ADVICE

   The command does not stop when sending not defined geometry elements. The command is executed and sends the following values:

   for char: 0x80

   for short: 0x8000

   for long: 0x80000000

   for float: 0.000000

   Test the command with the [Test] button.

6. Evaluation Area

   Here you can see the good/bad evaluation of the command, notes for parameterization as well as error messages.

7. Use the [OK] button to insert the command into the test program.

6.4.8 Enable/disable Demo Mode

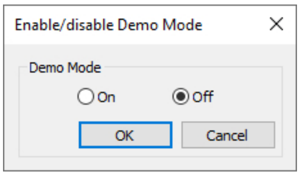


Image 177: Enable/disable Demo Mode

With **Control > Enable/disable Demo Mode** you write a command into the test program, which switches the demo mode (display of graphical elements) on or off. *see "Overlay Graphic", page 122*

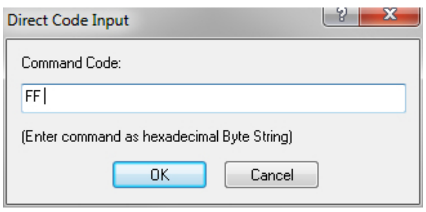
6.4.9 Direct Code Input

ADVICE

This function should only be called by experienced users.

ADVICE

All the command codes are described in the vicorem reference manual. However, it is possible that your vision system cannot interpret all the commands. Please also refer to the hardware manual that came with your vision system.



Use the **Control > Direct Code Input** command to call directly vicorem commands that the vision system supports. So you are able to call options, which are not available in vcwin dialogs. Also, you can use this to call new commands (created f.e. with vicosys SDK) that have no dialogs yet.

6.4.10 Save Settings

ADVICE
This command corresponds to the Direct Code Input F9.

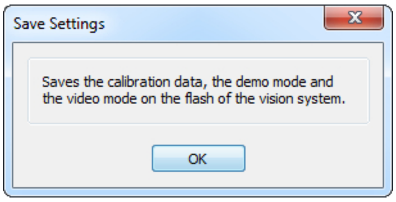


Image 178: Dialogue Save Settings

With **Control > Save Settings** you write a command into the test program that saves the calibration data, the demomode and the video mode on the flash of the vision system.

6.4.11 External Storage Device

Overview

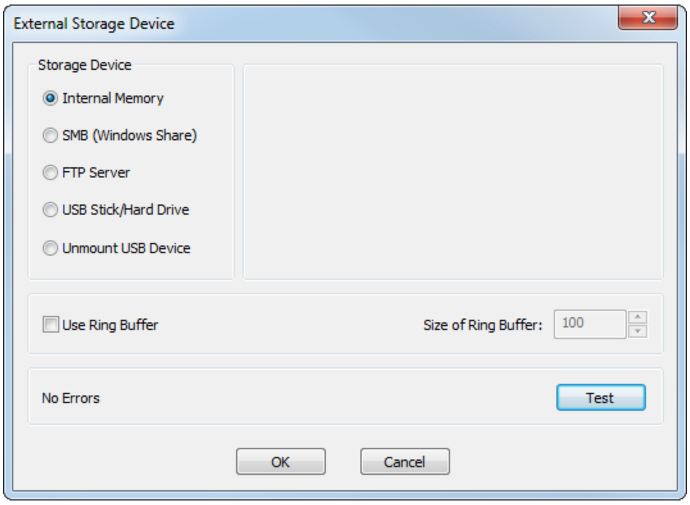


Image 179: Dialogue External Storage Device

With **Control > External Storage Device**, you define the storage device on which images are stored.

The optionally switchable ring buffer avoids overflow in the storage device. Data from the external storage device can be inspected and, if required, deleted by use of **System Preferences > System Resources / Initial Program** Area **Images**.

Selectable storage devices

- Internal Memory (pre-setting)
- SMB (Windows Share, Ethernet) for pictor T and vicosys
- FTP Server (Ethernet) for pictor M, pictor T and vicosys
- USB Stick/Hard Drive of vicosys(at vicosys 4xxx: also CompactFlash-Slot)

ADVICE

The **FTP Server** and **SMB (Windows Share)** options permit data to be viewed or processed further during the run operation.

Optimizing Speed

In order to achieve maximum memory speed, use USB hard drives on a vicosys.

If a network storage device is used, it should be noted that the network topology has an effect on the execution time of save and load commands.

Setting Global Parameters

Parameter	Description
Use Ring Buffer	If this option is activated, as many images can be saved simultaneously as the size of the ring buffer permits.

- The file names of the images saved have consecutive numbers.
- The ring buffer advances one number for each image saved.
- After the ring buffer is full, further storage starts again using the file name of the first image.
- Images with the same name are overwritten, independently of the settings in the **Save Image** command.

ADVICE

The ring buffer counter resets itself with every call of the **External Storage Device** command to the value 1.

Internal Memory

This option is pre-set. Select this option to save images to the internal memory.

Characteristics

- Storing images on the internal memory is slower than storing on other media.
- The storage time is heavily dependent on the image size.
- Saving the image is directly synchronous, i.e. saving of the image is completely finished after execution of the command.

SMB (Windows Share)

All Microsoft operating systems with extended support are supported.

Select this option to save images on the hard disk of a SMB server. The storage time depends on the image size, network speed and performance of the server.

ADVICE

Take note of upper / lower case with identification of subdirectories.

Parameter	Description
Server	DNS name or IP address of the SMB server. In order to be able to correctly resolve the server DNS name, the vision system must be informed of a server name via DHCP. Also see DHCP in the section regarding changing the IP address >> *"Change the IP Address of the Vision System", page 88*.
Directory	Name of the subdirectory (incl. share) on the SMB server (Windows Share). Subdirectories must be given in the form subdirectory 1/ subdirectory 2.
User name / Password	Login name and password of the SMB server.

FTP Server

Select the FTP Server option to save images on the hard disk of a FTP server. The storage time depends on the image size, network speed and performance of the server.

ADVICE

Take note of upper / lower case with identification of subdirectories.

ADVICE

Note that when connecting with External Storage Device FTP to a server, the server must not stop or be interrupted during operation. The functionality of the vision system can thereby be endangered.

Parameter	Description
Server	IP address of the FTP server.
Directory	Name of the subdirectory (incl. share) on the FTP server. Subdirectories must be given in the form subdirectory 1/ subdirectory 2.
User name / Password	Login name and password of the FTP server.

USB Stick/Hard Drive

ADVICE

To prevent data loss, unmount USB sticks/hard drives before shutting down or removal with the command **External Storage Device > Unmount USB Device**

Select the USB Stick/Hard Drive option to store images on the first partition of a USB stick or a USB hard disk (file system FAT 16 or FAT 32). The storage time depends on the image size and write speed of the USB stick/hard drive.

Parameter	Description
Directory	Name of the subdirectory on the USB stick/hard drive. Subdirectories must be given in the form subdirectory 1/ subdirectory 2.

Unmount USB Device

Select Unmount USB Device, in order to deregister a USB stick or a hard drive.

After unmounting the storage device, the program saves no more images until execution of the next **External Storage Device** command.

6.4.12 Copy Geometry Variables

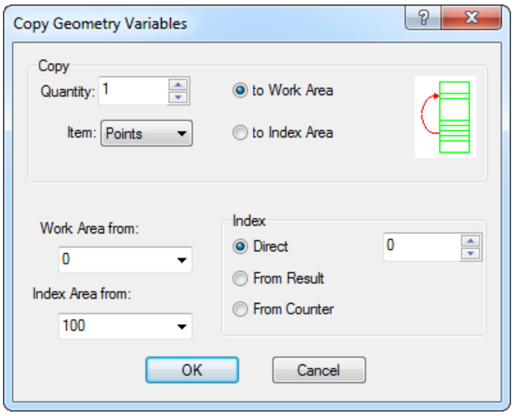


Image 180: Dialogue External Storage Device

Use **Control > Copy Geometry Variables** as a complex control command for time-saving access to geometry variables and results.

Copy groups of geometry variables or results backwards or forwards between the work area and index area. Use this command in combination with commands which create a large number of similar geometry variables or results (e.g. blob analysis, transition at line), for each of which a separate subroutine must normally be written.

Index Area and Work Area

The Index and Work Areas are not specific memory areas, but rather areas that can be individually defined for the groups of elements points, lines, circles, results and contours. Elements should always be changed in the Work Area, not in the Index Area.

Loop Control

Programs only execute in the **work area**. By generating a loop, one can load the geometry variables from the **index area** into the work area one after the other for processing or vice versa. In this way, a single evaluation program can be used for all similar geometry variables and results. (See also the example at the end of this topic.)

Input Parameters

Copy	Description
Quantity	Quantity of variables to be copied.
Item	Category of variables to be copied: points, lines, circles, results, contours.
to Work Area	Copy takes place from the index area to the work area.
to Index Area	Copy takes place from the work area to the index area.

The index defines which geometry element is to be copied next to the “index area”/“work area”. In so doing, an increment (integer) is added to the number of previously copied elements.

Index	Description
Direct	Increment is entered directly.
From Result	Increment is read from the result no. to be entered.
From Counter	Increment is read from the counter no. to be entered.

Areas	Description
Work Area from	First number in the work area of the geometry structure
Index Area from	First number in the index area of the geometry structure

Examples

Example 1: Using the Blob Analysis Command

Data are copied out of the index area into the work area.

1. Store the area center points which have been found by the **Blob Analysis** command (e.g. 10 points) one after the other in the point structure.
2. Set the number of the first point stored in point 1 as the start of the index area.
3. Set a free number as the start of the work area. The index area and work area must not overlap.
4. Set the number of variables “1” to be copied, type of variable **point** and copying direction **to work area**.
5. Select index **From Result**.
6. Interleave with the **Combine Results** command and execute program control commands.

Example 2: See the demo program **geocopy.vc**. >>*"Demo Programs", page 429*

6.4.13 Indexed Branch

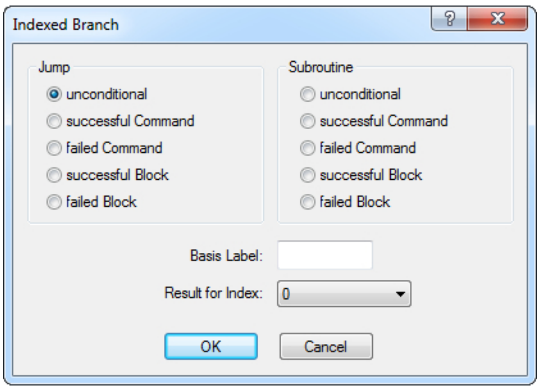


Image 181: Dialogue Indexed Branch

Insert several branches (jumps, subroutines) into the test program using **Control > Indexed Branch**. In this, the command jumps to a label whose name is composed on the fly from a fixed part – the basis label – and a variable part which is loaded from a result no. of the index. The advantage with this is that various subroutines or labels can be called with one and the same command, dependent upon the index.

For better clarity, the command is colored blue by default. You can change the colour using **Options -> application settings -> Font** *see "Application Settings", page 102*

Indexed branch commands always refer to the command or command block previously executed in the test program. In other words – indexed branch commands are inserted into the test program after the command or command block to be evaluated. Depending on the result of the previous command or command block the **Indexed Branch** command functions as follows:

Result of the previous command / command block	The program branch is implemented (it is otherwise ignored), if one of the following branch conditions is fulfilled …
Successful	• Successful command / successful command block or • Unconditional
Failed	• Failed command / failed block or • Unconditional

If the result of the previous command does not fulfill the branch conditions then the branch does not take place and the test program proceeds to process the next command.

Command Types and Branching Conditions

Command Types

The following branch operations are possible:

- Set labels.
- Set the start of a condition block.
- Set the end of the main program.
- Exit program.

Options for Branches	Description
Subroutine	The main program notes the start position behind the **Indexed Branch** command for the subsequent return from the subroutine to the main program. The subroutine must contain a return command.
Jumps	The program does not note the start position: used for jumping to labels.

Possible Branching Conditions

ADVICE

A conditional block counts as failed if a single command of the conditional blocks has failed, e.g. because a tolerance was exceeded.

Conditions used for determining whether to branch or execute the next command:

- successful command
- successful command block
- failed command
- failed command block

Input Parameters

Parameters	Description
Basis Label	First (fixed) part of the name of the label/subroutine which is to be jumped to.
Result for Index	Result number for saving the second (variable) part of the label name/ subroutine which is to be jumped to.

Example

See the demo program **verzweig.vc**. >> *"Demo Programs", page 429*

6.4.14 Calibrate

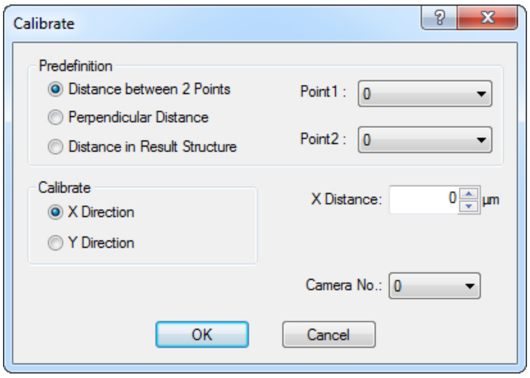


Image 182: Dialogue Calibrate

Using **Control > Calibrate** you add a command for a calibration program. The calibration program should be run every time, when cameras are changed on the vision system, the working distance changes or objective is replaced.

In order to carry out measurements with image sensors and other optical devices, a calibration factor (image scale) has to be ascertained i.e. how many pixels correspond to a given distance within the image. The calibration factor can be saved for further tests. Based on the features of image sensors, separate factors are determined for the X and Y directions. Calibration is carried out with a previously measured reference object.

ADVICE

You use calibration commands in special test programs. These should be written very carefully! You must make sure that the reference object is scanned precisely and that the data makes sense.

Calibration Factors

The calibration factor – the scale between image and world coordinates – is determined through a calibration command. Calibration is always active for the currently selected vision system.

The calibration factors are valid only until the system is restarted or a new calibration starts. They can be saved by using the command **Control > Save Settings** *see "Save Settings", page 350*.

Absolute and Relative Calibration

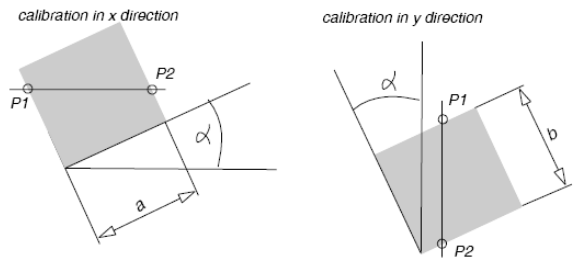
There are two basic types of calibration - absolute and relative.

Absolute calibration: Geometrical variables are defined in image coordinates.

Every relative calibration requires that the previous absolute calibration was successful. Geometrical variables are defined in world coordinates. The coordinate units used during relative calibration must be the same in the X and Y dimensions. The calibration factors are determined by comparing measured distances with the nominal distances from geometrical variables of the reference object.

ADVICE
You should make a point of carrying out relative calibration immediately after absolute calibration to improve the precision of tests.

To calibrate using distance between two points



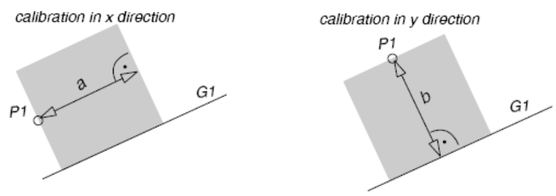
Geometrical elements in image coordinates (absolute calibration)

The calibration method **Distance between 2 Points** is a type of absolute calibration. The points must be defined in image coordinates. Any deviation in the position of the reference object from the X or Y image coordinate direction causes a significant calibration error. This can be compensated by adding either a **Perpendicular Distance** (in world coordinates) or a **Distance in Result Structure** calibration command.

Parameters

1. Enter the first point number.
2. Enter the second point number.
3. Enter the distance between the points in the X and Y directions (in world coordinates) in [µm].

To calibrate using perpendicular distance point-line



Geometrical elements in image coordinates (absolute calibration / relative calibration)

Absolute Calibration
Absolute calibration Perpendicular Distance (in image coordinates) requires square pixels because any change in the position of the reference object affects calibration.

Although the margin of error here is smaller than when calibrating with “Distance between two Points” (see above), the difference between the height and width of pixels during absolute calibration cannot be compensated for.

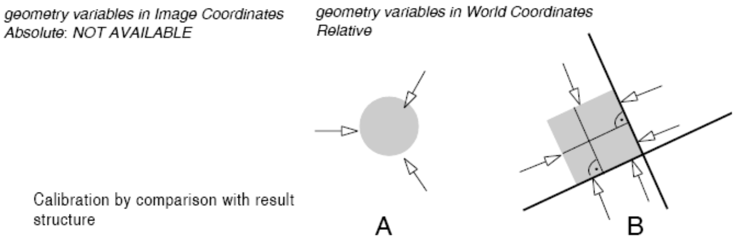
Relative Calibration
The relative calibration Perpendicular Distance (in world coordinates) must be preceded by an absolute Distance between two Points calibration. Any error due to changes in the position of the reference object can be compensated for.

Parameters

1. Enter the point number.
2. Enter the line number.
3. Enter the distance between the point and the line in world coordinates in µm.

To calibrate using the distance from the result structure

ADVICE
Relative calibration: the geometrical elements or distances (both X and Y) must be determined with all their calibration factors before the final calibration factors (X and Y) can be calculated. Calibration factors can be made more precise by grabbing images repeatedly and averaging out the distances.



Calibration compares relative distances within the result structure to known values (relative calibration). Distances can be measured several times, written into the result structure and the calibration factors adjusted accordingly.

The ratio between the X and Y scales must be known before you can use this type of calibration. This can be determined using e.g. the calibration command **Distance Point - Point**. >> *see "Distance", page 276*

Parameters

1. Enter the result number.
2. Enter the distance in world units in µm.

Example

See the demo program in the appendix **calib_x.vc**. >> *see "Demo Programs", page 429*

6.4.15 Line I/O

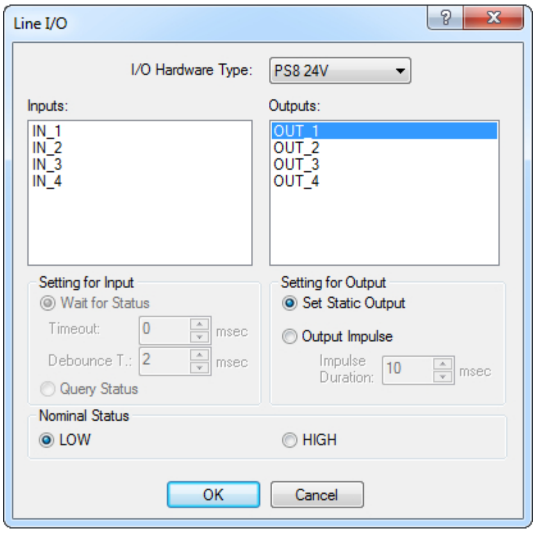


Image 183: Dialogue Line I/O

Write control commands for communication between the vision system and the process environment using **Control > Line I/O**.

Use this command for instance to read digital inputs from PLC or to set or reset outputs.

The current signals are compared to defined nominal states. Depending upon whether the condition is fulfilled, the command will return either **successful** or **unsuccessful**. Evaluate this result for a single command or for a condition block and can you use it to control branching.

>> *"Branch", page 367*

Test functions	Description
Wait	Wait until the required signal appears at the input.
Test Signals	Test for the required signal at the input.
Output Impulse	Send an impulse to the output.
Output	Send a defined signal to the output.

Input Parameters

Hardwaretype	Beschreibung
PS8	4/4-way input/output at pictor M ; 4 digital input, 4 digital output.
DIO4/6	4/6-way input/output port at pictor T-ECAN; 4 digital input, 6 digital output.
DIO4	4/4-way input/output port at pictor T-SC; 4 digital input, 4 digital output.
DIO8	8/8-way input/output port at vicosys; 8 digital input, 8 digital output.
DIO16	16/16-way input/output port atvicosys; 16 digital input, 16 digital output.

Parameters	Description
Inputs/Outputs	Input or output which needs testing/setting.
Query Status	If the **Query Status** option is activated, then the status at the input is queried immediately.
Set Static Output	If the **Set Static Output** option is activated, then the output is set to nominal status.
Output Impulse	If the **Output Impulse** option is activated, then an impulse with nominal status is output at the output set.
Impulse Duration	Duration of the impulse.
Nominal Status > Low/High	Status of the input/output to be queried / set.

Wait for Status

If the **Wait for Status** option is activated, the following parameters apply to the wait for a signal:

Wait for Status	Description
Timeout [ms]	Input is interrogated for the duration of the timeout.
Debounce [ms]	If the status of the input coincides with the nominal status, the program waits for the duration of the debounce time entered. If this input status still exists after the debounce time has expired, the result successful is returned.

Test Options for Inputs

Procedure for the I/O Test
The dialog box is the same for testing either type of I/O hardware. Available inputs and outputs are listed in one window each. Switch between both windows with the TAB key.

1. Select the line with the mouse.
2. Enter signal parameters to be queried (for inputs) or to be output (for outputs).
3. Press the [OK] button.
   The command is transferred into the test program.

Wait for Condition
The maximum time the test program will wait for a signal at an input (timeout) and how long the signal must remain at the port (impulse duration) can be defined with millisecond precision.

1. Select an input e.g. {IN_1}.
2. Activate the **Wait for Status** option.
3. Set timeout to the nominal value e.g. 1000.
   Note: If you set the waiting time to {0}, vcwin will carry on waiting until the condition is fulfilled (**No Timeout**).
4. The nominal dwell time (debounce time) is set internally at 10 ms and is not altered, even if a different value is entered into the dialog window.
5. Specify the state: **off** (LOW) or **on** (HIGH).

Testing Nominal Signal at Input
You can program checking a defined input signal as follows:

1. Select one of the inputs.
2. Activate the **Query Status** option.
3. Specify the state: **off** (LOW) or **on** (HIGH).

Output of Signals

Set Static Output
You can program the output of a defined output signal as follows:

1. Select one of the outputs, e.g. {OUT_3}.
2. Activate the **Set Static Output** option.
3. Specify the state: **off** (LOW) or **on** (HIGH).

Output Impulse
You can program the output of an impulse as follows:

1. Select one of the outputs.
2. Activate the **Output Impulse** option.
3. Set the length of the impulse in ms.
4. Specify the state: **off** (LOW) or **on** (HIGH).

Example

See the demo program **up_lio.vc**. >> *"Demo Programs", page 429*

6.4.16 Activate Parameter-Set

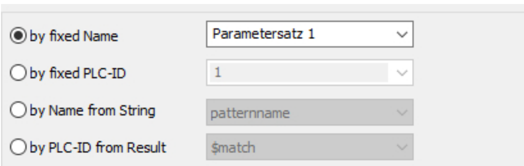
With **Control > Activate Parameter-Set** you write a command into the test program that changes the active Parameter-Set-Definition.

If the selected Parameter-Set-Definition does not exist on the vision system or the geometry variable used is not defined, no change takes place.

If errors occur when opening the new Parameter-Set-Definition (e.g. defective file), the default Parameter-Set is activated. In this case, all values are set to the default values of the Parameter-Set-Declaration.

Setting Parameters

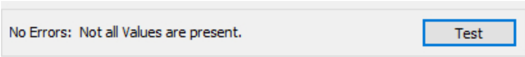
Enter the parameters in the dialog:



Parameter	Description
by fixed Name by fixed PLC-ID by Name from String by PLC-ID from Result	Select the method with which the Parameter-Set-Definition is to be selected. Select the respective Parameter-Set from the corresponding drop-down list.

Evaluation Parameters

By activating the [Test] button, the command is executed with the set parameters.



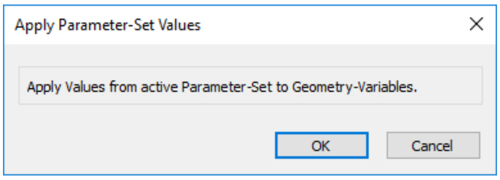
In the area to the left of the [Test] button the evaluation of the command (No Errors and a note, whether all parameters of the Parameter-Set-Declaration are present in the Parameter-Set-Definition) or an error message (for example: Parameter-Set-Definition not found).

If not all parameters of the Parameter-Set-Declaration are present in the Parameter-Set-Definition, the geometry variables assigned to the parameters are set with the respective default value when the command **Control > Apply Parameter-Set Values** is called.

After completing the configuration, add the command to the test program with the [OK] button.

6.4.17 Apply Parameter-Set Values

With **Control > Apply Parameter-Set Values** you write a command to the test program, which applies the values of the active Parameter-Set to the assigned Geometry-Variables.



The command fails if there is no Parameter-Set-Declaration.

6.4.18 Port I/O

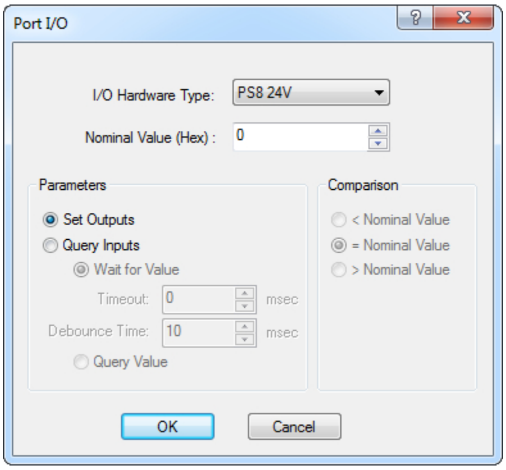


Image 184: Dialogue Port I/O

Create control commands for either communication between the vision system and the process environment (e.g. with a PLC), or for direct control of feed systems using **Control > Port I/O**.

Use this command to read digital inputs and set or reset outputs. A common use would be input from ADCs and output to DACs.

Basics

The port I/O commands group several lines of a port together. The method is similar to line I/O except that more than one input/output can be tested at the same time using a single command.

The current signals are compared to defined nominal states. Depending upon whether the condition is fulfilled, the command will return either **successful** or **unsuccessful**. This result can be evaluated for a single command or for a condition block and can be used to control branching.

>>*"Branch", page 367*

Input Parameters

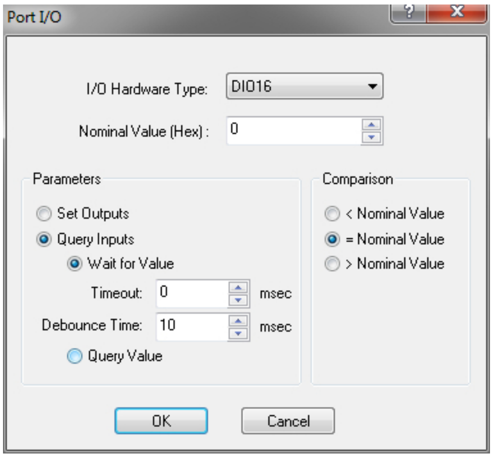
The procedure for controlling interfaces is the same for testing either type of I/O hardware. The Port I/O command always refers to all lines of an interface

Hardwaretype	Beschreibung
PS8	4/4-way input/output at pictor M ; 4 digital input, 4 digital output.
DIO4/6	4/6-way input/output port at pictor T-ECAN; 4 digital input, 6 digital output.
DIO4	4/4-way input/output port at pictor T-SC; 4 digital input, 4 digital output.
DIO8	8/8-way input/output port at vicosys; 8 digital input, 8 digital output.
DIO16	16/16-way input/output port atvicosys; 16 digital input, 16 digital output.

Query Inputs

The nominal signal is valid for all lines of the port and is entered as a hexadecimal value.

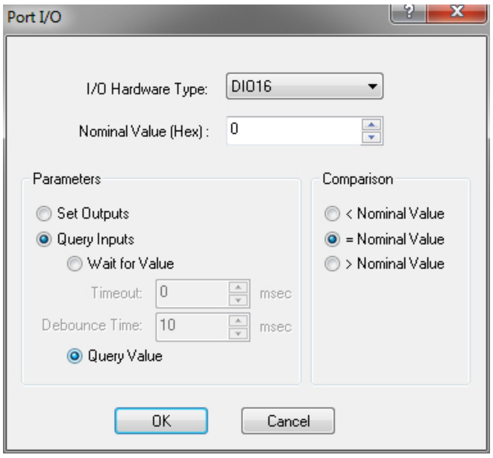
Wait for Value defines the permissible delay time programmed before a nominal value (signal pattern for a single input) has to appear or no longer appears at the input port
In addition, the length of time before the signal pattern must be available can be entered.



1. Enter the nominal value e.g. {4}.
2. Activate the **Query Inputs** and **Wait for Value** options.
3. Set timeout to the nominal value e.g. {1000}.
   Note: If you set the waiting time to {0}, vcwin will carry on waiting until the condition is fulfilled (no timeout).
4. Select whether the system should wait for exactly the nominal value or for the first result below or above the nominal value.
5. The nominal dwell time in the nominal condition is set under debounce time.

Query Value

With this you can write a command for testing the input of the I/O module for a defined input signal pattern:

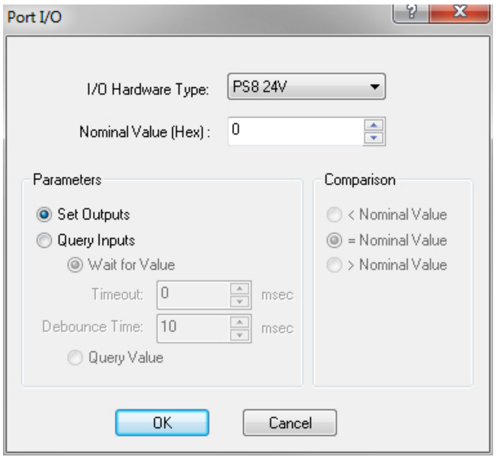


1. Enter the nominal value (hex) e.g. {4}.
2. Activate the **Query Inputs** and **Query Value** options.
3. Select whether the system should wait for exactly the nominal value (=) or for the first result below (<) or above (>) the nominal value.

Set Outputs

Setting an Output to a Value

With this you can program a defined output signal.



1. Enter the nominal value e.g. {4}.
2. Activate the **Set Outputs** option.

Example

See the demo program **up_pio.vc**. >> *"Demo Programs", page 429*

6.4.19 Port Control

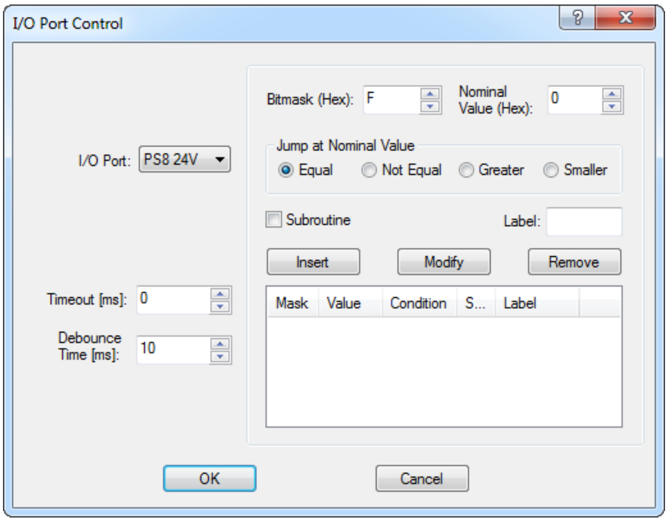


Image 185: Dialogue Port Control

Depending upon the input signal of the hardware inputs, you can define jump commands to labels or subroutines using **Control > Port Control**. With this, you can arrange several conditions in a branching list if necessary. Branching is executed for the first condition fulfilled.

For better clarity, the command is colored blue by default. You can change the colour using Options -> application settings -> Font *see "Application Settings", page 102*

Basics for Command Processing

The hardware inputs are interrogated cyclically. The program only leaves the command if one of the two following results occurs:

- Condition was fulfilled.
- Timeout was exceeded.

Timeout does not have any effect on its own. Because of this, a condition must always be fulfilled before a command can be exited.

In the case of several conditions to be fulfilled, the label at the bottom of the branch list is jumped to first.

Input Parameters

I/O Port
The hardware input to be interrogated is set:

Hardwaretype	Beschreibung
PS8	4/4-way input/output at pictor M ; 4 digital input, 4 digital output.
DIO4/6	4/6-way input/output port at pictor T-ECAN; 4 digital input, 6 digital output.
DIO4	4/4-way input/output port at pictor T-SC; 4 digital input, 4 digital output.
DIO8	8/8-way input/output port at vicosys; 8 digital input, 8 digital output.
DIO16	16/16-way input/output port atvicosys; 16 digital input, 16 digital output.

Times of the Input Signal

Times of the Input Signal	Description
Timeout [ms]	Input is interrogated for the duration of the timeout. If timeout = 0 is entered, the program waits until one of the conditions is fulfilled, with a label as the jump target.
Debounce Time [ms]	If the status of the input coincides with the nominal value, the program waits for the duration of the debounce time entered. If the status still coincides after the debounce time has expired, then the program branching is executed.

Bitmask and Nominal Value
There is a bitmask and a nominal value for defining the nominal signals for executing the branching.

Bitmask and Nominal Value	Description
Bitmask	Hexadecimal number which defines which bits of the input are relevant. Example: 30 (Hex) means bit 4 = 1 and bit 5 =1, i.e. bit 4 and 5 are used to determine the nominal value.
Nominal Value	Hexadecimal number which defines the nominal value of the port signal

Jump at Nominal Value
The following branching conditions are possible:

Jump at Nominal Value	Description
Equal	Actual value at input is equal to nominal value.
Not Equal	Actual value at input is not equal to nominal value.
Greater	Actual value at input is greater than nominal value.
Smaller	Actual value at input is smaller than nominal value.

Jump Types

ADVICE

After executing the called program part, the program jumps to the instruction before the Port Control instruction and executes the next instruction.

Jump types	Description
Subroutine	Subroutine called if condition fulfilled.
Label	Jump to the label entered if the condition is fulfilled.

Branching List

All branchings which can be controlled from the input are edited in the branching list.

To insert an entry into the branching list

1. Mark the position in the branching list under which the new branching condition is to be inserted with a mouse click (does not apply to the first entry).
2. Set the branching parameters (above the table).
3. Click on the [Insert] button.

To change an entry in the branching list

1. Mark the branching condition to be replaced with a mouse click.
2. Replace branching parameters (above the table).
3. Click on [Edit] button.

To delete an entry in the branching list

1. Mark the entry with a mouse click.
2. Click on the [Remove] button.

6.4.20 Branch

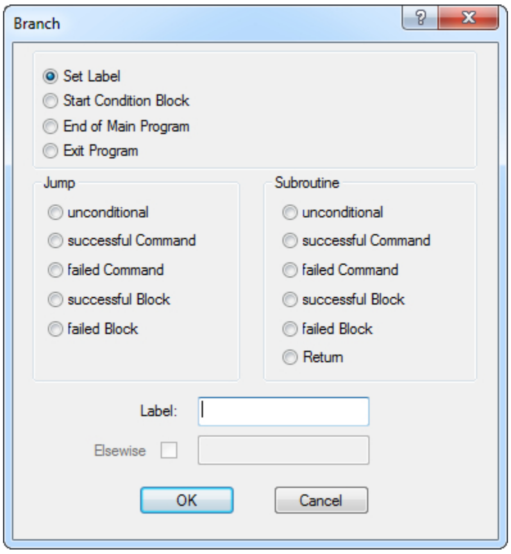


Image 186: Dialogue Branch

Create program branches and labels for program branches in test programs using **Control > Branch**. Conditional jump commands use the success/failure state of the most recent command or command block. Always write program control commands immediately after commands where success or failure determine how the program should react.

For better clarity, the command is colored blue by default. You can change the colour using Options -> application settings -> Font *see "Application Settings", page 102*.

Depending on the result of the previous command or command block the **Branch** command functions as follows:

Result of the previous command / command block	The program branch is implemented (it is otherwise ignored), if one of the following branch conditions is fulfilled …
Successful	• Successful command / successful command block or • Unconditional
Failed	• Failed command / failed block or • Unconditional

If the result of the previous command does not fulfill the branch conditions then a branch does only take place, if the **Elsewise** option is active. If the **Elsewise** option is not active, then the next command in the test program is executed. When the test program is sent to the vision system, program control commands are not checked.

Command Types and Branching Conditions

Command Types

The following branch operations are possible:

- Set labels.
- Set the start of a condition block.
- Set the end of the main program.
- Exit program.

The following branch operations are possible:

- Unconditional jump to a label.
- Conditional jump to a label.

The following control commands are possible for jumps:

- Unconditional subroutine call.
- Subroutine call if condition fulfilled.
- Exit subroutine.

Possible Branching Conditions

ADVICE

A conditional block counts as failed if a single command of the conditional blocks has failed, e.g. because a tolerance was exceeded.

Conditions used for determining whether to branch or execute the next command:

- successful command
- successful command block
- failed command
- failed command block

Input Parameters

General Control Commands

General Control Commands	Description
Set Label	Inserts a marker label into the test program: A name for the label has to be entered.
Start Condition Block	Defines the beginning of a block of commands in the test program: The condition block unites the results of several individual commands into a single result. The final result of a condition block is therefore **unsuccessful** if any of the commands within the block are **unsuccessful**.
End of Main Program	Defines the end of the main part of a program. This command ends the test cycle and starts the program loop anew from the first command.
Exit Program	Defines a command for ending the program. This causes the program to terminate checkups and return to the start menu. In the pictorsystem, run mode is stopped and the system then waits for remote control commands.

Control Commands for Jumps

Control Commands for Jumps	Description
Unconditional jump to a label	Activating the unconditional option causes an immediate jump to a defined label within the test program. A name for the label has to be entered.
Conditional jump to a label	The various conditions possible for a conditional jump command can be chosen from the menu during editing. A name for the label has to be entered.

The jump is executed during program execution if the conditions chosen are fulfilled. Each time the conditions refer to the last command or command block before the jump command and can be:

- a successful condition block
- a successful command
- a failed condition block
- a failed command

Control Commands for Subroutines

ADVICE

Subroutines should never be finished or excited by any other command except **Return**. Otherwise the jump back addresses collected in the stack are not cancelled and this leads to the error message **Too many subroutine calls**. In order to reset the stack, insert the **Return** command into the subroutine, then call the **Utilities > Test Section** command.

Control Commands for Subroutines	Description
Unconditional subroutine calls	The options for **Subroutine** creates calls to subroutines in the program. The name of the subroutine must be entered from the keyboard in the Label field.
Conditional subroutine calls	Select the condition so that the test program will call a subroutine during editing. The name of the subroutine must be entered from the keyboard in the Label field. When the test program is run, the program will jump to the defined subroutine if the condition is fulfilled, otherwise the next command will be executed. The conditions are success or failure of the preceding command or condition block: • a successful condition block • a successful command • a failed condition block • a failed command
Leave subroutine (Return)	A command to **leave the subroutine** is put in the test program. After the subroutine executes, it jumps back to the call location.
Elsewise	Permits conditional jumps and subroutine calls for conditions that are not fulfilled and so allows the construction of an IF-THEN-ELSE logical structure.

Example for Elsewise

```
*****         Last Change at vcwin pro 2.13.126; Vision System: All commands
*****         ****
*****         conventional branch instruction
*****         ****
00000         Delete Graphic Colour
00001         Standard Image Acquisition with camera 0
00002         RES[0] = RES[0] + 1; Nominal Value 0 (+10/-10)
00003         Average Gray Value in Window [100, 100/100, 100]
                Result 1; Nominal 200(+10/-10)
00004 -->     Jump to Label "NIO" after failed Command
00005 -->     Jump to Label "IO" (unconditional)
*****         ****
*****         branch construction with ELSEWISE
*****         ****
00006         Delete Graphic Colour
00007         Standard Image Acquisition with camera 0
00008         RES[0] = RES[0] + 1; Nominal Value 0 (+10/-10)
00009         Average Gray Value in Window [100, 100/100, 100]
                Result 1; Nominal 200(+10/-10)
00010 -->     Jump to Label "NIO" after failed Command; elsewise "IO"
```

6.4.21 Save Process Data

Overview

The command **Save Process Data** is located in the Command Selection under **Control**.

Use and function

- Storage of process data in a CSV file.
- Storage of the CSV files on the device or on USB flash drives, CF cards or servers (command **External Storage Device**)

- First time running the command creates a table with the specified names. The values of the elements are stored in the table as a single row. Each subsequent run extends the table with a new row with the current values of the elements.
- The option to append the date to the file name automatically creates every day a new CSV file.
- The CSV files can be opened with a spreadsheet program or processed automatically.

TIP

Use the command **External Storage Device** to store the file. As a result, the processing time of the command will be shortened.

Dialog Structure

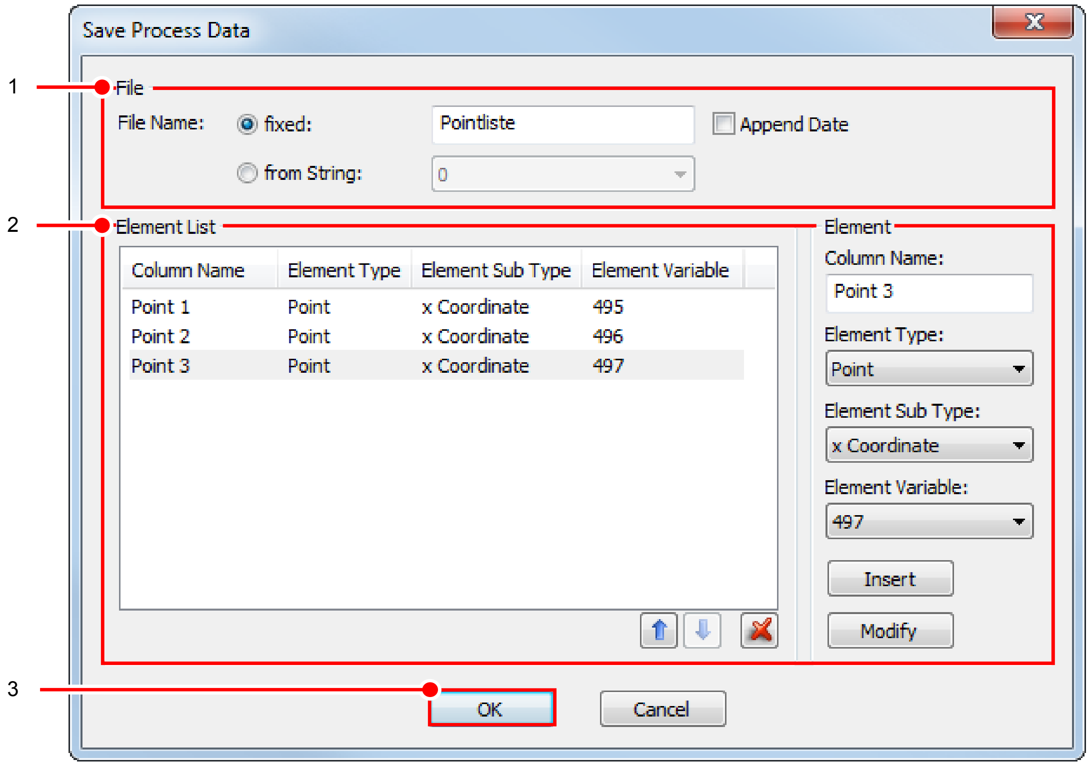


Image 187: Dialog Save Process Data

Work Flow

1. Assign a file name

   Select whether the file name is assigned fixed or assumed from a string.

Parameter	Description
fixed	File name under which the file is stored. The file name is appended with .csv.
from String	The file is stored with the name that is in the string. The String has to be created in the program before. The file name is appended with .csv. Please note: For error-free saving, the content of the string has to accord to the name conventions of the vision system.
Append Date	If this option is enabled, a new file is created for each day. The file is stored in the form "File Name_YYYY-MM-DD.CSV ".

2. Select elements

   Group box **Element**

   Here you can select the elements to be stored.

Parameter	Description
Column Name	Enter here the column name. It is inserted in the table header (first line). If no name is specified for all columns, no first row (table header) is inserted in the table.
Element Type	Select here the element type. (e.g.: Result, Point, Line, Circle, String, Date, Time).
Element Sub Type*	Select here the Element Sub Type (e.g.: x Coordinate at Element Type Point).
Element Variable*	Select here the Element Variable. Depending on the selected Element Type, the corresponding element list is displayed.
* Grayed out for date and time. The date is written in the column in the form YYYY-MM-DD, the time HH:MM:SS.	

   - With [Insert] you add an element to the Element List.
   - With [Modify], you can configure an element from the Element List. To do this, select the element in the group box "**Element List**". Change its values in the group box "**Element**".
     Apply the changed values by clicking the [Modify] button.

   Group box "**Element Liste**"

   Here all the elements stored in the table are listed.

   - With the button you can move an element in the element list up resp. one column forward. For this purpose select the item and click the button.
   - With the button you can move an element in the element list down resp. one column backward. For this purpose select the item and click the button.
   - With the button you can remove an element from the element list. For this purpose select the item and click the button.

3. Use the [OK] button to insert the command into the test program.

   If the program executes the command, all elements from the list are saved in a new row. Each subsequent call of the command adds a new row.

Structure of the data file

- Character set: ISO-8859-15
- Column separation by ";"
- Strings in quotation marks
- Date and time according to ISO 8601 in the format YYYY-MM-DD and HH:MM:SS

Error messages

Invalid Parameter

- At least one parameter is invalid.
- Check the settings of the command. Check that the used mode of the command is supported by the vision system used.

File system error

- An error occurred while saving.
- Check if the storage medium is reachable.
- Check if you have access rights for the chosen directory.

Not enough memory

- The storage medium is full.
- Create more space on the storage medium.

Invalid file name

- The file name contains invalid characters.
- Use a valid file name.

Not supported for FTP

- Do not use FTP.

6.4.22 Process Module

Overview

ADVICE

For information also refer to the manual of the process coupling module.

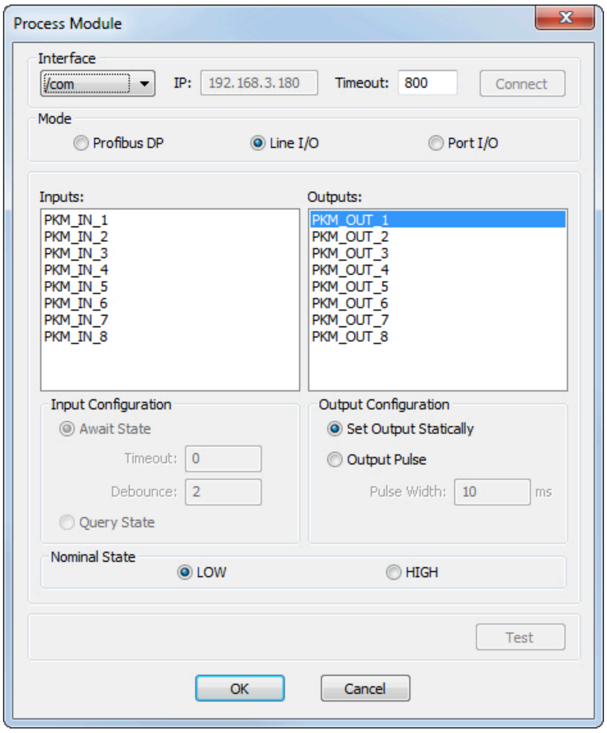


Image 188: Dialog Process Module

With **Control > Process Module**, vision systems can communicate with other systems via Fieldbus, Industrial Ethernet and/or through digital I/O.

The data exchange takes place by writing and reading elements.

Interfaces

As an internal interface (device-dependent)

- PROFINET
- sercos III
- modbus
- CANopen

ADVICE

The internal interface is configured via **System Preferences > Fieldbus settings**.

Via gateways of the Hilscher netTAP series

- PROFIBUS-DP
- DeviceNet
- CC-Link
- CANopen
- RS232
- EtherCAT
- Ethernet Powerlink
- PROFINET-IO
- Ethernet/IP
- sercos III

ADVICE

The gateways of the Hilscher netTAP series must be connected via /modbus.

The configuration of the gateways of the Hilscher netTAP series takes place via the corresponding software of the manufacturer.

Via gateways of the PKM series

- Digital I/Os (8/8)
- Profibus DP (Slave, 30/8)

Establish a connection to the process coupling module

1. Establish a connection to the vision system.
2. Select **Control > Process Module**.
3. Select the interface.

Parameter	Description
/com, /ethernet, /CANopen, /sercos III, /modbus, /PROFINET	Display of the available interfaces of the device (device-dependent).
Timeout	Waiting time of the vision system for a response from the process coupling module.

After selecting the interface, press the [**Connect**] button to establish the communication with the process coupling module. This function is not required for some interfaces (PROFINET, sercos III, CANopen).

Read elements

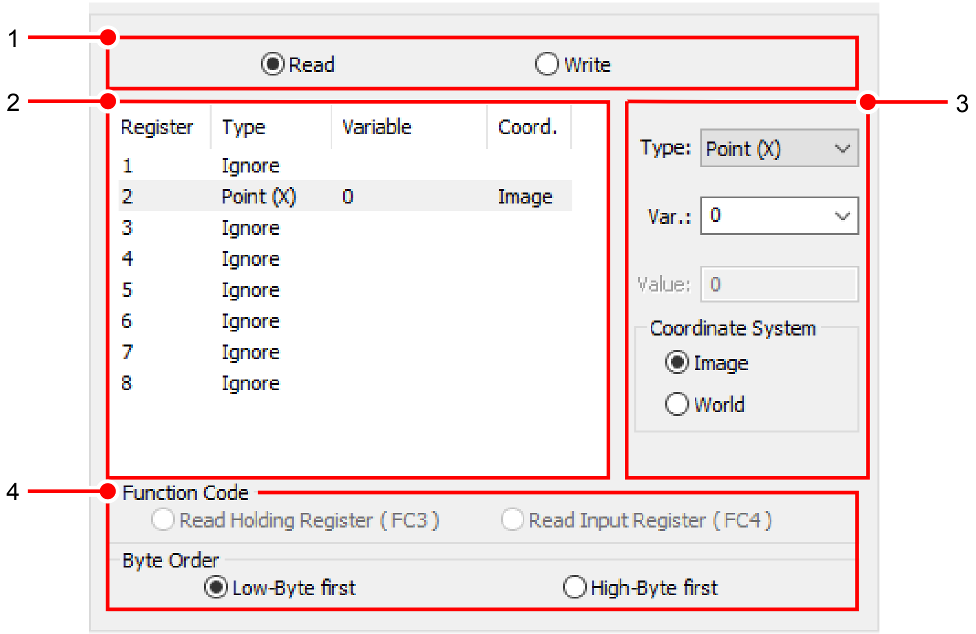


Image 189: Dialog Process Module - Read Elements

1. Activate the **Read** mode.
2. Select the register from which elements should be read.
3. Under *Type* select the element type.

Type	Description
Ignore	Nothing is read from this register.
Result	A value of the type Result is read from this register.
Point (X)	The X coordinate of a point is read from this register.
Point (Y)	The Y coordinate of a point is read from this register.

   If required, enter under *Var.* the corresponding number or name.

   For point coordinates, select the coordinate system. This is a specific setting for the selected register.

   Repeat step 2 and 3 until all registers from which elements are to be read are configured.

4. Depending on the selected interface, enter further required parameters (e.g. byte order or register limit).

Write Elements

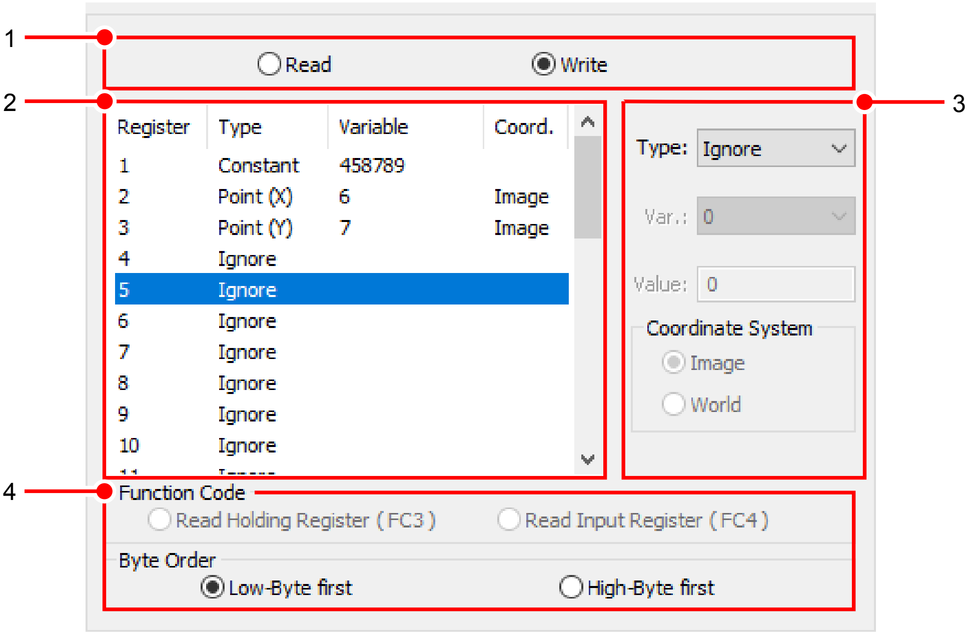


Image 190: Dialog Process Module - Write Elements

ADVICE

The command does not interrupt when sending undefined geometry elements. The command is executed and sends the following values:

short: 0x8000

long: 0x80000000

1. Activate the **Write** mode.
2. Select the register into which elements should be written.
3. Under *Type* select the element type.

Type	Description
Ignore	Nothing is written in this register.
Constant	A constant value is written in this register. Enter this as an integer in the *Value:* field.
Result	A value of the type Result is written in this register.
Point (X)	The X coordinate of a point is written in this register.
Point (Y)	The Y coordinate of a point is written in this register.

If required, enter under *Var.* the corresponding number or name.

For point coordinates, select the coordinate system. This is a specific setting for the selected register.

Repeat step 2 and 3 until all registers in which elements are to be written are configured.

4. Depending on the selected interface, enter further required parameters (eg byte order or register limit).

Function /modbus

Image 191: Dialog Process Module - option Function Code

With this option, the Modbus function code for reading registers is selected:

Selectable register areas

- Read Holding Register (FC3)
- Read Input Register (FC4)

16 bit registers Interface /modbus

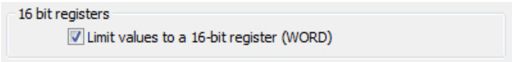


Image 192: Dialog Process Module - option 16 bit registers

With this option, the internal 32 bit values of the vision system are shortened on a 16 bit register. The upper 16 bits are cropped. With this option, the number of available registers doubled.

If the option is not active, the internal 32 bit values of the vision system are divided to two 16 bit register (WORD-register).

Synchronization

When using external gateways (PKM-PB, PKM-PB/v, Hilscher netTAP...), the command Process Module is not synchronized to the bus cycle of the bus system on which the gateway converts.

If several registers of the process coupling module are written, these changes can be effective in different bus cycles. Therefore, the PLC for data transmission must be synchronized with the Vision System.

On way to synchronize can be done with a digital START signal from the PLC and a READY signal from the Vision System. These signals can be sent through the digital inputs and outputs of the communication partner device.

Another way to synchronize can be done with a counter on the Vision System, which counts up at each test and writes that value to the field bus register. This should be done either in a separate Process Module command after the Process Module command or as the last (used) register in the Process Module command. On the master should be monitored, at which time the value increases. With each increase, a measurement is made and all new values are available.

IO-Modul

Connection of the IO module (PKM-PLC8) can be optionally done through Ethernet or through RS-232. After configuring, establish communication between the process interfacing module and the operating software vcwin with the [**Connect**] button. The **Line IO** and **Port IO** operating modes are activated. The use of these two operating modes is analogous to the standard control commands **Line IO** and **Port IO**.

>> *see "Line I/O", page 359*

>> *see "Port I/O", page 362*

6.4.23 Define Test Window Detection Window

Overview

With **Control > Define Test Window Detection Window** you write a command for creating a test window in the test program. The test window can be defined from already determined values of other variables or by fixed assigned values.

When defining, each test window can be assigned an X and Y point for position tracking. These points are saved with the test window. If a command containing a test window is executed, the values of the variables are applied and the test window is adjusted accordingly.

The defined test window is stored in the geometry list Test Windows under a number or a name and can be selected as a geometry in suitable commands.

Treatment of world coordinates

- Test Windows are always in image coordinates.
- Values from results are interpreted as image coordinates.
- Fixed values are interpreted as image coordinates.
- Points that exist in world coordinates are internally converted into image coordinates.

Workflow

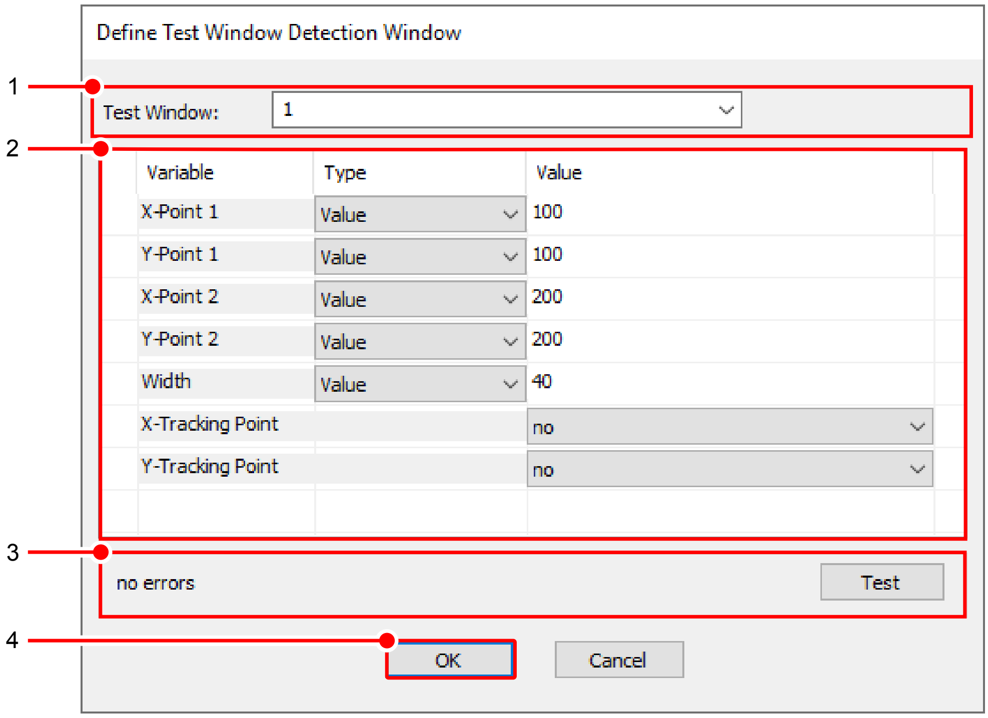


Image 193: Dialogue Define Test Window Detection Window

1. Enter here the number or the name of the defined test window under which it should be stored in the geometry list Test Windows.
2. For each variable, select the type from the drop-down list. Enter the values or select the corresponding variables.

   For a position tracking, select the relevant variables of the tracking points under x-tracking point or y-tracking point.

3. By activating the [Test] button, the command is executed with the set parameters.

   The test window is stored in the geometry list Test window and the test window is displayed in the monitor window.

4. Use the [OK] button to insert the command into the test program.

6.4.24 Define Test Window Detection Beam

Overview

With **Control > Define Test Window Detection Beam** you write a command for creating a test window in the test program. The test window can be defined from already determined values of other variables or by fixed assigned values.

When defining, each test window can be assigned an X and Y point for position tracking. These points are saved with the test window. If a command containing a test window is executed, the values of the variables are applied and the test window is adjusted accordingly.

The defined test window is stored in the geometry list Test Windows under a number or a name and can be selected as a geometry in suitable commands.

Treatment of world coordinates

- Test Windows are always in image coordinates.
- Values from results are interpreted as image coordinates.
- Fixed values are interpreted as image coordinates.
- Points that exist in world coordinates are internally converted into image coordinates.

Workflow

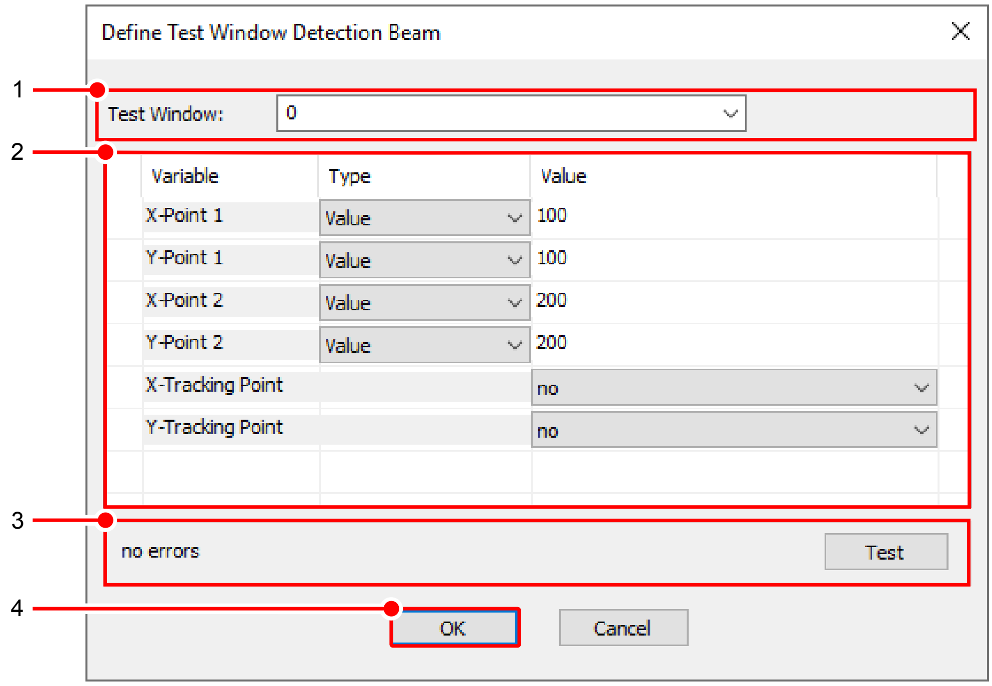


Image 194: Dialogue Define Test Window Detection Beam

1. Enter here the number or the name of the defined test window under which it should be stored in the geometry list Test Windows.

2. For each variable, select the type from the drop-down list. Enter the values or select the corresponding variables.

   For a position tracking, select the relevant variables of the tracking points under x-tracking point or y-tracking point.

3. By activating the [Test] button, the command is executed with the set parameters.

   The test window is stored in the geometry list Test window and the test window is displayed in the monitor window.
4. Use the [OK] button to insert the command into the test program.

6.4.25 Define Test Window ROI

Overview

With **Control > Define Test Window ROI** you write a command for creating a test window in the test program. The test window is defined by a graphic control element.

When defining, each test window can be assigned an X and Y point for position tracking. These points are saved with the test window. If a command containing a test window is executed, the values of the variables are applied and the test window is adjusted accordingly.

The defined test window is stored in the geometry list Test Windows under a number or a name and can be selected as a geometry in suitable commands.

Treatment of world coordinates

- Test Windows are always in image coordinates.
- Values from results are interpreted as image coordinates.
- Fixed values are interpreted as image coordinates.
- Points that exist in world coordinates are internally converted into image coordinates.

Properties

- The test window is formed by a polygon.
- The polygon must consist of at least 3 points.
- The polygon can contain a maximum of 100 points.
- Points and lines of the polygon are part of the test window.
- The lines of the polygon must not intersect.
- Due to position tracking, points can lie outside the image area. In this case a warning is displayed.

Workflow

The current image memory page is displayed in the dialog. Above this, the test area, taking into account the position tracking, is displayed as a polygon.

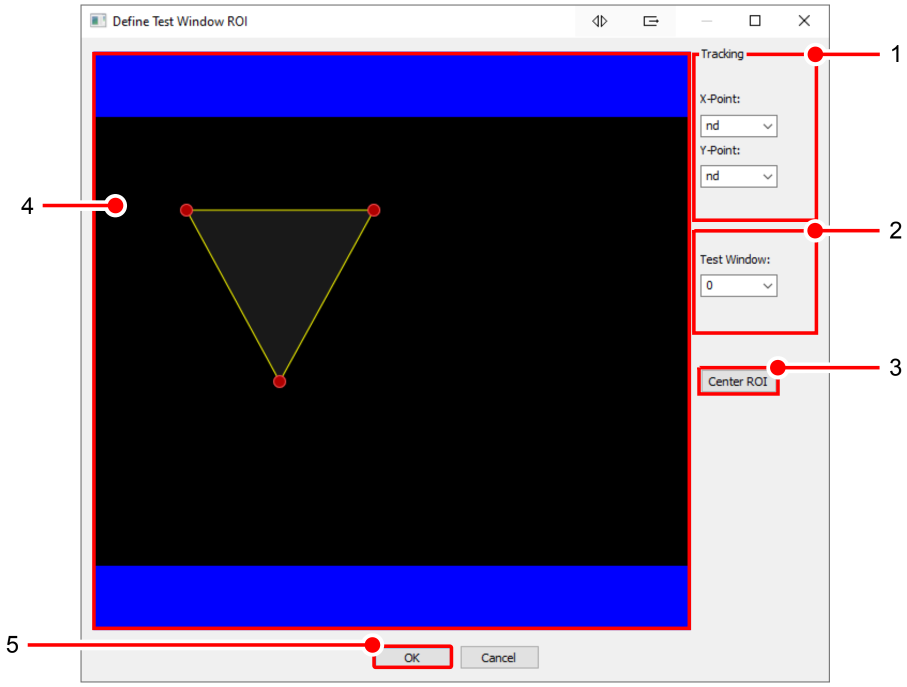


Image 195: Dialogue Define Test Window ROI

1. For position tracking of the test window, select the relevant variables of the tracking points X and Y under Tracking.
2. Enter here the number or the name of the defined test window under which it should be stored in the geometry list Test Windows.
3. By activating the button [Center] the test area is centered in the middle of the image area. Use this function if the polygon is no longer visible in the image area due to position tracking.
4. Create the desired test window here.
5. Use the [OK] button to insert the command into the test program.

Create test window

Add point

- Move the mouse pointer to the position on the line at which the point should be inserted. The mouse pointer is shown as a hand.
- Double click on the line with the left mouse button. A new point is inserted at the position.

Remove point

This function is only available if the polygon consists of more than 3 points.

- Move the mouse pointer to the point to be removed. The color of the point changes.
- Right click on the point. The point is removed from the polygon.

Move point

- Move the mouse pointer to the point to be moved. The color of the point changes.
- Hold down the left mouse button. Move the mouse to move the point.

Move polygon

- Move the mouse pointer into the polygon. The mouse pointer is shown as a cross arrow.
- Hold down the left mouse button. Move the mouse to move the polygon.
- The polygon cannot be moved completely out of image area.

6.4.26 Define Test Window Rectangle

Overview

With **Control > Define Test Window Rectangle** you write a command for creating a test window in the test program. The test window can be defined from already determined values of other variables or by fixed assigned values.

When defining, each test window can be assigned an X and Y point for position tracking. These points are saved with the test window. If a command containing a test window is executed, the values of the variables are applied and the test window is adjusted accordingly.

The defined test window is stored in the geometry list Test Windows under a number or a name and can be selected as a geometry in suitable commands.

Treatment of world coordinates

- Test Windows are always in image coordinates.
- Values from results are interpreted as image coordinates.
- Fixed values are interpreted as image coordinates.
- Points that exist in world coordinates are internally converted into image coordinates.

Workflow

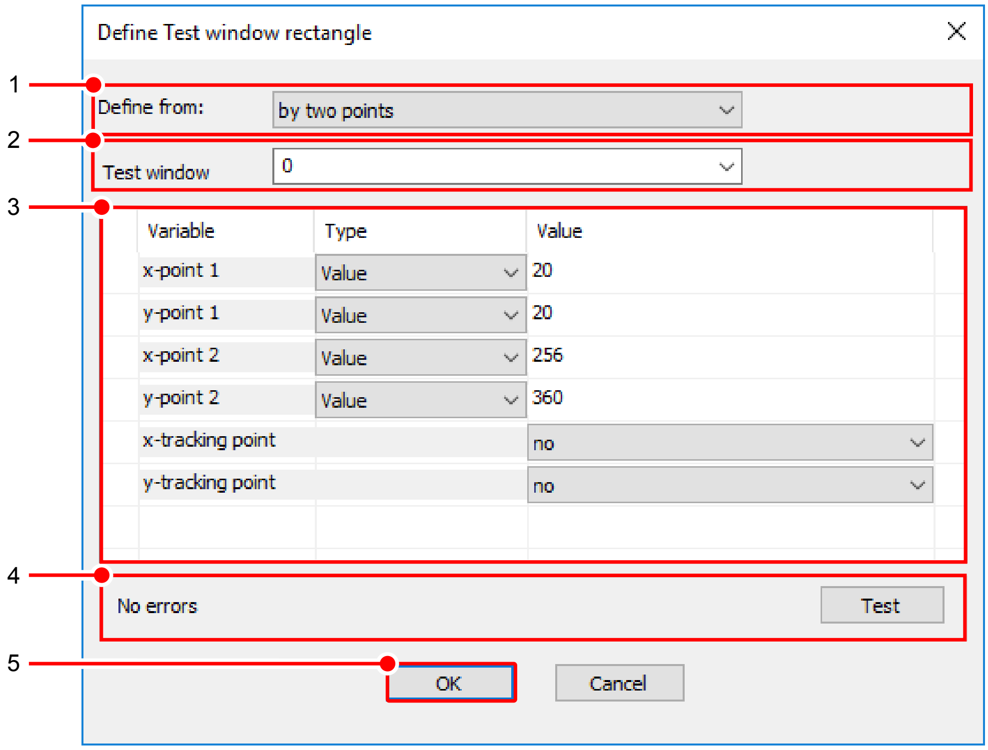


Image 196: Dialogue Define Test Window Rectangle

1. Define from which points the test window is generated.
2. Enter here the number or the name of the defined test window under which it should be stored in the geometry list Test Windows.

3. For each variable, select the type from the drop-down list. Enter the values or select the corresponding variables.

   For a position tracking, select the relevant variables of the tracking points under x-tracking point or y-tracking point.
4. By activating the [Test] button, the command is executed with the set parameters.

   The test window is stored in the geometry list Test window and the test window is displayed in the monitor window.
5. Use the [OK] button to insert the command into the test program.

6.4.27 Geometry Sets

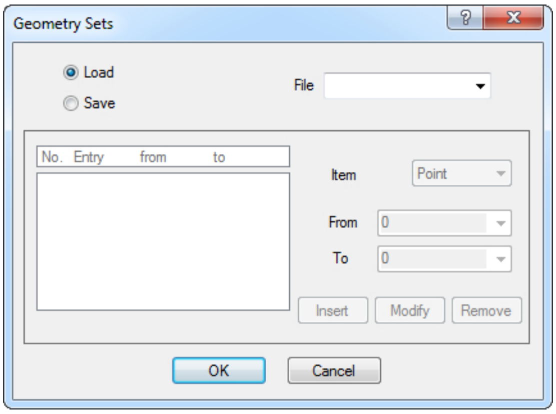


Image 197: Dialogue Geometry Sets

A command is written into the test program which allows geometry structures to be loaded or saved.

A reference geometry set is a set of points, circles, lines and contours which can be saved as a file while a test program is running and loaded again later on.

The various geometry elements can also be merged and saved in a single file.

To save geometry structures from the program into files

Follow these steps to select geometrical elements and save them as sets to disk:

1. Select the **Save** option.
2. Enter the file name under which you want the geometry structure to be saved.
3. Select an element: circle, line, point or contour.
4. Give the number or range of numbers for the type of geometry variable selected from above.
5. Use the [Insert], [Remove] or [Edit] buttons to put in, remove or change the respective element in the element list.
6. Click on the [OK] button.

To load geometry structures from files into the test program

ADVICE
The program makes a note of the most recently loaded geometry set to prevent unnecessary access to the hard disk. A geometry set is only reloaded after the test program is restarted i.e. the elements in a set cannot be changed while the test program is running.

1. Select the **Load** option.
2. Enter the name of the geometry set you want to load.
3. Click on the [OK] button.

Input Parameters

Parameters	Description
Load/Save	Instruction as to whether the reference geometry should be loaded or saved.
File	File in which the reference geometry should be saved.

6.4.28 Stopwatch

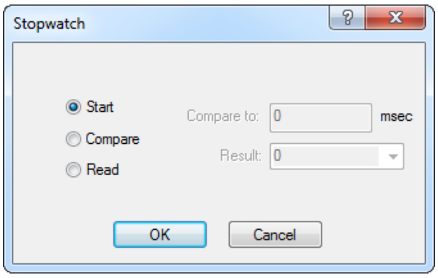


Image 198: Dialogue Stopwatch

Start an internal stopwatch and compare the time since the start of a function to a nominal value or result using **Control > Stopwatch**. Use the stopwatch for controlling or checking the time taken by parts or all of the test program.

Start Stopwatch

Select the Start option. Ready.

Comparing a Measured Time with a Value
During program execution: If the elapsed time is longer than the comparison value given, then the time has expired (command failed), otherwise not.
Teach-in:

1. Activate the **Compare** option.
2. Enter a comparative value.

Storing the Measured Time in the Result Structure
When employing this option, the stopwatch continues to run during the continuing program execution.
Teach-in:

1. Activate the **Read** option.
2. Enter the result number.

6.4.29 Save System Time

Evaluate system time in programs

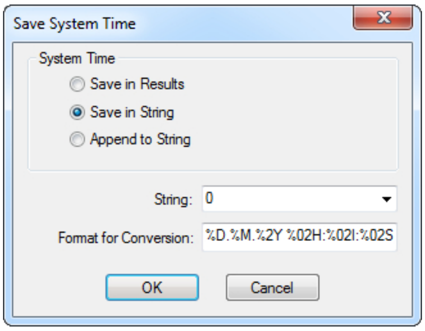
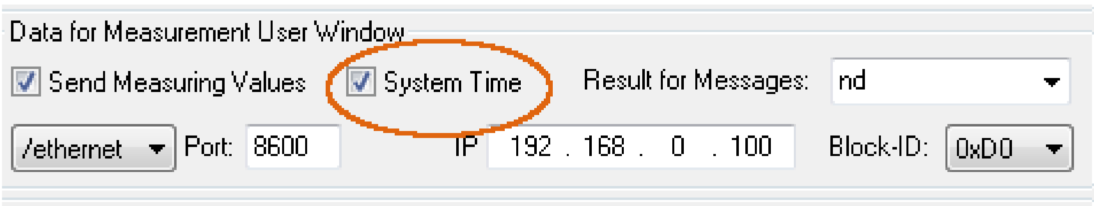


Image 199: Dialogue Save System Time

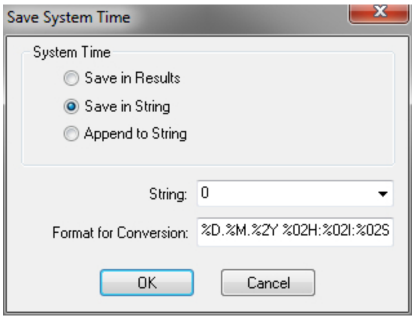
1. Save the system time in a result variable (RES) with **Save System Time > Save in Results**.
2. Send the result variable with **Control > Send Results > HEX**, e.g. to a PLC.
3. Leave the system time in Hex format for machine level evaluation.

Evaluate system time in the user interface and with machine code level evaluation



1. Activate the **System Time** option in the **Evaluation > Evaluate Result** command.
   The following data are sent if the **System Time** option is not active:
   - Total (4 Bytes)
   - Res no. 1, res no. 2, ….
     The additional following data can only be sent with the System Time option activated:
   - Date (4 Bytes)
   - Time (4 Bytes)
2. Use the **Evaluate Result** command to transfer the system time to, for example, a PLC.
3. Leave the system time in Hex format for machine level evaluation.
4. In addition, the system time can also be displayed via the **User Window**.

To use the Vision System's system time in ASCII format for logging and visualization



1. Save the system time in ASCII format as a string using **Save System Time > Save as String**.
2. You can define the formatting of the string yourself (see later on in this section.) Additional text is possible.
3. Send the string using **Control > Send Results > ASCII** to the protocol computer. Superimpose the string on the screen using **Evaluation > Superimpose**.

Syntax

Format for Date and Time		
Date	YYMD	(year, month, day)
Time	HMST	(hour, minute, second, tick)

Format for conversion: %[flags][width][type]		
Flags	0	
Width	length of the parameter	
Types	Y	year
	M	month
	D	day
	H	hour
	I	minute
	S	second
	T	tick

Example: Format for Conversion	**The ASCII text sent using Send Results**
%2D.%2M.%2Y %2H :%2I :%2S	25.03.07 14 :03 :45
Date 2D.%2M.%2Y; Clock time %2H:%2I: %2S	Date 25.03.07; Clock time 14:03:45

6.4.30 Wait

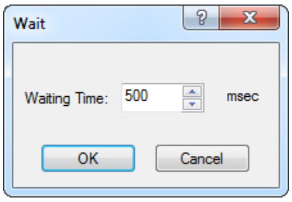


Image 200: Dialogue Wait

Define a command which sets how long the test program should wait until the required time has elapsed using **Control > Wait**. Enter the wait time required [ms] into the dialog window.

6.4.31 Counter

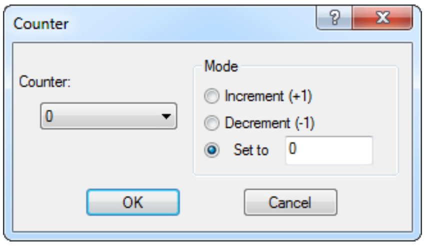


Image 201: Dialogue Counter

Insert a counter value operation into the test program using **Control > Counter**. Every counter is accessed via its number. Use this command to program loops and branches. Proceed as follows:

1. Enter the number of the counter to be changed.
2. Select the counter operation:
   - Increment (+1) … Adds 1 to the counter.
   - Decrement (-1) … Subtracts 1 from the counter. If 0 is reached or when the counter is negative, the result is unsuccessful. Use this option to program loops in programs in conjunction with Branch commands.
   - Set to.... Sets the counter to a positive integer: This field must contain a value. Entering 0 resets the counter.

ADVICE

Display the counters values using the **Evaluation > Superimpose** command. Counter operations do not relate to geometry variables and/or result variables in any way.

Example

See the demo program **count.vc**. >> *"Demo Programs", page 429*

6.5 Obsolete commands

6.5.1 Advanced Pattern Search

Overview

ADVICE

Important: The obsolete command **Advanced Pattern Search** should no longer be used for new programs. Instead of it use **Locate > 360° Pattern Search**.

Both commands work similar. The difference is that the **Advanced Pattern Search** uses correlation based on binary values and **360° Pattern Search** uses correlation based on gray values.

Image 202: Advanced Pattern Search -taught-in pattern

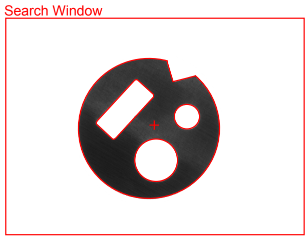


Image 203: Advanced Pattern Search -taught-in pattern is detected

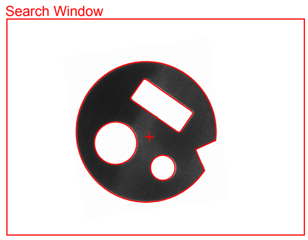


Image 204: Advanced Pattern Search -rotated pattern is detected

Use **Obsolete > Advanced Pattern Search** to locate objects in the image. By doing so, the angular position (in addition to the position) relative to a taught pattern is determined. The command finds the angular position with respect to a pattern. The search takes place in two steps, first a coarse search and then a finer search with smaller rotation steps.

Use this command to find rotated objects, to classify and subsequently to save the rotations in another image memory page. The image content so saved can be used for further processed with other algorithms.

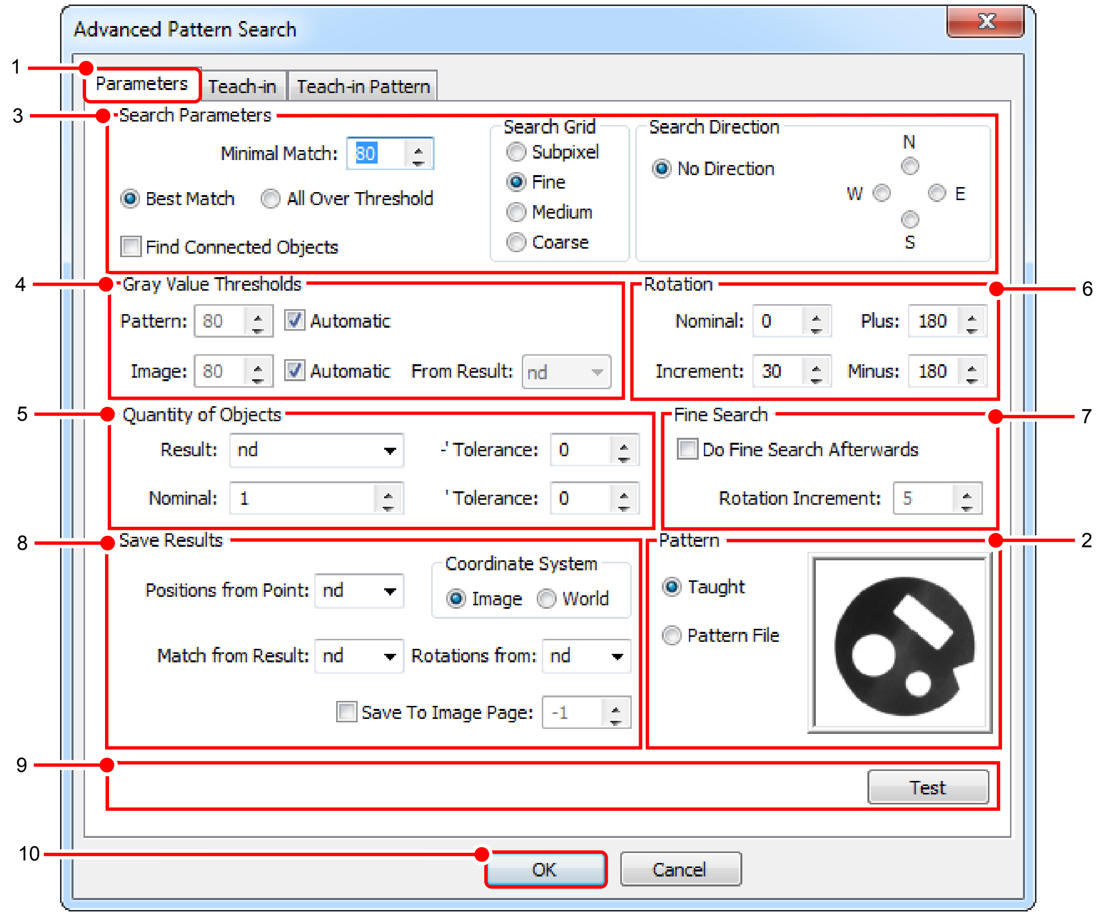
Work Flow:
Teach-in Pattern

1. Switch to the tab control "***Teach-in Pattern***".
2. Determine, by means of the geometry rectangle, which image content has to be read as the pattern. (Information for teaching, see also: *"Teaching in Test, Detection and Search windows", page 26*.
3. With the button [Read Pattern] you read-in all image information located in the geometry as the pattern, and stored it internally in the command.
4. In the reference window, verify if the pattern is complete.

Work Flow:
Teach-in Search Area

1. Switch to the tab control "***Teach-in***".
2. Please set the geometry, size and position of your search window. In this area, your pattern is searched (see also: *"Teaching in Test, Detection and Search windows", page 26*).
3. If necessary, set the parameters (X Point, Y Point) for position tracking (see also: *"Position Tracking of Objects", page 35*).

Work Flow:
Parameterizing and Testing



1. Switch to the tab control "***Parameters***".
2. Determine in the area "Pattern" whether a pattern should be taken from a file or the taught-in pattern.
3. In the area "Search Parameters" define the following parameters for the search.

Parameter	Description
Minimal Match	Minimum compliance (correlation) of the pattern found with the taught pattern in per cent, so that a pattern may be further processed as a valid pattern. Useful settings are 60 to 80%.
Best Match	Only the pattern with the best degree of match is saved.
All Over Threshold	All objects whose matching degree will be higher than the minimal match are found.
Search Grid	
Subpixel	Very fine search based on subpixel.
Fine	Fine search, e.g. for writing or gravure.
Medium	Search for medium-fine structures.

Parameter	Description
Coarse	Coarse search, e.g. for a rectangle without fine structure.
Search Direction	
No Direction	Every pattern is taken into account.
N/E/S/W	With a preferred search direction, the object is always chosen that is found first in the preferred direction. (N means that the first object is searched - starting from bottom to top.) The command fails if the first object found does not have the required degree of match

4. Set in the area "Gray Value Thresholds" the parameters for binarization.

Parameter	Description
Pattern	Binarization threshold for the taught pattern. In automatic mode, the threshold is determined automatically.
Image	Binarization threshold for the test image. In automatic mode, the threshold is determined automatically.
From Result	Binarization threshold for the test image is taken from a result.

5. Quantity of Objects

Parameter	Description
Result	Enter the result number or name in which the number of patterns found, is to be saved.
Nominal	Enter the number of objects that should be found, so that the command is considered successful.
Tolerances	Enter the permitted upward and downward deviations from the nominal value.

6. Rotation

Parameter	Description
Nominal	Angle from which the first test starts.
Increment	Rotational increment between two tests.
Plus/Minus	Limit of the rotation range for all following tests.

7. Fine Search

Parameter	Description
Do Fine Search Afterwards	With this option selected, after successful angle determination by means of the given increment, a search with finer rotation increments is carried out.
Rotation Increment	Rotational increment between two tests.

8. Define in the area "Save Results" the settings for storing the results.

Parameter	Description
Positions from Point	Starting point name or number for saving the center of the pattern found. All following points are saved consecutively.
Coordinate System	Saving in image coordinates or world units.
Match from Result	Starting result name or number for saving the degree of match of the patterns found. All following matches are saved consecutively.
Rotations from	Starting result name or number for saving the angle of the patterns found. The angles are saved clockwise from 0° to 360°.
Save to Image Page	Image Page, in that the found pattern is stored derotated and centralized for post-processing. It is stored as follows: • using undirected search (without direction) the pattern with the best match • using directed search (N/E/S/W) the first pattern found in the chosen direction

9. Result Patterns Found

   After activating the [Test] button, the number of patters found is listed in the area left to the [Test] button. Furthermore, the Position (in subpixels), Rotation, and Match value of the first pattern found according to the taught pattern are shown. In parallel, the edges of the pattern found are marked in red in the video image. The center is symbolized by a red cross.

10. Use the [OK] button to insert the command into the test program.

Strategy for Setting Search Grid

- For critical applications, the pattern must always do a coarse search first.
- If Coarse is not successful in the [Test], activate Medium, Fine or Subpixel.

Function	Sensitivity	Speed	Remarks	Sketch
Subpixel	grid search 1 Pixel	slow	exact pixel search	1 x 1
Fine	grid search 2 pixels	–		2 x 2
Medium	grid search 4 pixels	4 x faster than fine search		4 x 4
Coarse	grid search 8 pixels	16 x faster than fine search	fastest, but least accurate algorithm	8 x 8

6.5.2 Gateway Field Bus

ADVICE

Important: The obsolete command "Gateway Field Bus" should no longer be used for new programs. Instead of it use the command "Process Coupling Module".

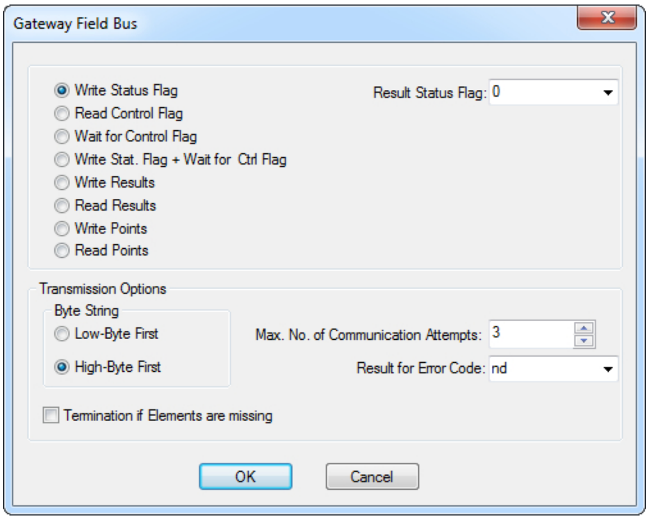


Image 205: Dialogue Gateway Field Bus

Enable the transfer of data into the test program via a field bus gateway to be configured using **Control > Gateway Field Bus**. There are various options for signals, results and points available.

Setting Options for Information Content

The upper left part of the dialog window allows specification of which information can be transferred. Every option requires is own individual parameters to be set in the corresponding fields on the right hand side.

The following options are available:

- Write status flag
- Read control flag
- Wait for control flag
- Write status flag and wait for control flag
- Write / read results
- Write / read points

For further details about the possibilities available for information transfer please refer to the **Fieldbus Gateway** documentation.

Setting Transfer Options

Byte Sequence	Description
Low-Byte First	Low-Byte is sent before the High-Byte.
High-Byte First	High-Byte is sent before the Low-Byte.
Max. No. of Communication Attempts	Number of repetitions possible with unsuccessful communication.
Result for Error Code	Result number in which the error code is saved. See vicorem reference manual.
Termination when Elements are missing	This option allows the command to be stopped if the elements specified to be transferred are not found. The results or points are not found if they are not adjusted to the correct value before sending, for example because a command failed.

6.5.3 Coordinate Transfomation

ADVICE

Important: The obsolete command "Coordinate Transfomation" should no longer be used for new programs.

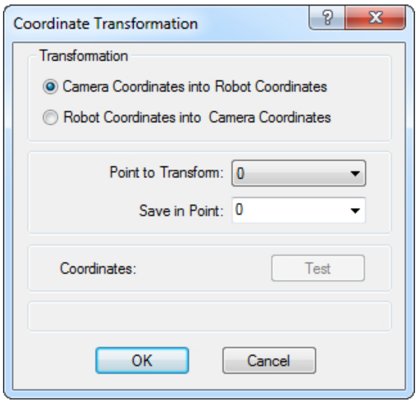


Image 206: Dialogue Coordinate Transfomation

Write a program to convert from robot coordinates to camera coordinates and vice versa in the test program using **Control > Coordinate Transformation**:

- **Camera coordinates** are in image coordinates.
- **Robot coordinates** are in world coordinates.

The Transformation is carried out based on the calibration.

Use this command before sending points to the robot and after receiving points from the robot.

Input Parameters

Transformation	Description
Camera Coordinates into Robot Coordinates / Robot Coordinates into Camera Coordinates	Instruction for which coordinate type should be transformed into which other coordinate type.
Point to Transform	Point number in which the point to be transformed is saved.
Save in Point	Point number that should be used to save the transformed point in.

Test

The test is used to show the transformed coordinates.

6.5.4 Send Measuring Values

ADVICE

Important: The obsolete command "Send Measuring Values" should no longer be used for new programs. Instead of it use the command "Data Transfer".

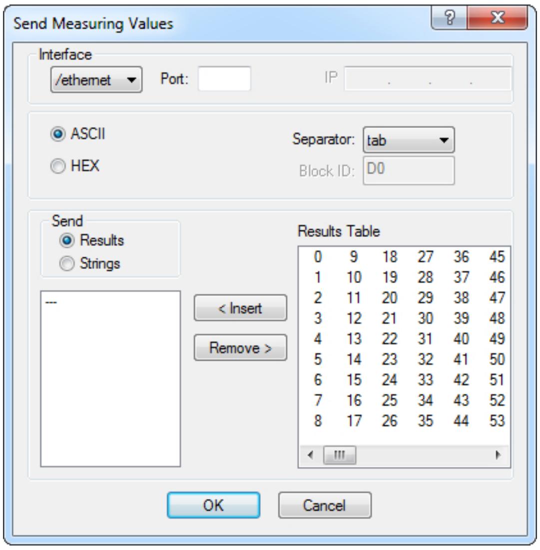


Image 207: Dialogue Send Measuring Values

Write a command to send measured values which have been stored under a result number using **Evaluation > Send Measuring Values**. This way, you can send measured values e.g. from a vision system to a CAQ system.

You can only send to an interface, not to a defined memory location in the file system of the receiver.

Do not use this command together with the send result option of the **Evaluate Result**command.
>> *"Evaluate Result", page 282*

Input Parameters

Send	Description
Results	Results are sent.
Strings	Strings are sent.

Interface	Description
Select one of the available data interfaces, to which the result is to be sent. The interfaces that are available depend on the type of vision system used.	
/com	Serial interface
/ethernet	Ethernet interface
In order to guarantee proper functioning, do not change the entries behind /com and / ethernet unless you are a network technician.	

Character Set

ASCII … The results are output directly to the serial interface in ASCII format. Information separators are sent between the individual results, a “new-line” character is sent at the end. No answer is expected.

ADVICE

A communication participant (e.g. vision system) must start as a TCP server and then it waits for the connection set-up. The other communication participant, as a TCP client, then sets up a TCP link to the server. Data can be sent in both directions when a connection exists.

The Interface »Character Set > ASCII«	Meaning of the ASCII setting
/ethernet without port and without IP	All vision systems are TCP servers on port 23.
/ethernet with port and IP	All vision systems are TCP clients on the set port.

HEX … The results are output directly to the serial interface as hexadecimal long values (4 bytes) in block format. No answer is expected.

The block format corresponds to the communication format in remote control mode.

The various data blocks must be identified by different block identification keys. The identification keys (d0 - df) are reserved for data acceptance in the vcwin communication server.

The Interface »HEX«	Meaning of the HEX setting
/ethernet without port and without IP	All vision systems are TCP servers on port 8500.
/ethernet with port and IP	All vision systems are TCP clients on the set port.

HEX to I/O Box … The results are output via the serial interface as hexadecimal long values (4 bytes) in block format. After sending, the command waits for confirmation from the I/O box.

The block format is a remote control format amended for the I/O box (e.g. pictor Datasave).

The various data blocks must be identified by different block identification keys. The identification keys (d0 - df) are reserved for data acceptance in vcwin communication server.

Transfer List

In the left part of the dialog box is a list of results and strings to be sent. Results and strings will be sent in the order in which they appear in the list. Firstly, an element in the result structure is selected from the field to the right of the dialog box using the mouse. The selected result is then inserted in the transfer list by clicking on the [< Insert] button. The [Remove >] button can be used to remove entries from the transfer list.

Test

In order to see the values sent, connect a terminal to the serial interface of the vision system. Use the identical connection set-up parameters (9600, 8, no parity, 1 stop bit, no handshake) that were used for connecting with vcwin.

6.5.5 Robot Communication

WARNING

Important: The obsolete command "Robot Communication" should no longer be used for new programs.

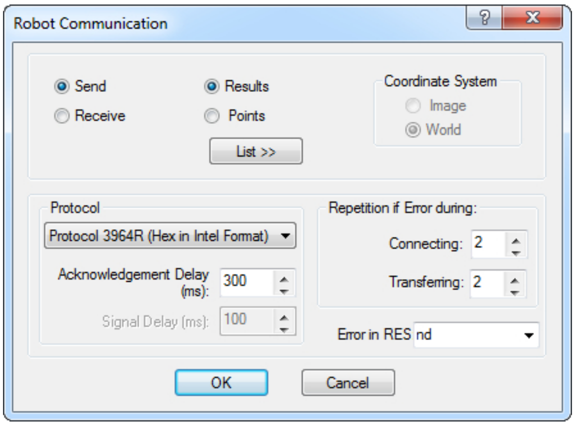


Image 208: Dialogue Robot Communication

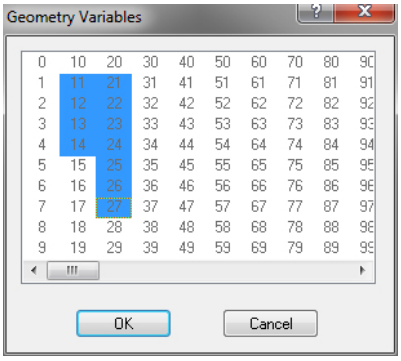
Insert commands for the communication with robots into the test program using **Control > Robot Communication**. The choice is available between sending and receiving points and results. The command allows the choice between and setting of robot-specific protocols.

Brief Guide

1. Select the type of operation **Send** or **Receive**.
2. Select between the sending/receiving of **results** or **points** from the **elements list**.
3. Select and configure the protocol required.
4. Insert the command into the test program with the [OK] button.

Input Parameters

Parameters	Description
Send/Receive	Type of operation for the robot communication.
Results/Points	Select the elements to send/receive from the list. The list of elements to be selected can be opened with the [List >>] button.



The contents of the marked element list is sent, beginning with the smallest number or the elements are set to the values received.

Co-ordinate system	Description
Image/World	Receiving in image or world coordinates.

Protocol	Description
Types of protocol	The 3964R Hex protocol in the Motorola and Intel formats are available.
Acknowledge-ment Delay [ms]	Wait time that is set for receiving the acknowledgement signal from the robot during sending/receiving.
Signal Delay [ms]	Wait time that is set for the acknowledgement signal from the robot when receiving results/points. The standard setting of 300 ms for the acknowledgement delay and 100 ms for the signal delay must be adjusted to correspond to the speed of the communication partner.

Repetition, when Error During	Description
Connecting	Number of repetitions permissible with an unsuccessful connection set-up.
Transferring	Number of repetitions permissible with unsuccessful transfer or with Timeout when receiving an acknowledgement or characters.

7 PART 3 - WORKING WITH THE SOFTWARE

7.1 Using the Web Server

7.1.1 Synchronized sending of images

Images can be sent from the test program to the web interface by means of a command. The images are buffered on the Web Server.

The image buffer memory must be allocated to the web server. This can reduce the number of available image memory pages. The buffers must be configured before images are buffered. If the buffer configuration is changed, all buffered images will be deleted.

Properties

- The images are buffered on the Web Server, the test program does not wait.
- Different images with different identifiers can be sent.
- The web server can buffer a history of the last images for each identifier.
- When the buffer is full, the oldest image will always be overwritten.
- Additional information (metadata) can be assigned to each image. These are also accessible from the WebHMI.

Procedure: Using the image buffers

1. Assign memory
   - Call up the command **Web Server Settings** in the **System Preferences** menu.
   - Assign the desired amount of RAM to the web server as a buffer in the dialogue under Memory allocation.
   - Confirm with [OK].
   - Restart the vision system to activate the web server with the new buffer size.
2. Initialize the image buffer
   - At the beginning of the test program, insert the command **Control > Initialize Image Buffer**.
   - Define in the command all buffers to be used in the program:
     e.g. Live Cam 1, Live Cam 2, Error Cam 1, Error Cam 2
   - Assign the cameras to the buffers whose images are to be stored in the buffer.
   - Configure the buffer depths for the different buffers.

   ADVICE

   The dialogue automatically updates the memory consumption if there is a connection to the vision system. If memory consumption is displayed in red, the buffer allocated to the Web Server is not sufficient for the buffer configuration.
3. Copy image into buffer
   - Insert the command **Control > Copy image into Buffer** into the test program. The command can be inserted at any position and multiple times into the test program.
   - Select the buffer in which you want to copy the image.
   - Select the image memory page you want to copy from.
   - Optionally, you can attach additional information that should be sent with the image.

7.2 Using the WebHMI

Conditions

The web server integrated in the machine vision systems makes it possible to deliver live images and process data via web interface.

The webHMI (webhmi.html) can be found in the vcwin installation folder. It contains a pre-programmed page that can be configured.

Basics

- Configuration is in json format.
- Use an editor with Syntax Highlighting and Checking to easily detect syntax errors (for example: Visual Studio Code).
- The config must be stored on the device or locally as <pagename>.json together with the webHMI website named <pagename>.html.
- The web pages are uploaded to the vision system with vcwin at **System Preferences > System Resources / Initial Program > Web Pages**. The page can then be accessed via http://<IP address of the vision system>/user-pages/<name of the webpage>.
- By storing the files logo.png and favicon.ico own logos and icons can be used.
- Several HMIs with different names can be stored on one device.
- Sample configuration with associated test programs for some typical scenarios can be found in the examples folder.

7.2.1 Structure

The configuration consists of the version and the 3 sections:

```
{
   "version": "2",
   "global": {
   },
   "systems": [
   ],
   "menu": [
   ]
}
```

Section	Description
version	Used version Used to check if the version of the config matches the version of the webHMI page. The version should be taken from the template in the installation path of the operating software.
global	Options for the whole page, e.g. colors
systems	Description of the referred vision systems with images and inspections on these vision systems
menu	List of the offered menu items and configuration of the corresponding pages

7.2.1.1 Section global

Example

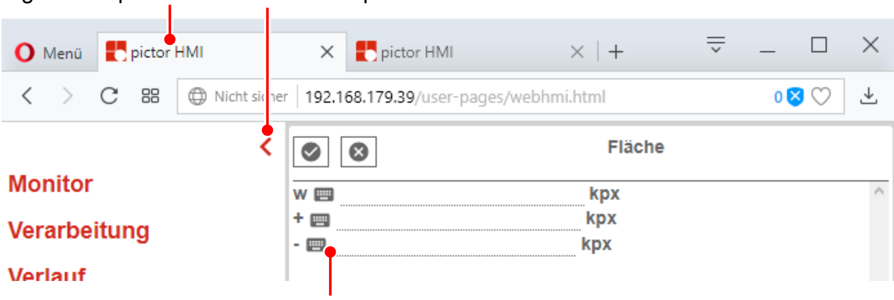
```
"global": {
         "pageTitle": "pictor HMI"
         "renderMethod": 0,
         "logLevel": "debug",
         "menuCollapsed": false,
         "keyboardVisibility": "visible"
```

```
        "menuWidth": "220px",
        "menuLineHeight": "150%",
        "menuBackgroundColor": "white",
        "menuFontColor": "#d52b20",
        "menuHoverFontColor": "#a92219",
        "menuActiveBackgroundColor": "#d52b20",
        "menuActiveColor": "#FFF",
        "menuFontFamily": "Arial, Helvetica, sans-serif",
        "menuFontSize": "18px",
        "menuFontWeight": "bold",

        "tableIdentifierWidth": "400px",
        "baseBackgroundColor": "grey",
        "formBackgroundColor": "white",
        "formFontColor": "#666",
        "formFontFamily": "Arial, Courier new",
        "formFontSize": "14px",
        "formFontWeight": "bold",
    },
```

Parameter

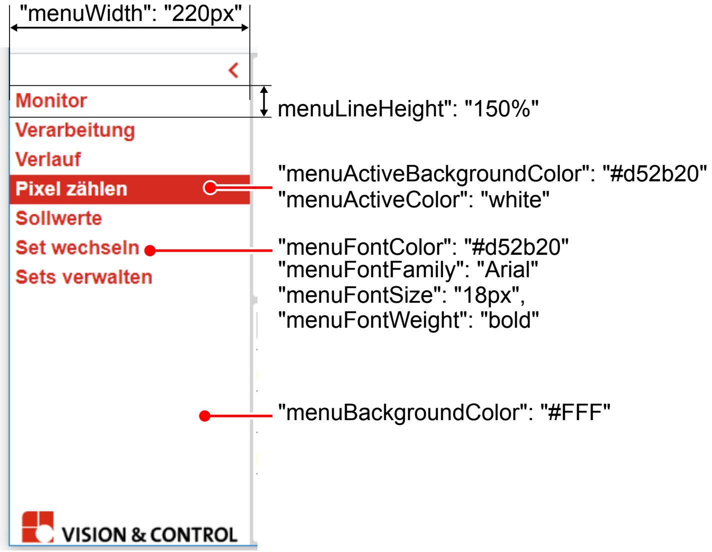
"pageTitle": "pictor HMI" "menuCollapsed": false



"keyboardVisibility": "visible"

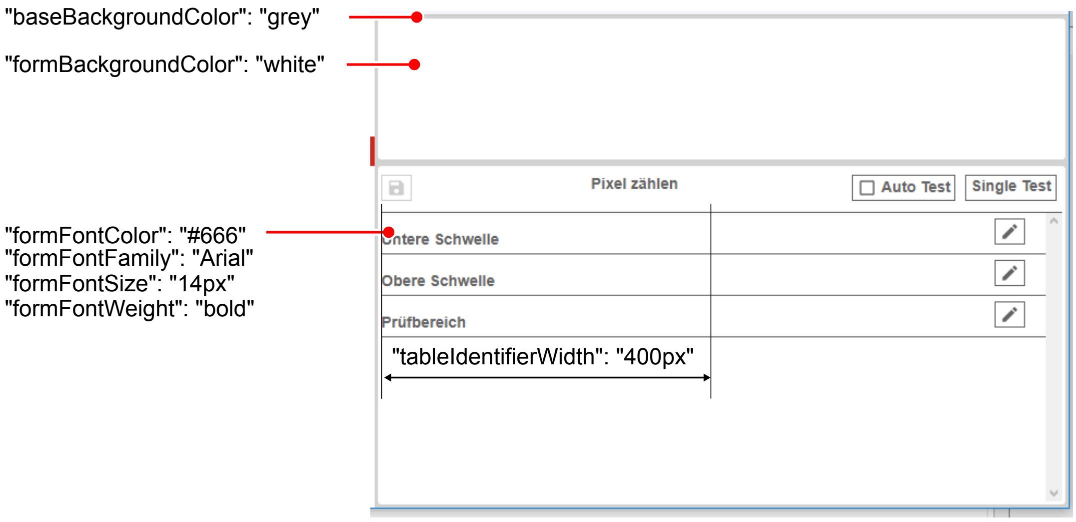
Parameter	Description
pageTitle	Name of the web page displayed in the browser.
renderMethod	renderMethod changes the display method. • 0 - Auto (Default) uses method 2 with Internet Explorer and method 1 in other browsers • 1 - Rendering via offscreen canvas • 2 - Rendering via HTMLImage and Blob This method can trigger memory leaks on older browsers (e.g. Siemens TIA V13). • 3 - Rendering via HTMLImage and DataURI
logLevel	Log level with which the webHMI sends errors. • none - no logs • error - critical errors are logged, e.g. Configuration error (recommended for productive operation) • warn - Errors that occur during operation are logged, e.g. configured image buffers and parameters that are not present on the system • info - short-term faults that occur during operation are logged e.g. lost images due to fast clock / too slow network connection or webHMI processing speed (recommended for commissioning) • debug - additional Reports, z.B. Browser error messages (recommended for investigating problems and submitting support requests) • verbose - all messages (provided for developers)
menuCollapsed	Specifies whether the menu is expanded false (default) or collapsed true at startup.
keyboardVisibility	Shows visible or hides hidden (default) the keyboard button on the inputs fields.

Parameters for layout, color and font of the menu



Parameter	Description
menuWidth	Width of the menu • in percent, e.g. "20%" as proportional width of the browser window. When changing the browser window size, the menu width is scaled. • in pixels, e.g. "200px" as fixed size. Changing the browser window size will not scale the menu width.
menuLineHeight	Line height of the menu items • relative to the selected font size e.g. "normal", "1.5", "150%" • fixed height e.g. "50px"
menuBackgroundColor*	Background color of the menu panel
menuFontColor*	Font color of the menu items
menuHoverFontColor*	Font color of the menu items with mouse-over
menuActiveBackgroundColor*	Background color of the selected menu item
menuActiveColor*	Font color of the selected menu item
menuFontFamily	Font of menu items, e.g. "Arial, Helvetica, sans-serif" The fonts can be specified as a list separated by a comma. The first browser-supported font is then used. Fonts are not reloaded.
menuFontSize	Font size of menu items, e.g. "18px"
menuFontWeight	Font weight of menu items, e.g. bold, normal, lighter, 100, 200, 900
* Colors are configured as html colors. CSS color codes like "#d52b20", "#FFF" or "red" are allowed.	

Parameters for layout, color and font of the pages



Parameter	Description
tableIdentifierWidth	Width of the fields for identifiers in the forms (e.g. parameters in ParameterAdjustment or ParameterSet-Name in ParameterSetActivation)
baseBackgroundColor*	Color of the outer background of the webHMI pages or the Overview page.
formBackgroundColor*	Background color of webHMI pages
formFrontColor*	Font color of the webHMI pages
formFontFamily	Font of the webHMI pages, e.g. "Arial, Helvetica, sans-serif" The fonts can be specified as a list separated by a comma. The first browser-supported font is then used. Fonts are not reloaded.
formFontSize	Font size of menu items, e.g. "18px"
formFontWeight	Font weight of menu items, e.g. bold, normal, lighter, 100, 200, 900
* Colors are configured as html colors. CSS color codes like "#d52b20", "#FFF" or "red" are allowed.	

7.2.1.2 Section systems

Example

```
"ip": "192.168.3.7",
"id": "pictor",
"imageBuffer": [
],
"inspections": [
],
```

Parameters

Parameter	Description
ip	IP and port of the vision system. If the string is empty, the source of the web page is used.
id	Identifier with which the vision system is referenced in the WebHMI.
imageBuffer	Name and configuration of the images to be received by the WebHMI.
inspections	Definition of the inspections on the vision system to be displayed or configured in the WebHMI.

7.2.1.2.1 Section imageBuffer

Example

```
"imageBuffer": [
        {
            "id": "buffer0",
            "displayName": "Kamera 1",
            "bufferName": "cam0",
            "historyLength": 1,
            "transmissionParameters": {
            }
        },
        {
            "id": "buffer2_defect",
            "displayName": "Fehlerbilder Kamera 3",
            "bufferName": "cam2_defect",
            "historyLength":10,
            "transmissionParameters": {
                "imageDownScale": 2,
                "imageFormat": "jpg",
                "imageQuality": "80"
            }
        },
    ],
```

Parameters

Parameter	Description
ID	Identifier of the buffer with which the buffer is referenced in the pages and inspections.
displayName	Name displayed as image name for images of this buffer.
bufferName	Name of the buffer on the vision system. The name corre-sponds to the name used in the vcwin test program for this buffer.
historyLength	Length of the history stored in the browser and displayed in pages of type HistoryDisplay. The length can be configured to be longer or shorter than the length configured on the vision system.
transmissionParameters	Enables the specification of additional optional transmission parameters. *see "transmissionParameters", page 405*

transmissionParameters

With these parameters the transmission can be adapted e.g. to adapt very high resolution images to the display size before the transmission and thus to save transmission time and computing power.

Parameter	Description
imageDownScale	Scales the image before transmission by the specified factor (default = "1").
imageFormat	Defines the format in which the image is transferred. Uncompressed as bitmap "bmp" (default) or compressed in jpeg format "jpg".
imageQuality	Defines the quality of the image transfer in the "jpg" format (default = 80).

7.2.1.2.2 Section inspection

Example

```
"id": "camera",
"imageBuffer": "result",
"setpoints": [
    ],
"parameters": [
    ],
"actions": [
    ],
"displayName" : "Name der Inspektion"
```

Parameters

Parameter	Description
ID	Identifier with which the inspection is referenced in the webHMI.
imageBuffer	Identifier of the image buffer with the image belonging to the inspection.
setpoints	Setpoints of the function, *see "Section inspection - setpoints", page 406*
parameters	Setting parameter of the inspection, *see "Section inspection - parameters", page 406*
actions	Test, teach or mode functions which should be available when setting the inspection. They provide actions on the setting pages and allow parameters to be set temporarily in order to call the corresponding branch in the test program. *see "Section inspection - actions", page 407*
displayName	Name of the inspection displayed on corresponding pages in the toolbar.

7.2.1.2.2.1 Section inspection - setpoints

Example

{
"type": "NominalTolerance",
"displayName": "Länge",
"conversionFactor": "0.001",
"unit": "mm",
"nominalParameterId": "LengthNominal",
"lowerParameterId": "LengthTolPlus",
"upperParameterId": "LengthTolMinus"
}

Parameter

Parameter	Description
type	Type of the Setpoint
displayName	Name with which the setpoint is displayed in the webHMI.
unit	Only for parameters with type Result. Unit displayed after the value in the webHMI.
conversionFactor	Conversion factor: Parameter values for the user interface are multiplied by that conversion factor. e.g.: factor of 0.001 converts a given value in [µm] of the vision system to [mm] in the webHMI.
maxParameterId minParameterId nominalParameterId lowerParameterId upperParameterId	Identifier of the parameters on the vision system with which the setpoints are configured.

7.2.1.2.2.2 Section inspection - parameters

Example

{
"displayName": "Belichtungszeit",
"parameter": "Shutter",
"unit": "ms"
}

Parameter

Parameter	Description
displayName	Displayed name of the parameter in the webHMI.
Parameter	Identifier of the parameter in the parameter set on the vision system.
unit	Only for parameters with type Result. Unit displayed behind the value in the webHMI.

7.2.1.2.2.3 Section inspection - actions

Example

```
{
"type": "Check",
"displayName": "Auto Test",
"parameter": "AutoTest",
"selectedValue": "1"
}
```

Parameter

Parameter	Description
type	Sets the type of the action
displayName	Designation that is displayed on the button resp. on the check box in the webHMI.
Parameter	Identifier of the parameter on the vision system that will be set by the action.
selectedValue	Value that is written to the parameter when the action is triggered.

Type check

- sets the vision system in a specific mode, e.g. "Autotest" or "Live Image"
- is displayed as a checkbox
- when activated, the specified parameter is set to the specified value (selectedValue)
- when deactivated, the specified parameter is set to "0"
- one vision system parameter can be assigned several "check" actions with different "selectedValue"s

Type button

- triggers a one-time action, e.g. "Teach-in pattern", "Single test" or "Image acquisition"
- is displayed as a button
- when activated, the specified parameter is set to the specified value (selectedValue)
- in the test program, the value must be reset to "0" before the button can be clicked again (command: **Evaluation > Write back Parameter**)
- the button remains active until resetting by the vision system

7.2.1.3 Section menu

In the section "menu" the visible menu items of the webHMI are created and the existing pages are configured.

A list of pages can be entered. The pages are available in the webHMI as menus in the entered order.

The possible parameters of the pages depend on the respective page type.

Page types

- OverviewDisplay
- ImageDisplay
- HistoryDisplay

- ParameterSetManagement
- ParameterSetActivation
- ParameterAdjustment
- SetPointValue

7.2.1.3.1 Page type OverviewDisplay

The page type "OverviewDisplay" displays multiple images at once in a configurable arrangement.

Example

```
{
   "type": "OverviewDisplay",
   "menuTitle": "Monitor",
   "imageBuffer": [
         {
             "id": "pictor/result01",
             "top": "25%",
             "left": "2%",
             "width": "46%",
             "height": "50%"
         },
         {
             "id": "pictor/result02",
             "top": "2%",
             "left": "52%",
             "width": "46%",
             "height": "46%"
         }
   ]
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
imageBuffer	List of image buffers to be displayed in the page. Configuration of the image "<system id>/<buffer id>" e.g.: "pictor/result01" stands for the image "result01" on the vision system "pictor". The IDs must match the IDs configured in the section "systems": [ ].
layout	Configuration of the position and size of the display of the different images. For each image to be displayed, the position and size must be configured. The position and sizes can be specified in percent (%) or in pixels (px). If configured as a percentage, this refers to the displayed page area and the display automatically scales with the window size. Percentages are indicated by a % character after the value (e.g. "45%"). If configured in pixels, the size is fix and is not scaled when the page size is changed. Specifications are made by a "px" after the value (e.g. "300px").

7.2.1.3.2 Page type ImageDisplay

The page type "ImageDisplay" displays an image from an image buffer, and optionally its metadata.

Example

```
{
   "type": "ImageDisplay",
   "menuTitle": "Aktuelles Bild",
   "imageBuffer": "pictor/result",
   "metaData": true,
   "metaDataHeight": "20%"
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
imageBuffer	Buffer ID of the image to be displayed in the page. Configuration: "<system id>/<buffer id>" e.g.: "pictor/result" stands for the image "result" on the vision system "pictor". The IDs must match the IDs configured in the section "systems": [ ].
metaData	Determines if the metadata of the image is displayed. true – display false – not display
metaDataHeight	Height of the display field for metadata. Specified as proportional height (e.g. "20%") or in pixels (e.g. "200px").

7.2.1.3.3 Page type HistoryDisplay

The page type "HistoryDisplay" displays the history of an image buffer (e.g. display the last error image from a buffer for error images).

The length of the history for the buffer is set in the imageBuffer area.

Example

```
{
   "type": "HistoryDisplay",
   "menuTitle": "Fehlerbilder Kamera",
   "imageBuffer": "pictor/result",
   "metaDataHeight": "20%"
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
imageBuffer	Buffer ID of the image to be displayed in the page. Configuration: "<system id>/<buffer id>" e.g.: "pictor/result" stands for the image "result" on the vision system "pictor". The IDs must match the IDs configured in the section "systems": [ ].
metaDataHeight	Height of the display field for metadata. Specified as proportional height (e.g. "20%") or in pixels (e.g. "200px").

7.2.1.3.4 Page type ParameterSetManagement

The page type "ParameterSetManagement" lists the parameter sets of a vision system.

The page type allows changing the active parameter set, deleting and creating parameter sets as well as changing the comment and PLC ID of the parameter sets.

Example

```
{
   "type": "ParameterSetManagement",
   "menuTitle": "Sets verwalten",
   "system": "pictor"
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
system	Identifier of the vision system on which the parameter set can be changed.

7.2.1.3.5 Page type ParameterSetActivation

The page type "ParameterSetActivation" lists the parameter sets of a vision system.

The page type allows changing the active parameter set.

Example

```
{
   "type": "ParameterSetActivation",
   "menuTitle": "Set wechseln",
   "system": "pictor"
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
system	Identifier of the vision system on which the parameter set can be changed.

7.2.1.3.6 Page type ParameterAdjustment

The page type "ParameterAdjustment" automatically displays all parameters and actions of an inspection and the image assigned with the inspection.

The page type allows setting the parameters of an inspection.

Example

```
{
   "type": "ParameterAdjustment",
   "menuTitle": "Pixel zählen",
   "inspection": "pictor/count pixels",
   "editHeight": "50%"
   "testAreaEditColor": "#99FF00"
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
inspection	Configuration of the inspection "<system id>/<inspection id>" e.g.: "pictor/count pixels" stands for the "count pixels" inspection on the vision system "pictor"
editHeight	Height of the edit field for parameters in the form. Specified as proportional height (e.g. "20%") or in pixels (e.g. "200px").
`testAreaEditColor`	Color with which the test areas are displayed in the image by the graphical editor (optional). Colors are configured as color name e.g. "yellow", rgb value e.g. "rgb (255,255,0)" or HTML color code (RGB) e.g. "# FFFF00". The default value is red.

7.2.1.3.7 Page type SetPointValue

The page type "SetPointValue" automatically displays all setpoints of an inspection.

The page type allows to configure the setpoints of an inspection.

Example

```
{
   "type": "SetPointValue",
   "menuTitle": "Sollwerte",
   "inspection": "pictor/count pixels"
}
```

Parameter

Parameter	Description
type	Sets the page type.
menuTitle	Designation of the page in the menu.
inspection	Configuration of the inspection "<system id>/<inspection id>" e.g.: "pictor/count pixels" stands for the "count pixels" inspection on the vision system "pictor"

7.3 Using Webpages

General information

If no page is specified, monitor.html is shown.

To ensure compatibility with future versions, you should always explicitly request monitor.html in HMI's explicitly.

7.3.1 Vision & Control Monitorpage

Description

This page displays the full-screen display of the current display page and the error status.

http://192.168.3.180/monitor.html

Parameters

ADVICE

When entering the parameters in the address line, make sure that they are spelled correctly (even upper-case and lower-case).

The parameters can be set using a query string. The string is introduced with the question mark (?). The parameter-value blocks are separated by a commercial & character. The parameters and values are separated by an equal sign =.

http://192.168.3.180/monitor.html/**?**target=192.168.3.180**&**showfps**=**true

target

- With target you can select another target system than the source of the website.
- IP and port can be specified as values.

showOverlay

- showOverlay switches the display of the overlay on or off.
  - true display is activated (default)
  - false display is deactivated

renderMethod

- renderMethod changes the display method.
  - 0 - Auto (Default) uses method 2 with Internet Explorer and method 1 in other browsers
  - 1 - Rendering via offscreen canvas
  - 2 - Rendering via HTMLImage and Blob
    This method can trigger memory leaks on older browsers (z.B. Siemens TIA V13).
  - 3 - Rendering via HTMLImage and DataURI

imageDownScale

- imageDownScale requests the image in the specified scale 1:n. With this setting, bandwidth can be reduced and a faster display can be achieved.
- The overlay is still transmitted in full resolution.

pageNo

- pageNo requests the specified image memory page instead of the current display page (-1).

imageFormat

- imageFormat defines the used image format "bmp" (default) or "jpg".

imageQuality

- imageQuality defines the quality of the image transfer for "jpg" format (default=80).

showfps

- showfps switches the display of Frames per Second on or off. Use showfps to determine the appropriate parameters for the transmission settings.
  - true display is activated
  - false display is deactivated

7.3.2 Image Buffer Monitor

Description

This page displays the full-screen display of the most recent image of an image buffer. If no special image buffer is configured, the first buffer is displayed. With multiple buffers, the determination of the first buffer depends on the software version and browser.

http://192.168.3.180/buffer-monitor.html

Parameters

ADVICE
When entering the parameters in the address line, make sure that they are spelled correctly (even upper-case and lower-case).

The parameters can be set using a query string. The string is introduced with the question mark (?). The parameter-value blocks are separated by a commercial & character. The parameters and values are separated by an equal sign =.

http://192.168.3.180/buffer-monitor.html?imageName=cam

target

- With target you can select another target system than the source of the website.
- IP and port can be specified as values.

showOverlay

- showOverlay switches the display of the overlay on or off.
  - true display is activated (default)
  - false display is deactivated

renderMethod

- renderMethod changes the display method.
  - 0 - Auto (default) uses method 2 with Internet Explorer and method 1 in other browsers
  - 1 - Rendering via offscreen canvas
  - 2 - Rendering via HTMLImage and Blob
    This method can trigger memory leaks on older browsers (z.B. Siemens TIA V13).
  - 3 - Rendering via HTMLImage and DataURI

showdroped

- With showdroped a counter is inserted in the lower left image corner which displays the number of lost frames.
- Frames get lost if the vision system writes images to the buffer faster than they can be received and displayed by the buffer monitor. Use the display to check for lost frames and, if necessary, optimize the transfer settings so that each image is displayed.
  - false - counter is not displayed (default) ° - counter is displayed
  - true - counter is displayed

imageDownScale

- imageDownScale requests the image in the specified scale 1:n. With this setting, bandwidth can be reduced and a faster display can be achieved.
- The overlay is still transmitted in full resolution.

imageFormat

- imageFormat defines the used image format "bmp" (default) or "jpg".

imageQuality

- imageQuality defines the quality of the image transfer for "jpg" format (default=80).

imageName

- imageName displays the selected image buffer.

7.4 Additional functions

7.4.1 ROI on cameras

ADVICE
This feature is not available in the simulator and for line-scan cameras.

By configuring an ROI / AOI (Region / Area of Interest), the captured camera image can be reduced to relevant areas. The ROI can be configured dynamically in the program or statically for connected cameras.

The dynamic configuration takes place directly in the test program in the respective commands (for example Grab Image). This method creates a full-resolution image memory page and then creates the ROI on it. This requires more RAM and leads to a display of the test image with a frame on connected monitors or the Web Monitor.

If no dynamic ROI is required, a camera with a static ROI can be created using a configuration file. The file will be read and used at the start of the vicosys. The respective camera receives the adjusted image section and acquires only in this section and transmits it. The static ROI is configured before the vicosys starts and can not be changed at runtime in the test program. This method requires less RAM and prevents the display of the test image with an frame.

Configuration

The configuration is done via a file in JSON format. This file must be loaded into the program directory of the vicosys. After the upload a reboot of the device is necessary.

The file name must be camera_aois.json.

In the vcwin installation directory "examples / static camera aoi" is a configuration file as an example.

Structure

```
{
"cam1" : {
"Vendor" : "Basler",
"Model" : "acA1300-22gc",
"Serial" : "21722365",
"PX" : "346",
"PY" : "182",
"DX" : "600",
"DY" : "600",
},
"cam2" : {
"Vendor" : "MATRIX VISION GmbH",
"Model" : "mvBlueCOUGAR-X104eG-POE",
"Serial" : "*",
"PX" : "200",
"PY" : "100",
"DX" : "800",
"DY" : "200",
},
"cam3" : {
"Vendor" : "V&C",
"Model" : "pictor N420M",
"Serial" : "*",
"PX" : "400",
"PY" : "300",
"DX" : "800",
"DY" : "600",
},
}
```

Parameter	Description

`cam1`
`cam2`
`...`
`camX`
	The label can be chosen freely, but must be valid for JSON (must start with a letter, may not contain ', ", \)

`Vendor`
`Model`
`Serial`
	The information must be identical to the information given by the corresponding camera.
	Which cameras are connected to the vicosys can be determined with **Utilities for vicosys > Sort Cameras**, button [Read].
	By using `"*"` this parameter is ignored so that e.g. all cameras of a type are addressed regardless of the serial number.

`PX` `PY`	Starting point of the ROI.
`DX` `DY`	Size of the ROI. This size then corresponds to the resolution of the camera at the vicosys.

8 PART 4 - APPENDICES

8.1 Function and Command Reference (Firmware Overview)

Function Reference Menu > Utilities

Function	Subfunction / Option	pictor M	pictor T	vicosys
Online Debugging		4.22.128	4.16.230	4.16.213

Function Reference Menu > Communication

Function	Subfunction / Option	pictor M	pictor T	vicosys
System Resources / Initial Program	Set Password	-	4.16.239	4.16.239
	Save on PC	-	4.16.239	4.16.239
	Load to Vision System	-	4.16.239	4.16.239
Make Data Backup		-	4.16.239	4.16.239
Restore Data Backup		-	4.16.239	4.16.239
Display Calibration Data		-	4.16.239	4.16.239
Calibrate Camera	Export Calibration Data	-	4.16.239	4.16.239
	Import Calibration Data	-	4.16.239	4.16.239
White Balance	Export White Balance Data	-	4.16.243	4.16.239
	Import White Balance Data	-	4.16.243	4.16.239

Function Reference Monitor Window

Function	Subfunction / Option	pictor M	pictor T	vicosys
Overlay	multicolored Overlay	-	4.16.245	4.16.245

Command Reference

Command	Option	pictor M	pictor T	vicosys
Image Commands				
False Colors	HSI Mode	-	-	4.16.213
GenICam Register	for GigE area cameras	-	-	4.16.220
	for GigE line scan cameras	-	-	4.16.239
Utilities for vicosys	Set Resolution without reboot	-	-	4.16.200
	Set Contrast Stretching	-	-	4.16.220[1]
Locate Commands				
360° Pattern Search		-	4.16.230	4.16.175
Data Matrix Code	Advanced Read Function	4.18.98[2]	-	-
Edge Based Object Search		4.20.113[3]	-	-
Measure Temperature		-	-	4.16.251[1]
Locate Helix		-	4.16.250	4.16.250
Locate Angle		-	4.16.240	4.16.240
Evaluation Commands				
Check Point Distances		-	4.16.250	4.16.250

Command	Option	pictor M	pictor T	vicosys
Calculation Script (Ruby)		-	4.16.230	4.16.213
Control Commands				
Asynchronous Processes	for GigE cameras	-	-	4.16.241[4]
	for FireWire cameras	-	-	4.16.224[4]
Save/Load Image	from String	-	4.16.251	4.16.251
Process Module	CANopen, Sercos	-	4.16.233	4.16.233
	Modbus	4.23.130	4.16.230	4.16.230
	Modbus: limit values to a 16 bit register	4.23.131	4.16.239	4.16.239
Save Process Data		-	4.16.251	4.16.251

[1] License Thermal imaging functions required
[2] License Data matrix code required
[3] License Edge-based object search required
[4] License Asynchronous processes required

8.2 Principles of Detection

Basics

The term **detection** comes from contact based measurement technology and was used to refer to obtaining measurement values by direct physical contact with the test object. Today, this term has been adopted to refer also to contact-less measurement procedures for the recognition of edges in images.

Synonyms for detect are “investigate” and “locate”. The results of detection are either points or contours (a contour is a series of points). These results are stored temporarily in the point structure and contour buffers respectively.

Positioning Options

An important aspect of detection is the ability to compensate for variable object position within the image. There are several basic methods of positioning tracking beams:

- Absolute positioning
- Relative positioning - follow a point
- Relative positioning - follow a line

Dependent upon the quality of the image (i.e. the transitions between bright and dark), the mathematical algorithms used for detection can become very complex. For this reason, vision systems from Vision & Control GmbH offer a choice of several methods. This ensures a flexible response to varied conditions in the real world and thus maximizes the quality of results.

Overview of Detection Algorithms

- *"Binary Detection", page 418*
- *"Gray Value Detection", page 418*
- *"Gradient method", page 419*
- *"Convolution Methods", page 420*
- *"Sub-Pixel Detection", page 421*
- *"Helix Detection (optional)", page 422*
- *"Contour Detection", page 422*

8.2.1 Binary Detection

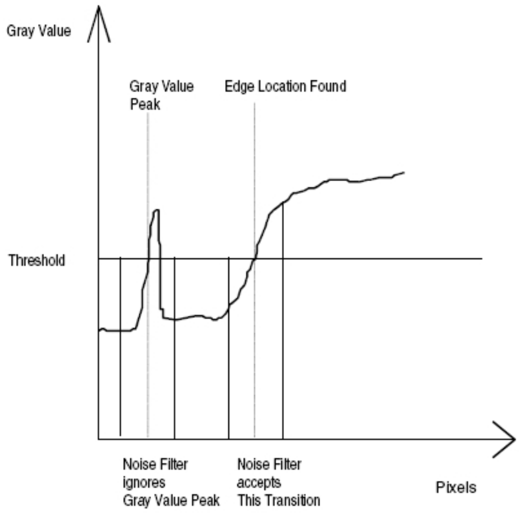
Parameters

- Gray value threshold
- Noise filter

The binary detection algorithm makes use of a simple **gray value threshold** and a noise filter to locate transitions in an image.

The noise filter ensures that spurious transitions e.g. dust are ignored. If the threshold is crossed more than once within the space of a few pixels, this is ignored and does not qualify as a true edge.

Binary detection is used for well-lit, high-contrast objects where speed is the dominant consideration (preferably transition detected).



Advantage

- Fast

Disadvantages

- Highly dependent upon lighting conditions
- No two-dimensional noise reduction

Attainable precision using Gradient with Threshold is usually ± 1 pixel, dependent upon the object.

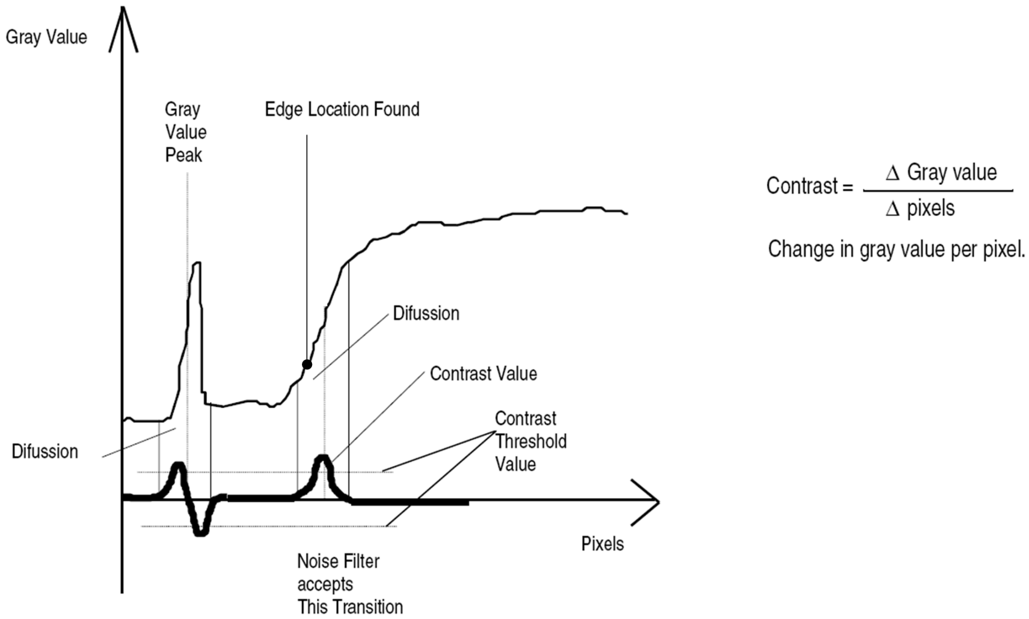
8.2.2 Gray Value Detection

Parameters

- Contrast
- Diffusion
- Noise filter

This detection method does not make use of a static threshold but relies on there being enough **contrast**. For a transition to be recognized, the gray value must change a minimum amount within a certain number of pixels (defined by the **diffusion** parameter).

The algorithm contains a **noise filter** to ensure that spurious peaks or troughs are ignored. To be accepted, the required gray value change must be maintained over the width of the noise filter. The process can also be used with vertical illumination.



Advantages

- Fast
- Fairly independent of lighting conditions

Disadvantage

- No two-dimensional noise reduction

Attainable precision using gray value detection is usually ± 1 pixel, dependent upon the object.

8.2.3 Gradient method

The gradient algorithms were developed as a scanning method specifically with the aspects of industrial image processing in mind. They link high scanning reliability with high scanning precision using two-dimensional interference suppression.

With the gradient method, the greyscale values of several neighbouring image points are offset against one another, thereby determining the greyscale value gradient. To achieve two-dimensional interference suppression, in addition to binary and greyscale value scanning, the image points along a scan beam as well as neighbouring image points of a scan window are offset against one another (**Scan Width** parameter).

Types of scanning

The history of the gradient can be evaluated mathematically in different ways. For the range of vision systems from Vision & Control GmbH, the following options were selected:

- Edge search until a gradient threshold is reached
- Search for the maximum gradient along a search area (maximum edge probability if only one edge is expected)

Gradient with threshold

Parameter

- Threshold (0…255)
- Scan width
- Other parameters, such as Diffusion and Noise Filter, can be set for some commands.

Advantage

- Two-dimensional interference suppression
- Minimal lighting dependency

Disadvantage

- When using two-dimensional interference suppression, somewhat slower than the greyscale value method

The precision that can be achieved depends on the object, generally approx. ±1 pixel.

Maximum gradient

ADVICE

Since a maximum gradient is always found with this method, regardless of the image information, you must make a concrete statement on whether there really is an edge behind the found gradient using the **Minimum gradient** parameter.

Parameter

- Scan width
- Minimum gradient (0…255, pre-setting in vcwin = 5)
- Other parameters, such as Diffusion and Noise Filter, can be set for some commands.

Advantage

- Edge location is defined mathematically
- Two-dimensional interference suppression
- Minimal lighting dependency

Disadvantage

- When using two-dimensional interference suppression, somewhat slower than the greyscale value method

The edge precision that can be achieved depends on the object, approx. ±0.5 pixels.

8.2.4 Convolution Methods

These detection algorithms have been specially developed for industrial image processing. They are similar to gradient methods, but also include a two-dimensional spectrum analysis.

Detection Methods

There are two types of detection in the Vision & Control GmbH vision systems, which are distinguished according to the post-processing of the specified coefficient gradient:

- Convolution with threshold
- Convolution with maximum coefficient

Convolution with Threshold

Parameters

- Threshold
- Detection width

Advantages

- Two-dimensional noise reduction
- Fairly independent of lighting conditions

Disadvantage

- Slightly slower than gray value detection if 2-dimensional noise reduction is used.

Attainable precision using Convolution with Threshold is usually ± 1 pixel, dependent upon the object.

Convolution Max. Coefficient

Parameters

- Detection Width
- Minimum Coefficient (0…255, default setting in vcwin = 5)

As with the **maximum coefficient** process, a minimum expected value for the edge is preset in the parameter **Minimum Coefficient**.

Advantages

- Position of edge is determined.
- Two-dimensional noise reduction.
- Fairly independent of lighting conditions

Disadvantages

- Slightly slower than gray value detection if 2-dimensional noise reduction is used.

Attainable precision using Gradient with Threshold is usually ± 0.5 pixels, dependent upon the object.

8.2.5 Sub-Pixel Detection

Parameters

- Contrast (for gray value detection)
- Diffusion (for gray value detection)
- Noise filter (for gray value detection)
- Number of detections (for fine analysis)

Advantages

- High precision

Disadvantages

- Calculation intensive
- Noise reduction possible

Attainable precision using sub-pixel detection can be as high as ± 1/10 of a pixel, dependent upon the object.

Procedure … Sub-pixel detection involves two processes:

1 Gray value detection: >> *"Gray Value Detection", page 418*

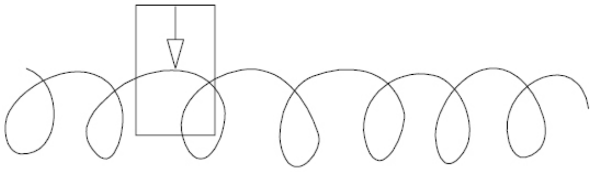
2 Fine analysis … Three areas are investigated during fine analysis:

- Dark area
- Transitions
- Bright area

Number of Detections … The **Number of Detections** parameter is used to further increase the precision of detection. Found edges are used to calculate the actual edge using statistical methods.

8.2.6 Helix Detection (optional)

The helix detection method is used for detecting the highest point of a coiled object e.g. a spring or the element in a light bulb. The process determines the highest point of a coil. Only horizontally or vertically aligned coils are possible.



Wendelantastung

Detection Methods

The object is detected using the gradient method and the peak of a coil in the helix is detected. There are two methods of helix detection:

- “Helix (max.Gradient)” detection
- “Helix (Gradient)” detection

Helix (Max. Gradient)

Parameters

- Minimum gradient
- Detection width
- *>> "Gradient method", page 419*

Helix (Gradient)

Parameters

- Gradient threshold
- Detection width
- *>> "Gradient method", page 419*

8.2.7 Contour Detection

Procedure … The contour of an object is followed from a defined starting point. The scope of the search can be confined to an image window or can be in the complete image.

Closed Contour … A closed contour is one where the starting point is reached again.

The coordinates of all points along a contour are temporarily stored in a “contour buffer”. The system supports up to 10 contour buffers, each containing up to 2000 coordinates. The contents of the contour buffers can then be used by test commands. Complete contour buffers can be used as operands for calculations e.g. to determine the dimensions of a circular object which is larger than the recorded image.

Detection Methods

There are three methods of contour detection:

- Binary method (along an ISO gray value line)
- Gradient method (along the maximum gradients of a gray value line)
- Gradient method with subsequent fine analysis of points along edges (optional)

Binary Contour Detection

Parameters

- Gray value threshold
- Noise filter

The contour of an object is detected using a constant gray value threshold i.e. the contour will be quasi "isometric".

Advantages

- The starting point is sure to be reached again in closed contours.

Disadvantages

- Highly dependent upon lighting conditions
- No noise reduction

Attainable precision can be as high as 1 pixel, dependent upon the object.

Gradient Contour Detection

Parameters

- Threshold
- Detection width
- Min. threshold

The contour of an object is detected using the maximum gradient of gray values around the object.

This method should only be used when the edges are distinct, as the maximum gray value gradients within images of real objects do not always coincide with the required edges.

If this prerequisite is fulfilled, a relatively high precision of about ± 0.5 of a pixel can be attained. Moreover, it is also possible to carry out a fine analysis of the found contour location (see Contour detection with the gradient procedure, with subsequent fine analysis of the found edge location).

The parameters **Threshold** and **Detection Width** are the parameters for detecting the starting point.

The parameter **Min. Threshold** refers to the determination of the contour. The contour determination is aborted if the threshold is undershot, i.e. the determined gradient is too small.

Advantages

- The actual position of the edge is determined.
- Fairly independent of lighting conditions.
- Higher precision than the binary method.

Disadvantages

- Risk of uncertainty when the object contours are not distinct .

Attainable precision using gradient contour detection is generally about ± 0,5 pixel, dependent upon the object.

Contour Detection Using the Gradient Method with Subsequent Fine Analysis

Parameters

- Threshold
- Detection width
- Min. threshold

Procedure … During the first step of this process, contour tracing is carried out according to the procedure described in the “Gradient Contour Detection" topic. This is followed by a fine analysis of the area around the contour (dark area, transition, bright area).

This method should only be used when the edges are distinct, as the maximum gray value gradients within images of real objects do not always coincide with the required edges. If this prerequisite is fulfilled, a relatively high precision can be attained.

The parameters **Threshold** and **Detection Width** are the parameters for detecting the starting point.

The parameter **Min. Threshold** refers to the determination of the contour. The contour determination is aborted if the threshold is undershot, i.e. the determined gradient is too small.

Advantages

- High precision fairly independent of lighting conditions.

Disadvantage

- Risk of uncertainty when the object contours are not distinct.

As in the sub-pixel detection method, attainable precision can be as high as 1/10 of a pixel, dependent upon the object.

8.3 Methods for calculation script

In this chapter, you will find the description of all methods intrinsic to vcwin.

Instance methods are methods which you can apply to each object or instance of an object, whereas class methods create new objects. Constants are values which you can use for comparison to individual attribute values of objects. They contain constant numerical values.

Result class

Method	Description
.value	Value of the result (Integer)
.state	Evaluation of the result (Boolean: true = good, false = bad)
.clone	Copies the object
.to_s	Conversion to string

Instance methods

Method	Description
VC::Result.new(value, state=true)	Defines a new VC::Result object with result value (value) and evaluation (state). If the parameter state is not defined, it is true by default.

Class methods

Point class

Method	Description
.x	Value of the x-co-ordinate (Integer)
.y	Value of the y-co-ordinate (Integer)
.coordinate_system	Co-ordinate system used (Integer: 0 for image co-ordinates, 1 for world co-ordinates)

Instance methods

Method	Description
.camera	Number of the camera (Integer) *Since the calibration of the camera is decisive when converting a point from image to world co-ordinates, you should use this command to identify the right camera.*
.isImage?	Indicates whether the point is in image co-ordinates (Boolean)
.isWorld?	Indicates whether the point is in world co-ordinates (Boolean)
.toImage(camera=-1)	Copies the point to image co-ordinates. If a camera is not defined, the camera of the current image page (-1) is set by default.
.toImage!(camera=-1)	Converts the point to image co-ordinates. If the point is already in image co-ordinates, no conversion is carried out. If the camera parameter is not defined, the camera of the current image page (-1) is set by default.
.toWorld	Copies the point to world co-ordinates.
.toWorld!	Converts the point to world co-ordinates. If the point is already in world co-ordinates, no conversion is carried out.
.clone	Copies the point without changing the co-ordinate system.
.to_a	Conversion to array
.to_s	Conversion to string

Instance methods

Method	Description
VC::Point.image(x, y, camera=-1)	Defines a new VC::Point object in the image co-ordinate system with x-co-ordinate (x), y-co-ordinate (y) and camera used (camera). If the camera parameter is not defined, the camera of the current image page (-1) is set by default.
VC::Point.world(x, y)	Defines a new VC::Point object in the world co-ordinate system with x-co-ordinate (x) and y-co-ordinate (y).

Class methods

Constant	Description
VC::CoordinateImage	Value 0 (Integer)
VC::CoordinateWorld	Value 1 (Integer)

Constants

Lines class

Method	Description
[0/1]	Co-ordinates of the first point [0] or second point [1] of the line
.x1, .x2	Value of the x-co-ordinate of point 1 or 2 (Integer)
.y1, .y2	Value of the y-co-ordinate of point 1 or 2 (Integer)
.coordinate_system	Co-ordinate system used (Integer: 0 for image co-ordinates, 1 for world co-ordinates)

Instance methods

Method	Description
.camera	Number of the camera (Integer) *Since the calibration of the camera is decisive when converting a line from image to world co-ordinates, you should use this command to identify the right camera.*
.isImage?	Indicates whether the line is in image co-ordinates (Boolean)
.isWorld?	Indicates whether the line is in world co-ordinates (Boolean)
.toImage(camera=-1)	Copies the line to image co-ordinates. If a camera is not defined, the camera of the current image page (-1) is set by default.
.toImage!(camera=-1)	Converts the line to image co-ordinates. If the line is already in image co-ordinates, no conversion is carried out. If the camera parameter is not defined, the camera of the current image page (-1) is set by default.
.toWorld	Copies the line to world co-ordinates.
.toWorld!	Converts the line to world co-ordinates. If the line is already in world co-ordinates, no conversion is carried out.
.clone	Copies the line without changing the co-ordinate system.
.to_a	Conversion to array
.to_s	Conversion to string

Instance methods

Method	Description
VC::Line.fromPoints(p1, p2)	Defines a new VC::Line object from two existing point (p1, p2). If the points are in different co-ordinate systems, they must first be converted to the same co-ordinate system.
VC::Line.image(x1, y1, x2, y2, camera=-1)	Defines a new VC::Line object in the image co-ordinate system with x- and y-co-ordinates of two points (x1, y1, x2, y2) and the camera used (camera). If the camera parameter is not defined, the camera of the current image page (-1) is set by default.
VC::Line.world(x1, y1, x2, y2)	Defines a new VC::Line object in the world co-ordinate system with x- and y-co-ordinates of two points (x1, y1, x2, y2).

Class methods

Constant	Description
VC::CoordinateImage	Value 0 (Integer)
VC::CoordinateWorld	Value 1 (Integer)

Constants

Circle class

Method	Description
.x	Value of the x-co-ordinate of the centre point (Integer)
.y	Value of the y-co-ordinate of the centre point (Integer)
.r	Value of the radius (Integer)
.coordinate_system	Co-ordinate system used (Integer: 0 for image co-ordinates, 1 for world co-ordinates)
.camera	Number of the camera (Integer) *Since the calibration of the camera is decisive when converting a point from image to world co-ordinates, you should use this command to identify the right camera.*
.isImage?	Indicates whether the circle is in image co-ordinates (Boolean)
.isWorld?	Indicates whether the circle is in world co-ordinates (Boolean)
.toImage(camera=-1)	Copies the circle to image co-ordinates. If a camera is not defined, the camera of the current image page (-1) is set by default.
.toImage!(camera=-1)	Converts the circle to image co-ordinates. If the circle is already in image co-ordinates, no conversion is carried out. If the camera parameter is not defined, the camera of the current image page (-1) is set by default.
.toWorld	Copies the circle to world co-ordinates.
.toWorld!	Converts the circle to world co-ordinates. If the circle is already in world co-ordinates, no conversion is carried out.
.clone	Copies the circle without changing the co-ordinate system.
.to_a	Conversion to array
.to_s	Conversion to string

Instance methods

Method	Description
VC::Circle.fromPoint (centre, radius)	Defines a new VC::Circle object from an existing point (centre) and a radius value (radius).
VC::Circle.image(x, y, r, camera=-1)	Defines a new VC::Circle object in the image co-ordinate system with x-co-ordinate (x), y-co-ordinate (y), radius size (r) and camera used (camera). If the camera parameter is not defined, the camera of the current image page (-1) is set by default.
VC::Circle.world(x, y, r)	Defines a new VC::Circle object in the world co-ordinate system with x-co-ordinate (x), y-co-ordinate (y) and radius size (r).

Class methods

Constant	Description
VC::CoordinateImage	Value 0 (Integer)
VC::CoordinateWorld	Value 1 (Integer)

Constants

String class

Method	Description
.string	Value of the character string (String)
.clone	Copies the object

Instance methods

Method	Description
VC::String.new(string)	Defines a new VC::String object with a character string (string).

Class methods

Contour class

Method	Description
<< point	Adds a new point (point) to the contour
[pointnumber]	Point (pointnumber) in the contour
.length	Number of points in the contour (Integer)
.points	Points of the contour in an array
.camera	Number of the camera (Integer) *Since the calibration of the camera is decisive when converting a point from image to world co-ordinates, you should use this command to identify the right camera.*
.clone	Copies the object
.to_a	Conversion to an array
.to_s	Conversion to a string

Instance methods

Method	Description
VC::Contour.fromPoints (*points)	Defines a new VC::Contour object from several existing points (points). If the first point is in image co-ordinates, the associated camera is used for the contour. If it is in world co-ordinates, the camera of the current image page is used. All points are converted to image co-ordinates appropriate for the specified camera.
VC::Contour.empty (camera=-1)	Defines a new, empty VC::Contour object with the specified camera (camera). If the camera parameter is not defined, the camera of the current image page (-1) is set by default. The new empty contour can then be provided with the method << with points.

Class methods

Counter class

Method	Description
.value	Value of the counter (Integer)
.clone	Copies the object
.to_a	Conversion to an array

Instance methods

Method	Description
VC::Counter.new(value)	Defines a new VC::Counter object with a specific initial value (value).

Class methods

Image class

Method	Description
.format	Image type (Integer: 0 = greyscale value, 3 = RGB image)
.width	Image width (Integer)
.height	Image height (Integer)
.grey	Greyscale value (Integer)
.red	Colour value red (Integer)
.green	Colour value green (Integer)
.blue	Colour value blue (Integer)

Instance methods

Method	Description
VC::Image.create(page_number=-1)	Defines a new VC::Image object from the specified image page (page_number). We do not recommend using this method, as it fails with images containing over 1,000,000 pixels.
VC::Image.createArea(page_number=-1, px, py, dx, dy)	Defines a new VC::Image object with image information from the specified image page (page_number). In addition, the position of the image section (px, py) and the x- and y-stretch (dx, dy) must be specified.

Class methods

Constant	Description
VC::Image::Grey8	Value 0 (Integer)
VC::Image::Rgbx	Value 3 (Integer)
VC::Image::MaxPixels	Value 1,000,000 (Integer)

Constants

8.4 Demo Programs

Overview

You can find the demo programs in the program directory vcwin . Open a demo program in the programming window in order to learn more about the demo programs through the program comments. The following table contains an overview of the demo program. The demo programs can be modified and new demo programs can be added.

File name	System	Description
Calibrating		
calib.vc	All	Calibrates the coordinate system of the camera.
Counter		
counter.vc	All	Displays the use of the **Counter** command in**vcwin**.
Image Acquisition		
single.vc	pictor Mxxxx, vicosys	vicosys: Usual image acquisition with standard image acquisition and display on a monitor.
single-fast.vc	pictor Mxxxx, vicosys	Fast image acquisition with standard image acquisition and **without display on a monitor**.
multi-camera-parallel.vc	vicosys	Simultaneous image acquisition with two cameras in a multi-screen.
multi-camera-sequential.vc	vicosys	Alternating image acquisition from two cameras in a multi-screen.
Program Handling		
simple.vc	pictor Mxxxx, vicosys	Program structure with trigger input, image acquisition, image processing, analysis, and output (digital outputs / monitor).
geocopy.vc	All	Displays the effect of the vcwin command **Copy Geometry Variables** for the work with program loops in vcwin .
demo.vc	pictor Mxxxx/E, vicosys	A simple test program with image acquisition, image processing, and visualization.
UP_LIO.vc	pictor Mxxxx/E, vicosys	Program structure with selection of the subprogram through digital I/O (LineI/O).
UP_PIO.vc	pictor Mxxxx/E, vicosys	Program structure with selection of the subprogram through digital I/O (PortI/O).
Measurement User Window		
send-pic.vc	pictor Mxxxx/E, vicosys	Cyclically sends the currently recorded JPG image through Ethernet.
send-data_ethernet.vc	pictor Mxxxx/E, vicosys	Sends measurements to the user window over Ethernet.
send-data_rs232.vc	pictor Mxxxx/E, vicosys	Sends measurements to the user window over RS232
Applications		
distance.vc	All	Measures a section in the image, incl. position tracking of the test object and good/bad evaluation.
radius.vc	All	Measures the radius of a circle in the image, incl. position tracking of the test object and good/bad evaluation.

File name	System	Description
datamatrix.vc	pictor Mxxxx/E, vicosys	Data-Matrix
ocr.vc	pictor Mxxxx/E, vicosys	OCR
completeness1.vc	pictor Mxxxx/E, vicosys	Integrity checks on orbits.
completeness2.vc	pictor Mxxxx/E, vicosys	Integrity checks on fixed positions.

8.5 Evaluating the print quality

Guidelines

ISO/IEC 15415 and AIM DPM-1-2006 are standardised guidelines for evaluating the print quality of 2D and DPM codes. They define ten parameters in each case which are intended to provide a measurement of the readability and quality of a code. AIM DPM-1-2006 is an ISO enhancement which concerns itself specifically with the quality of DPM codes.

The guidelines contain the following parameters whose results are stored in vcwin in the same order in the result structure:

ISO/IEC 15415

[0] Overall Quality
[1] Fixed Pattern Damage
[2] Decode Quality
[3] Axial Non-Uniformity
[4] Grid Non-Uniformity
[5] Unused Error Correction
[6] Modulation
[7] Symbol Contrast
[8] Mean Light
[9] Mean Dark

AIM DPM-1-2006

[0] Overall Quality
[1] Fixed Pattern Damage
[2] Decode Quality
[3] Axial Non-Uniformity
[4] Grid Non-Uniformity
[5] Unused Error Correction
[6] Cell Modulation
[7] Cell Contrast
[8] Mean Light
[9] Mean Dark

The functions of the Symbol Contrast / Cell Contrast and Modulation / Cell Modulation parameters are identical. They were renamed with AIM DPM-1-2006 to emphasise differences in the way of determining the value.

Each parameter can have a result between 0 and 4, whereby 4 is the best value and 0 the worst. The Overall Quality parameter receives the value of the lowest result of the evaluated parameters [1] - [7] and describes the determined print quality. For a satisfactory result, it should have a value of at least 2. However, in theory, the code can also be read and output even if the overall evaluation is lower.

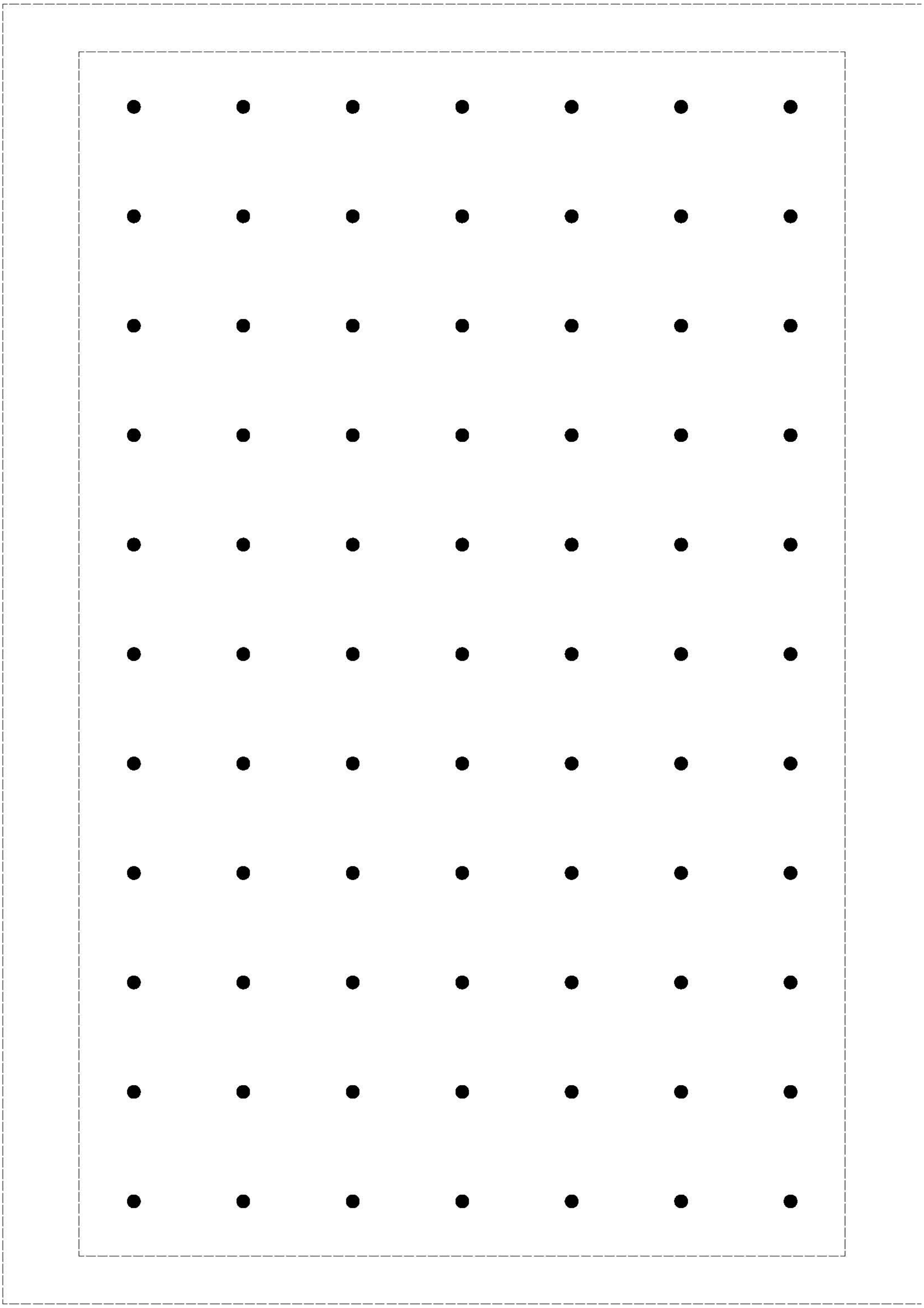
Parameter

- **Overall Quality** is the lowest result of the individual sub-evaluations [1] - [7] and describes the overall quality in terms of readability.
- **Fixed Pattern Damage** evaluates the condition of the fixed elements "Finder Pattern" (the lines arranged in an L-shape which delimit the code), "Alternative Pattern" (dotted lines opposite the Finder Pattern) and the quiet zones around the code. If there are gaps or print faults in these areas, this can lead to read problems and will accordingly be given a worse evaluation.
- **Decode Quality** examines the syntax of the code and decodes it (if possible) with a reference decoder algorithm. Only the two results "Pass" (4) or "Fail" (0) are possible for this parameter. If this test is not successful, no further quality checking can be carried out.

- **Axial Non-Uniformity** describes any horizontal or vertical distortion of the code or the module.
- **Grid Non-Uniformity** describes any general distortion or displacement of the code or the module.
- **Unused Error Correction** evaluates the number of remaining correction bytes in the error correction overhead that were not used for error correction. The higher the number the better the quality of the code and therefore the higher the result.
- **Modulation** or **Cell Modulation** returns a value for the uniformity of the intensity of light and dark modules in the code. A consistent intensity of the light and dark modules simplifies matching and makes the code easier to decode.
- **Symbol Contrast** or **Cell Contrast** evaluates the contrast between light and dark modules within the code.
- **Mean Light** is the average brightness of the light modules in % or greyscale value (for 8-bit resolutions). According to AIM DPM-1-2006, this value should be between 70 and 86% or, as the case may be, a greyscale value between 180 and 220 in order to create a good, reproducible brightness ratio. In ISO/IEC 15415, the lighting is defined additionally.
- **Mean Dark** is the average brightness of the dark modules in % or greyscale value (for 8-bit resolutions) (see Mean Light).

For evaluating the print quality according to one of the guidelines, you should use the corresponding lighting definition for the respective guideline.

8.6 Dot Matrix



8.7 Technical Support

Please contact our technical support if you have any technical questions concerning our products.

We will be glad to be of service:

Monday to Thursday 8:00 to 17:00, and Friday 8:00 to 15:00.

Vision & Control GmbH

Mittelbergstraße 16

98527 Suhl, Germany

Phone: +49 (0) 3681 7974-0

www.vision-control.com

Update of the operating software

The current software version of the operating software can be downloaded from our website *www.vision-control.com*

9 LIST OF FIGURES

10 INDEX

Vision & Control GmbH
Mittelbergstraße 16
98527 Suhl, Germany
Telephone +49 (0) 3681 7974-0
Telefax: +49 (0) 3681 7974-33
www.vision-control.com