User Manual

SCORPION VISION SOFTWARE®

Evolving visions

Contents

1 Introduction

Scorpion Vision Software® is software for configuration and operation of vision systems. Scorpion is targeting tasks as identification, sorting, robot guiding, assembly verification and quality control. It's designed to secure the quality of a production process. Scorpion is as easy to use as a vision-sensor still having the flexibility and power of a real vision-pc solution. The system is founded on top of a standard Windows PC platform.

Scorpion can identify units based on criteria like dimension, form, grey scale, colour, text and code. Quality control can be based on dimension, surface, flaw, assembly etc. Scorpion can also be used in factory automation to identify parts, sorting, robot guiding and automatic program selection.

To monitor the inspection result and the production process, Scorpion offers statistics and detailed result information. Scorpion offers the end-user a feature rich and functional graphical user interface with image display, result panels, image history list, real time trends, alarm management, event log, quality alarms, remote control, user configurable logging and rich configuration profiles.

The flexible and configurable communication line interface allows managing of control signals and data exposure through Profibus-DP, digital and analogue I/O, rs-232 and TCP/IP. The text based communication protocol, TdvCmdProtocol, is the kernel of the communication.

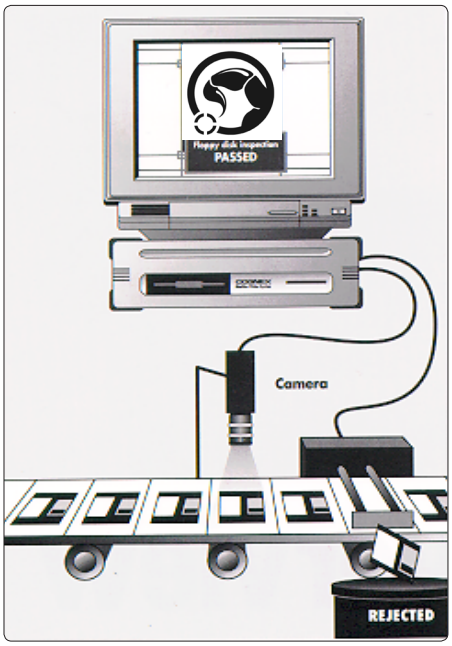
Scorpion contains an extensible set of configuration tools. They are categorized in Basic, Data, Edge, Geometry, Reference, 3D and Advanced tools. Combining these tools, Scorpion targets both the simple and the advanced inspection tasks.

Scorpion can simultaneously serve and communicate with multiple independent units like PLCs, robots and production lines.

Additionally Scorpion supports on-the-fly reconfiguring to handle different product variants within one single production line. A broad range of camera configurations are supported, including colour cameras.

A unique feature within Scorpion is the possibility to perform a complete offline system verification automatically or interactively using the integrated image history list or captured image data.

This manual guides you through the Scorpion Vision Software user interface, and gives you some hints and ideas on the way.



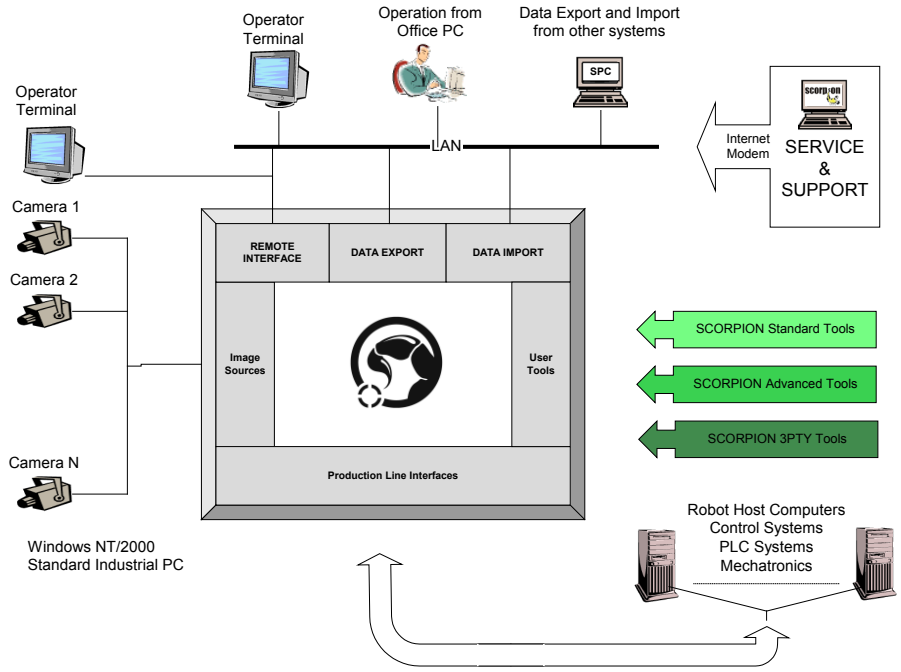
1.1 Customer support

Tordivel AS offers many levels of Scorpion Vision Software® customer support. You find more information on our home page - http://www.scorpionvision.com.

2 System Description

A Scorpion Vision Software® installation typically consists of an industrial PC running Windows, the inspection system consisting of the Scorpion software with system profile, one or more cameras, lighting, mechanics and a production line interface for external communication. The profile decides how the system shall operate.

The Scorpion Vision Software architecture is based on a kernel providing basic functionality where you plug in image sources and user tools. The user interface is highly configurable.
The Scorpion Vision Software architecture makes the product very flexible and suitable for a large variety of inspection tasks.



Typical units in a Scorpion Vision Software® system.

3 Installation and Start up

Scorpion Vision Software® is distributed on a CD with the following contents:

- Scorpion Vision Software
- System requirements
- Scorpion setup program
- Demonstration profiles
- Camera drivers
- Documentation
- Support program

3.1 Installation

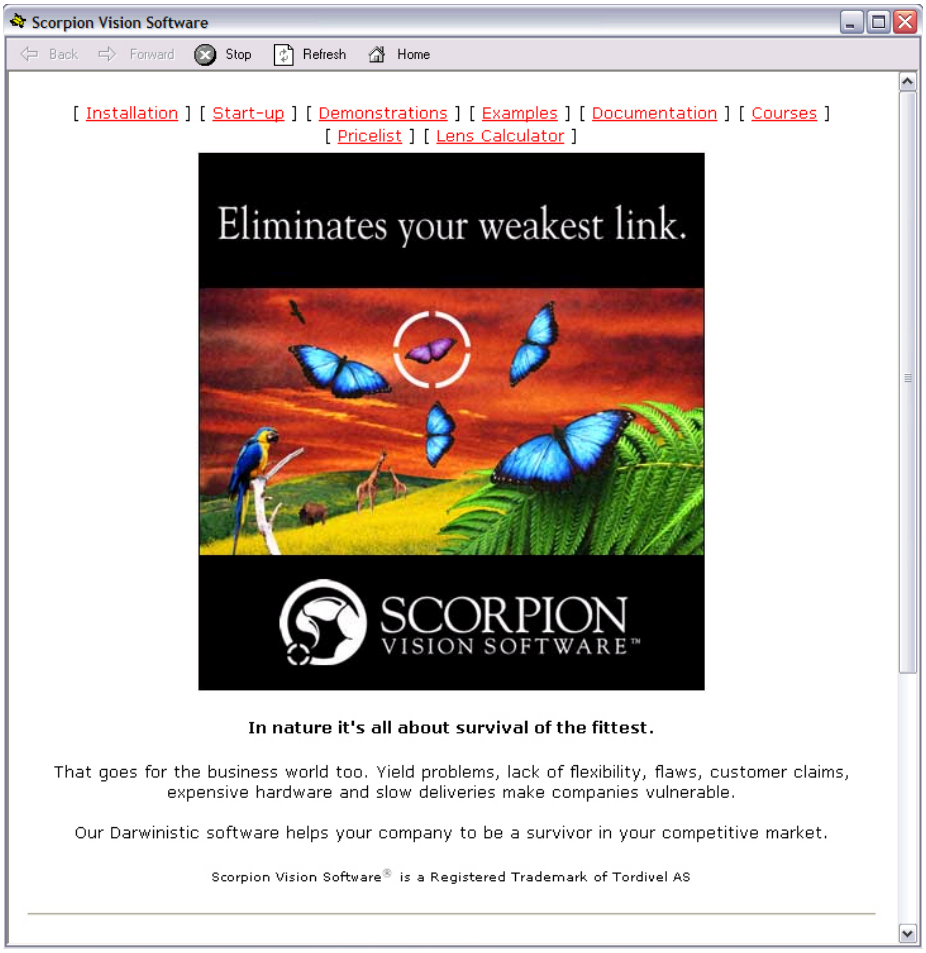
1. Turn on your computer. Insert the Scorpion Vision Software® CD-ROM disc into your CD-ROM drive. A window like the one below appears.

2. Select *Installation* to start the installation process. It is wise to read the *System Requirements* and the *ReadMe* file before installing Scorpion Vision Software.

3. Select *Install - Scorpion Vision Software*. Follow the instructions coming up.

4. Select *www.scorpionvision.com* and get a demonstration license. Copy the license code in the edit field when prompted. Read more about licensing in the "Licensing" chapter.

5. To learn about Scorpion, select *Start-up* and install the demonstration profiles using stored images. We highly recommend you to install and go through these demos.



CD-ROM start up

You find the Scorpion user documentation under *Documentation*. For live demonstrations select *Demonstrations*. More profiles are available under *Examples*. Choose *Courses* to see the Scorpion Vision Software Introduction and Advanced course content. Select *Support Applications* to get information on applications supporting the Scorpion Vision Software.

You also find camera drivers and links to Python used in some Scorpion profiles, on the CD. The *Lens calculator* finds the lens size and the appropriate field of view based on information you give.

3.2 Licensing

Scorpion Vision Software® is licensed software. A license is related to the computer's network board. There are three types of licenses:

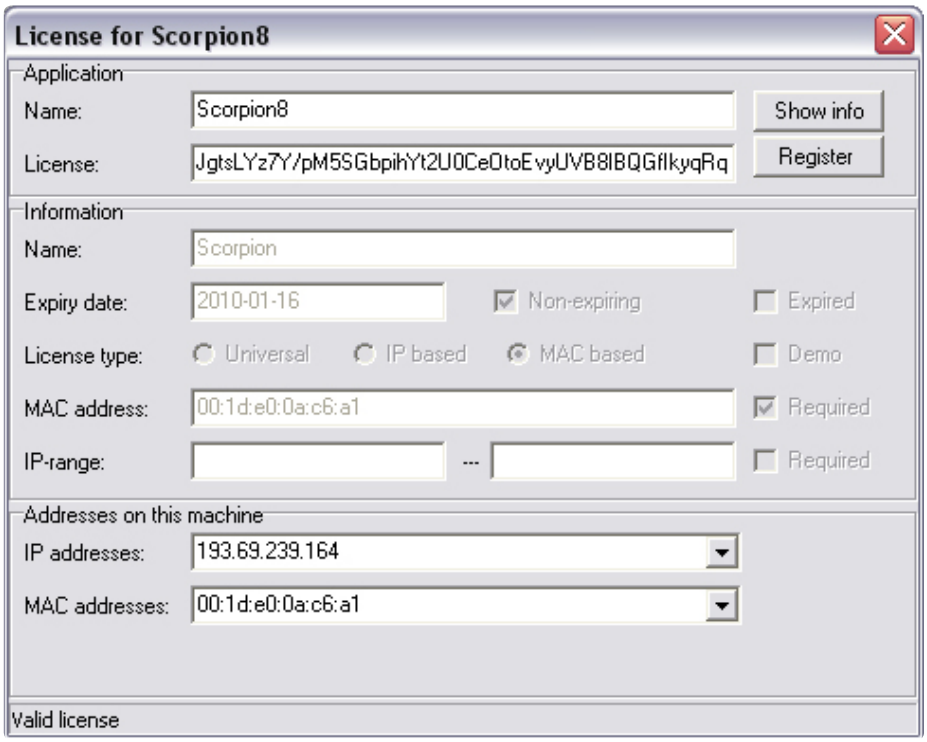
- Demo license – can be run in a limited period of time for evaluation purposes
- Maintenance license – is used for profile maintenance based on images stored on file
- Full license: *Lite*, *Basic*, *Premium* and *Vision Server* – is related to available functionality and the number of cameras and meant to run in a production environment

Licenses can be retrieved over the Internet: www.scorpionvision.com.

If your computer does not have a correct license, Scorpion will at start up ask the user to apply the license code.

You write or copy the license in the *License* field. Then you press the *Register* button. If the license is accepted, Scorpion will start as normal. The IP-address and MAC-address fields are only for information and are used when requesting a license. Normally the license is related to the computer's network board. The MAC-address uniquely identifies this board.

If you want to check which Scorpion license that is installed on your computer, open the About window. Here you can activate the license information window as shown below.



License information

3.3 Start up with the Demo profiles

Scorpion is a general inspection system. A profile makes it dedicated and special for an inspection task. The configuration done to perform an inspection task leads to a profile.

The demonstration profiles are valuable examples and demonstrate Scorpion's capabilities as a vision system. They get their images from file and give a good insight in Scorpion set up and operation. A number of such profiles with corresponding presentations are included on the Scorpion CD. The *Getting Started* exercises lead you through the basic Scorpion system concepts. This is a compact user's course and leads you through the most important Scorpion features. The *Robot Vision Start* and *Final* are the profiles to go for if you are looking into robot vision.

1. Select *Start-up* on the CD to install one or more of the demos.

2. Load the profile's zip file and save it in the Scorpion\Archive directory.

3. Start Scorpion. Go to the Windows Start menu. Select Start - Programs - Tordivel Vision Solutions - Scorpion. A profile selection dialog is then shown.

4. In the dialog window, right click the mouse and select *Restore*.

5. In the folder coming up, browse to the Archive directory where you saved the zip file and open it.

6. The Profile is now in the dialog window and you can open the application by double-clicking on it.

On the CD there is also a directory of example demo profiles - *Examples*. They are installed in the same way.
After installation, you'll find a shortcut to the profiles under the Windows Start menu in the Tordivel Vision Solutions\Scorpion program group. You also find a shortcut to this User Manual in the same program group.

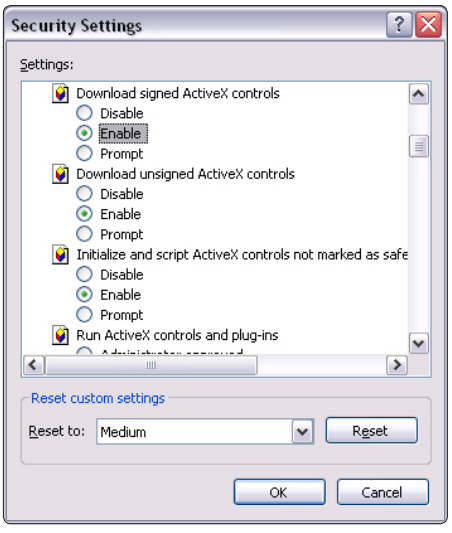
3.3.1 Requirements for demos

- *Internet Explorer 5.0 or higher*
- *A registered demo-license*

The Label demo runs ActiveX controls from Tordivel Software Solutions in Internet Explorer. This may cause errors or warnings from Internet Explorer at start up of the demo.

These errors and warnings can be avoided by activating the following options in Internet Explorer under Tools - Internet Options – Security - Custom Level:

- Download signed and unsigned ActiveX controls
- Initialise and script ActiveX controls not marked as safe

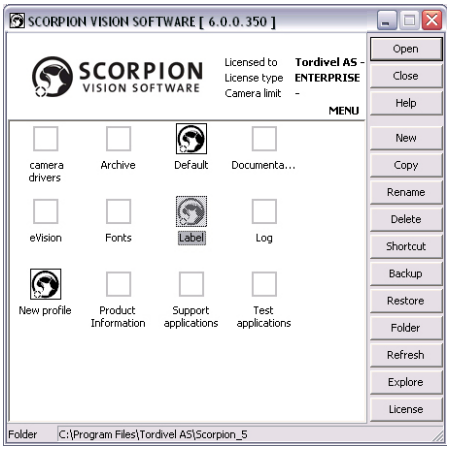


Internet Explorer Security Settings

3.3.2 Start up - select a Profile

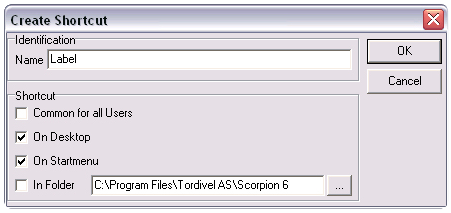
Go to the Windows Start menu and select Tordivel Vision Solutions\Scorpion to start Scorpion. A profile selection dialog is then shown.

The following operations are available:



Select a profile to start from the profile selection dialog. Here seen with large icons.

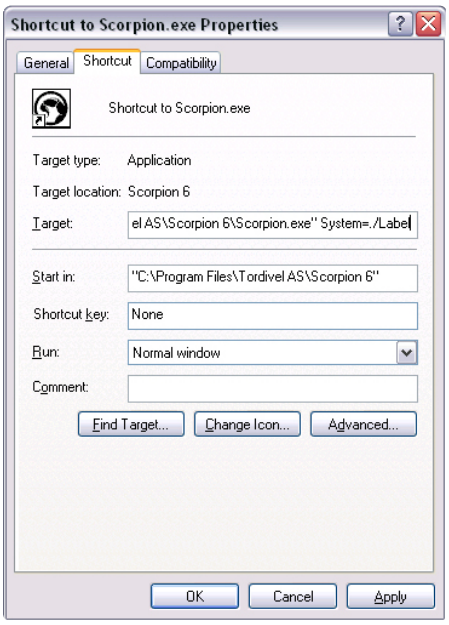
- Open - opens a selected profile
- Close - closes the profile selection dialog
- New - creates a new profile
- Copy - copies a Scorpion profile to another profile
- Rename - renames a selected profile
- Delete - deletes a Scorpion profile
- Shortcut - creates a shortcut to the profile. It is vice to establish a shortcut to each profile if you have more than one profile on your computer. You place the shortcut either on the Desktop, under the Scorpion program group on the Start menu or in the system folder. This makes the start up easy and convenient.
- Backup - makes a backup of a Scorpion profile in zip-format. Subdirectories are included.
- Restore - restores a profile from a given directory. A folder is opened for browsing.
- Folder - selects a folder for the profile selection dialog.
- Refresh - refreshes the profile selection dialog – if there has been changes.
- Explore - opens the Windows Explorer
- License information
- Large icons - toggles the list view
- Show buttons - toggles the buttons to the right on or off
- Help - activates the help window



Create a shortcut to the profile

Select the wanted profile, press Open (or double-click the profile) and Scorpion starts.

3.3.3 Shortcut at start up



Shortcut to a Scorpion profile

A Scorpion system has the following command line parameters:

- Scorpion System=<path>

If no path is given, Scorpion uses the .\Default path. The path normally points to a Scorpion profile – a system.

It is vise to establish a shortcut to each profile if you have more than one profile on your computer. A shortcut setup is shown above. You make a shortcut by opening the Windows Explorer and navigate to Scorpion.exe. You normally find this file on the path ..\program files\ tordivel as\scorpion… Select the Scorpion.exe file, press the right mouse button and choose *Create shortcut*. In the shortcut properties you change the Target to System=.\<path to profile>.

Example: ”C:\Program files\Tordivel AS\Scorpion\Scorpion.exe” System=.\Label

You will in this example start the Label profile placed on a sub folder.
You place the shortcut on the Desktop or under the Scorpion program group. Name the shortcut ’Scorpion - Label’, and then you can easily see which system that starts.

3.4 Making a new Profile

To make a new profile, you do as follows:

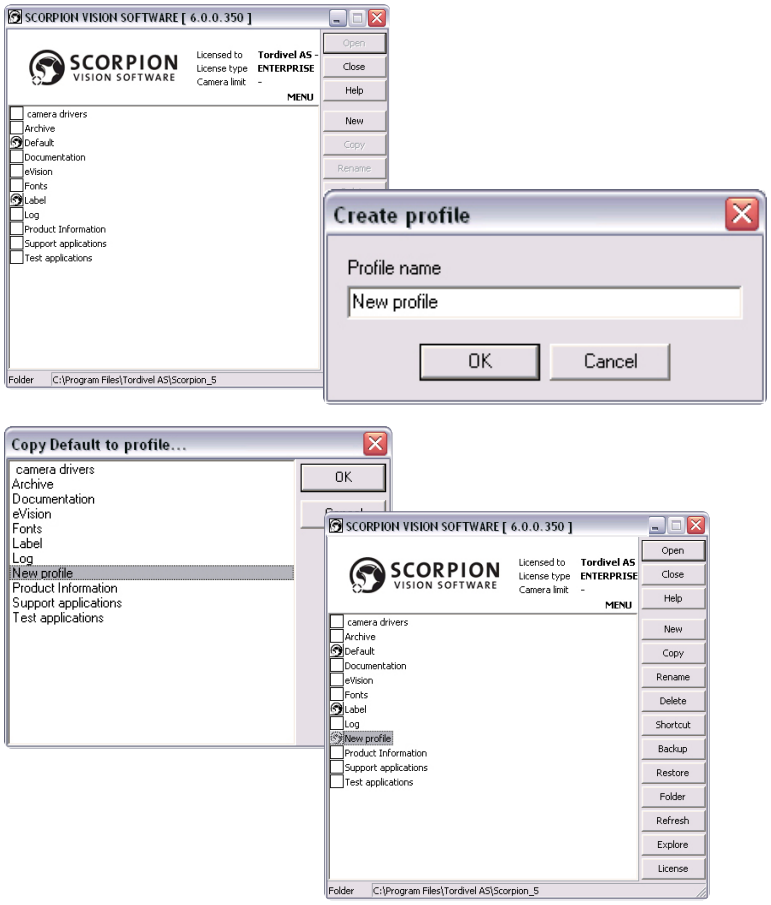
1. Go to the Windows Start menu

2. Select Start - Programs - Tordivel Vision Solutions - Scorpion. A profile selection dialog is then shown.

3. Press New and give the profile a name in the box coming up.

4. Press OK and the name is shown in the profile selection dialog. Select the Default profile, which contains the basic setup, press Copy and a window *Copy Default to profile* appears. Select your new profile and press OK.

5. You can now select the profile in the profile selection dialog and press OK or double click the profile name, and Scorpion starts with your profile.



Making a new profile

Backup
To make a backup of the profile, go to the *Service – Maintenance* panel. You can also use the backup to move the profile to another computer having the same Scorpion version installed.

3.4.1 Profile content

A Scorpion profile normally consists of the following information:

- <path>\General.ini – contains the set up of the Scorpion profile
- <path>\Statistics.ini – contains the statistical information
- <path>\CVLGrab.ini – contains the set up for the image source
- <path>\Scorpion.spb – contains the Scorpion set up in xml-format

Additionally you find the Images sub folder containing the profile's images.

3.5 Activating a camera

Connect your camera and capture images with Scorpion starting with the Default profile. The Default profile can be changed to take live images with the following steps:

1. Select *Drivers* from the CD-ROM window to install selected camera drivers. The DirectX8.1 driver is required for camera operation under Windows 2000.

2. Connect the camera to the PC. The FireGrab.dll camera interface supports all windows imaging devices including web cameras and Firewire cameras.

3. Go to the Windows *Start* menu

4. Select *Start - Programs - Tordivel Vision Solutions - Scorpion*. A profile selection dialog is then shown.

5. Select the *Default* profile

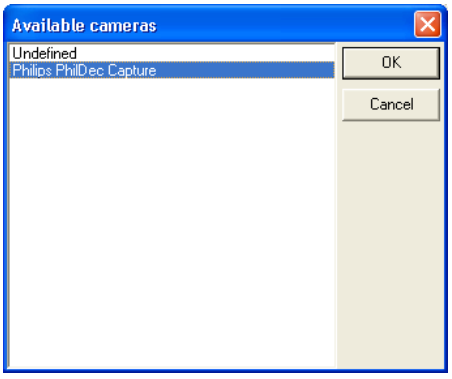
6. Press *Service* in the main toolbar - 911 is the initial password

7. Activate *Service* in the minor toolbar

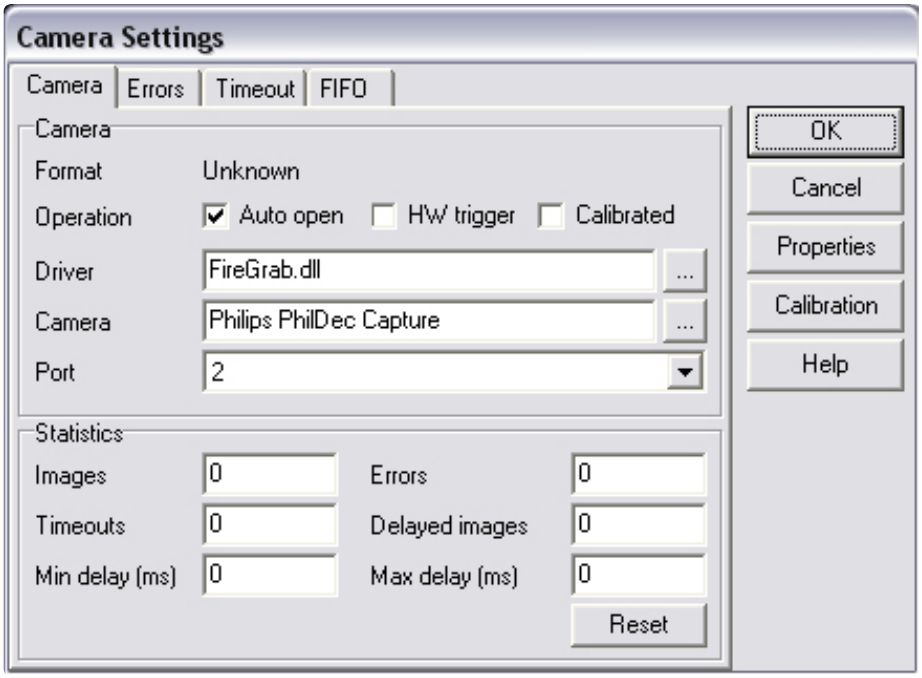
8. Select the *Camera* tag and *New*

9. Under *Camera Settings* select the Camera **…** box

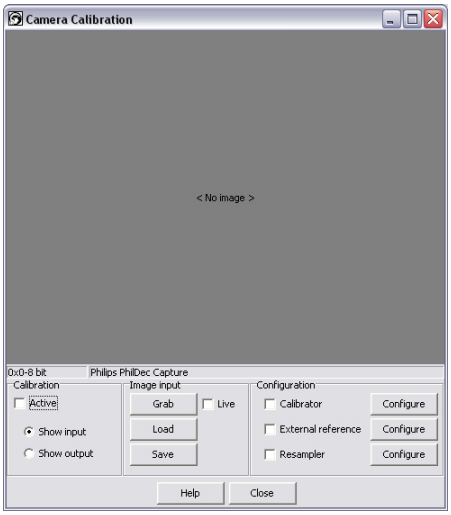
10. Browse to select a new camera and the list of available cameras is shown. Select the right one.



List of available cameras



10. Setup is used to set the camera properties. The avaiable information differs dependant on the type of camera. Usually the image format and number of frames per second are available. Using Firewire cameras it is wise to reduce the frames per seconds to a minimum. Calibration opens a camera calibration dialog.



Camera calibration

11. When closing the Camera dialog a green checkmark shall appear - this means that the camera is open - a red checkmark can mean that another application has opened the camera or simply an error.

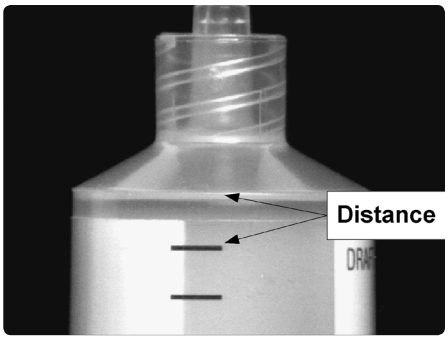
12. Uncheck the Simulate option under Camera - Image Settings.
Scorpion is now ready for capturing live images.

13. Press Snapshot in the main toolbar - the image shall appear in the left pane of Scorpion.

14. Under Camera - Image Settings the name of the images can be changed using edit and setting the image properties.

4 Introduction to example - Label on Syringe

We use an inspection task from the pharmaceutical industry as an example through out the book.



Measuring distance on syringe

To the left you see an image of a syringe taken by Scorpion. The image is taken with an industrial black and white camera, Sony XC-75, with 760x575 pixels and 256 grey tones. To ensure contrast on the syringe edges, we have chosen a dark background. The syringe is diffusely lightened from above.

The task is to check that the label with lines for the measuring level (level indicator) is correctly positioned on the syringe. We are doing this by controlling the distance between the upper level indicator and the line that defines the bending point (transition to tip) on the syringe. The image is approximately 30 x 23 mm. This means that the image point resolution is 0.04 mm.
The task is to control that the distance is within 3.6 to 3.85 mm. To ensure this, the measure resolution must be better than 0.01 mm and the measure tolerance better than 0.05 mm.

Scorpion solves this by using edge-finding tools. These tools can find edges with a precision 10 - 20 times better than the point resolution. Better than 0.004mm in our example. This precision is necessary in the in-between results to ensure the precision of the final result, the distance between the upper level indicator and the line between the syringe bending points.

Scorpion defines a set of possible states connected to an inspection task. In our example these are:

- Pass – the measured value is within the given limits
- Distance low – the measured value is below the lower limit
- Distance high – the measured value is above the upper limit
- Cannot measure – the inspection failed and no values are found
- No syringe – no syringe is found in the camera image

It is important to define a descriptive set of states and establish statistics for them when evaluating the quality of an inspection system.
In our example a high number of “Distance low” will indicate that the placement process systematically places the label to low. This must be corrected in the placement process. A high number of “Cannot measure” can indicate too high variations in the looks of the label, the image analyses may be too week or the measurement construction is unstable. If the “No syringe” state appears too often, it can be the syringe presentation that fails. More than one error may occur at the same time, and then the analyses of the statistics are more complicated.

See the appendix for more details on this inspection example.

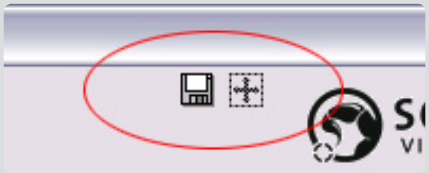
5 Normal operation

Scorpion has three modes of operation:
- *Normal operation*
- *Settings*
- *Service*

When running in normal operation mode, the following information is available:
- Description - Web page that contains a short description of the inspection task and buttons for operating the system
- History - displays the latest inspection results
- Curves - give a graphical view of measured values
- Results - show measured values of the latest inspection
- Statistics - give a periodical view of the inspection results
- Camera image(s)
- Inspection result with indicator panels

Additionally you can start and stop the system.

Shortcut Symbols



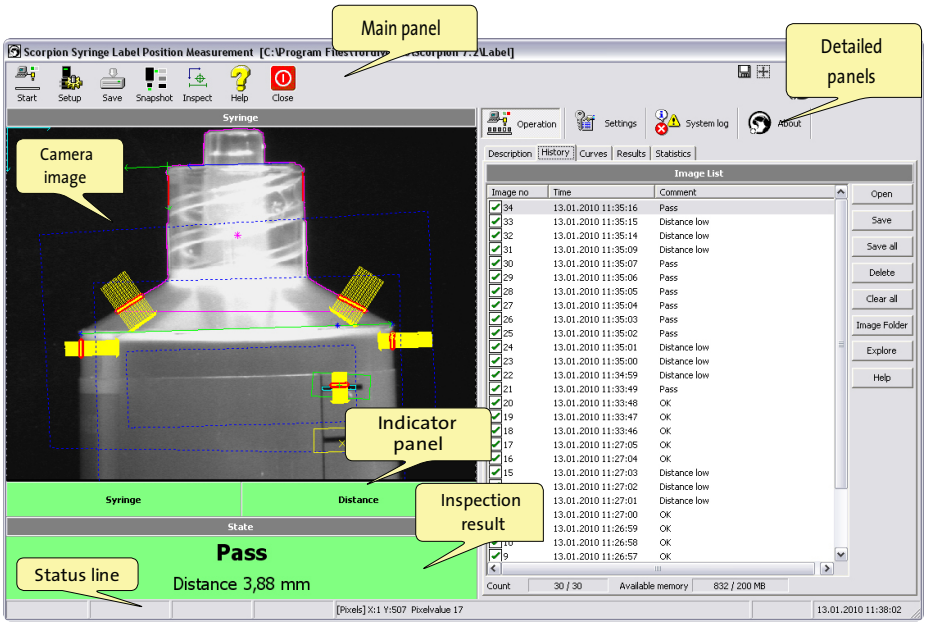
At the upper right you find convenient shortcuts for often used functions. Move the mouse over the symbol and a descriptive text is shown.

- Save current image to disk
- Full image mode - image and result panels displayed

Icon Symbols

Icon symbols are often used in detailed panels to indicate the state of for example an inspection, a tool or a system operation. Their meaning is as follows:

- *Not run*
- *Ok*
- *Blocked by guard or reference*
- *Error or No result*
- *Not active*
- *The license is not covering the use of this tool*
- *Manual execution*



Main window in normal operation mode

The screen picture under normal operation is shown above. The picture can roughly be divided in the following parts:

1. *Main buttons* (upper row, below the main window title). These buttons are used to *Start*, *Stop* and *Close* the system. Additionally there are two buttons to lock/unlock the password protected Settings and Service panels. *Snapshot* takes an image and you can do a manual inspection by pressing *Inspect*. Save the profile by pressing *Save*. If two or more systems are simultaneously running on the same machine, the *Next* button is used to toggle between the systems.

2. *Image* (left, below the main buttons). The camera images of the unit to be identified are shown here. Selecting and dragging the image can zoom in details. A simple click zooms back. You can choose to see one or all images at a time in systems using more than one image for classification. See the 'Camera image processing' chapter for a description of available camera image features.

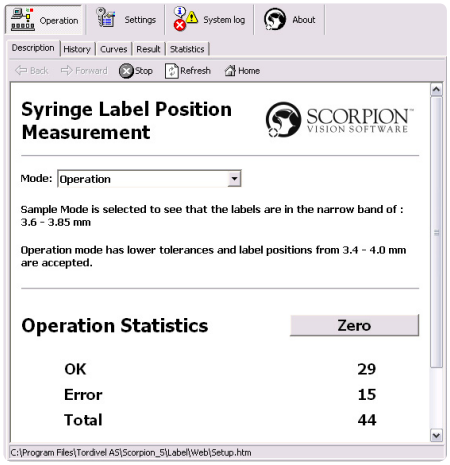
3. *Inspection Results* (lower left) shows the running inspection results. The classification result is shown in text – we call it the state. Additionally you can display one or more result parameters in this field. (In the above example given by the 'Distance' parameter.) Right click the mouse over the field, and you see the menu to choose from. You need however to be authorized to change the result panel set-up. Normal conditions are usually indicated by a green background colour, other conditions by for instance yellow or red. You set the colour under Settings-States.

4. The *Indicator panel* is showing selected measured values calculated by Scorpion's logical tools. The inspection result is based on these values. On error, the representing indicator panel field changes colour to e.g. red to illustrate the cause of the error. If you are authorized, you can change the value to be shown as well as the title and colour of the indicator panel fields. Press the right mouse button over the field and select from the menu showing up. You can also extend the panel by adding more fields or remove fields from the panel.

5. *Detailed panels* (right), here you can chose different detailed information: history, curves, results or statistics. Closer descriptions of these panels are given in the rest of this chapter.

6. *Status bar* (bottom line) shows different indicators of the system status. From left to right:
- Image trigger - status for ready signal from the production line
- Status for manual code signal from the production line
- Status for reset signal from the production line
- Status for quality alarm
- Row/column coordinates, pixel value and name of graphical image components. The values change when moving the mouse within an image.
- Date and time

5.1 Web explorer



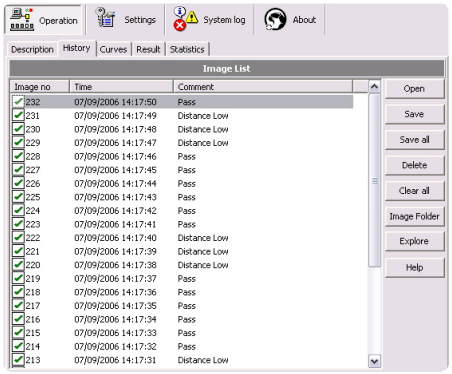
Operating the system from the web explorer

Here you find a web page with a short description of the inspection task. Buttons are available for operating the system. System parameters can be made available for changing from this page. In systems where i.e. limits are varying this can be very useful. In most cases you also have a simple statistics overview here.

Under Properties in the web page's tool panel you can, if you are authorized, change the page setup. You can decide which page to be the home page and if the tool panel with buttons and text, status line and page title shall be shown.

You make the page content with an editor, for example Microsoft FrontPage. You can relate Scorpion's commands and parameters to buttons and boxes on the web page.

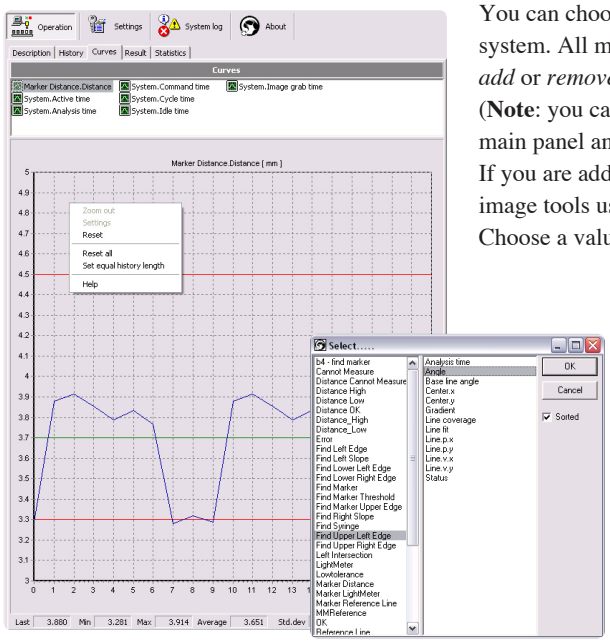
5.2 History



List if images with inspection result

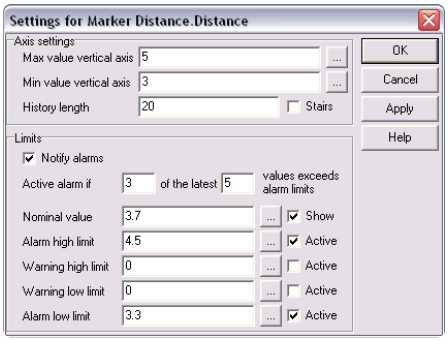
The latest measurements are shown in the image list. A number identifies each image taken. The time and the classification result (the state) are additionally given in the list. By selecting an entry in the list, the image is displayed in the Image window to the left on the screen.
You can choose to save or delete all or single images, choose how many images to display in the list, open images from another folder and choose to show the buttons to the right. You can also do these operations with the buttons.

5.3 Curves



You can choose between numbers of curves illustrating different values calculated by the system. All measured values can be graphed. Right click the mouse to see a menu and select *add* or *remove curve*.
(**Note**: you can only add or remove a curve in Settings or Service mode. Select Settings in the main panel and give the password.)
If you are adding a curve, you get a list of all values available. The left column shows the image tools used in the analyses. The right column shows the values measured by the tool. Choose a value, click *Ok* and the value is graphed.

Right click the mouse over the curve, and you can reset this or all curves, open the settings panel (if you are authorized) or set the history length similar for all curves.



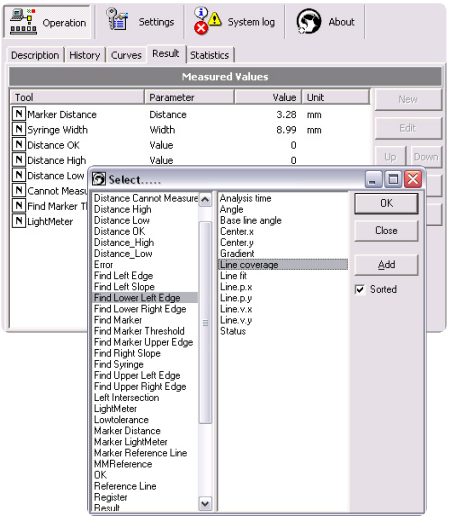
Curve adjustments

Adjustments of alarm limits and curve values can be done if you are in settings mode (click on Settings in the main menu and give the correct PIN code). Double click the curve name to see the adjustment parameters. You can adjust the curve axes and the alarm limits (alarm limits are here shown in red). The image to the left gives you an example.

The curves show if measured values are within given limits. This is useful e.g. to check the light conditions. Select the *Notify alarm* field if alarms are to be notified. Alarm limits are normally defined based on operator experience. There are two independent types of limits: operational and alarm limits. An event in the system log and a quality alarm are the results of an exceeded operational limit. If an alarm limit is exceeded you can make the system stop if the option for this is chosen (see Service-General).

Supervise only the parameters that operator experience finds useful and give them realistic limits. You may loose overview, control and trust in the system if the number of supervised parameters and generated alarms are too large.

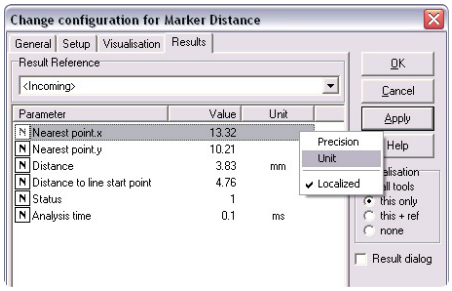
5.4 Results



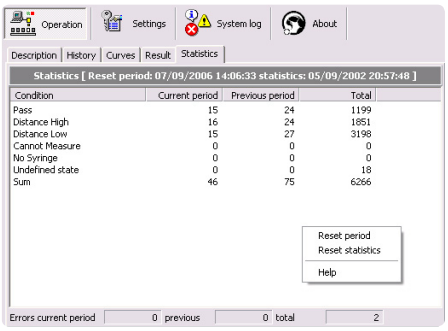
Results

Measured values of each inspection are shown in this panel. You can choose which parameters to display by selecting *New* and choose from the list coming up. The left column shows the image tools used in the classification. The right column shows the values measured by the tool. Choose a value, click *Add* (then the window stays open – smart if you want to add more values) or *Ok*, and the value is included in the overview.

You set the unit and precision of the measured values from the tool window in the Toolbox (to the right).
See the Service - Toolbox chapter.



5.5 Statistics



Statistics

A periodical view of the inspection results is shown here. The table has one row for each classification state. The columns show the inspection result for this period, last period and the total.

By right clicking the mouse and selecting from the menu shown, you can manually reset the statistics. You can also choose to automatically save the statistics. The time and frequency for doing this is chosen under Scheduler in the Service panel.

With the system command *Statistics*, the following operations can be done either by the Scheduler, through the communication interface or at given system events:

- Statistics;cmd=zero resets the period statistics
- Statistics;cmd=reset resets all statistics
- Statistics;cmd=save saves the statistics to file

The statistics are saved if the system is terminated and is reloaded on restart.

5.6 Camera image processing

Scorpion supports many operations on the camera image. Right click the mouse over the image to activate the menu.

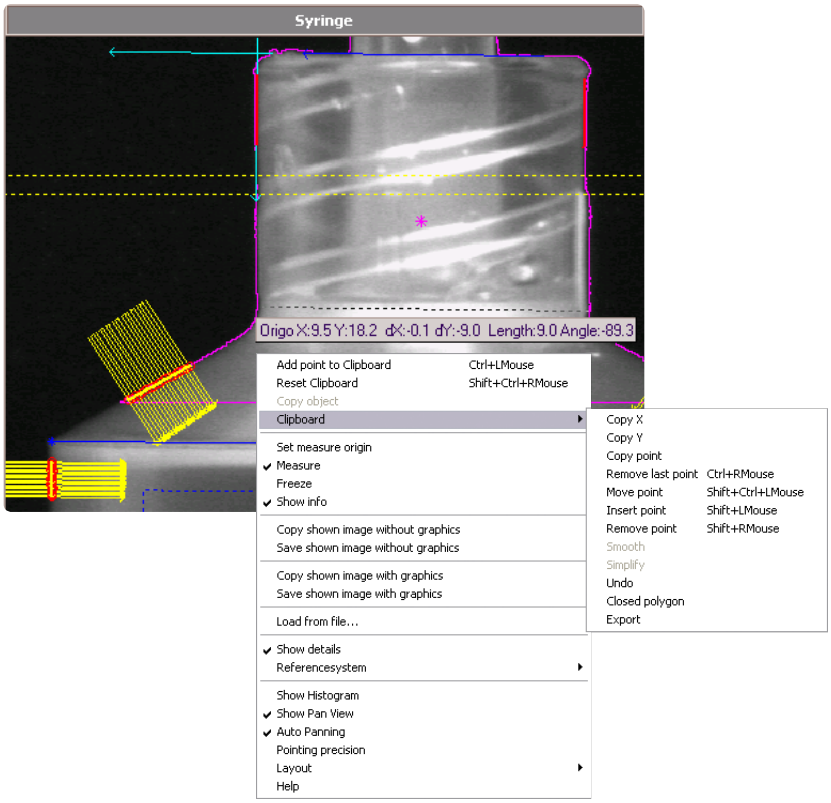


Image operations

5.6.1 Image zoom

Zoom in and out this way:

- Zoom in - click left mouse button and drag the cursor
- Pan the zoomed rectangle - while zooming in, press the *Alt* button
- Zoom out - left click the mouse in the image

Note: multiple zoom operations will create a stack of zoom levels - to completely un-zoom the image left click the mouse button repeatedly.

5.6.2 Measure intensity values

Click the right mouse button in the camera image, and see the menu as shown in the image above.
Select *Show info* and a text line with point information will follow the cursor.

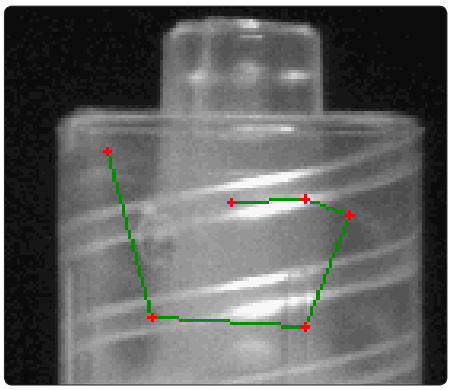
5.6.3 Measure

You can measure distances and angles directly in the image by using the *Set measure origin*, *Measure* and *Freeze* commands. Click the right mouse button when starting (*Set measure origin*) and ending (*Freeze*). The measuring result is shown on the cursor text line. This value relates to the reference system chosen for the image.

You can make new reference systems with the Scorpion tools. These will show up under *Reference system* in the menu above. This is useful e.g. in robot vision systems where Scorpion can be set up to work in the same coordinate system as the robot. The default reference system is set using Reference system in the menu below. By default pixels are used to give the results. When a tool in the toolbox is active the reference system is set by the tool.

5.6.4 Polygons

You can draw and edit polygons on an image with the *Clipboard* operations. The purpose is to define lines and region of interests for image tools like PolyLineGapFinder and Blob2.
Open the pop-up menu on the right mouse button over the wanted point and select *Add point to Clipboard*. You will see the point marked in the image. Continue adding points and Scorpion will automatically draw lines between them. If you choose the *Closed polygon* option, a line will also automatically connect the first and last points. A shortcut to add point is pressing *Ctrl* and the right mouse button - points are removed by pressing *Ctrl* and the left mouse button. To the left such a line is shown.



Drawing lines and defining polygons on an image.

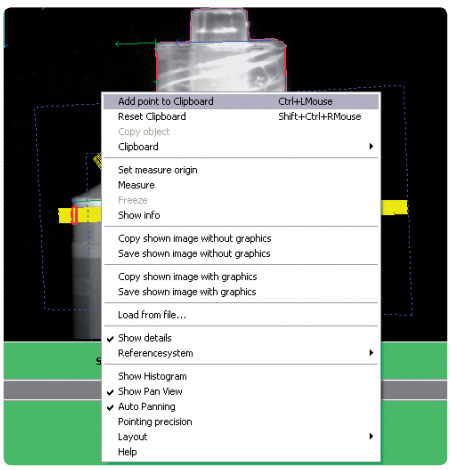
These are the image operations:

- Add point to clipboard - Ctrl+LMouse - adds point to clipboard
- Reset Clipboard - Ctrl+Shift+RMouse - clears polygon on clipboard
- Copy objects - copies any overlay object into clipboard
  - useful when configuring polygon tools
- Clipboard - see screenshot on the previous page
  - Copy X - copies the x- cursor position to the clipboard
  - Copy Y - copies the y-cursor position to the clipboard
  - Copy point - copies x and y cursor position to the clipboard
    - can be used to paste the position into a tool
  - Remove last point - Ctrl-RMouse
  - Move point - Shift+Ctrl+RMouse - move selected point
    - it can be easier to remove and insert a new point than moving
  - Insert point - Shift+LMouse
  - Remove point - Shift+RMouse
  - Smooth - smoothes the clipboard polygon
  - Simplify - removes points on a straight line
  - Undo - will undo last operation
  - Closed polygon - will open or close polygon
  - Export - exports the polygon

Note: The image operations combined with image zoom are useful to define lines and regions of interest.

5.6.5 Point & Click Clipboard Support

Scorpion tools support copy and paste of ROIs (Region Of Interest) to and from the image. An ROI is managed by the tool's copy and paste buttons. *Copy* copies the ROI to the image from the Scorpion clipboard. *Paste* pastes the ROI from the image to the Scorpion clipboard. Rectangular ROIs are defined with four points. One point will change the center point. *Ctrl-Z* submits the ROI to the selected tool. More on *Copy* and *Paste* of ROIs in the Service-Toolbox chapter.



Right click the mouse above a tool's graphic and the tool is available in the menu. This is only possible in Service mode.

5.6.5 Save and Copy

It might be useful to document the changes and additions you have made to an image. Use either *Copy shown image with graphics* (copies to the clipboard) and paste it in for example a document, or *Save shown image with graphics* for saving to file. Likewise use *Copy/Save shown image without graphics* to copy or save the image view itself without the image graphics.
When working without graphics the raw image is transferred, with graphics, a copy of the screen is transferred.

5.6.6 Pointing precision and Panning

You can set the *Pointing precision* to be a number of pixels between 3 and 10.
When *Show Pan View* is selected and you zoom in, a pan window is shown at the upper left corner of the image.

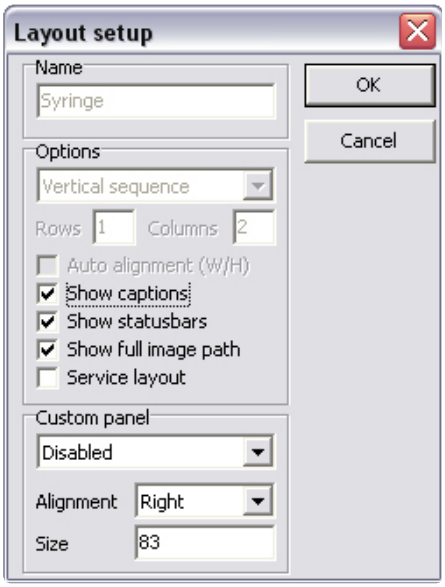
Auto Panning is useful if you work with polygons and have zoomed into the image. If you place a point outside the image border, the image view will automatically change to show the area where you pointed.

5.6.7 Help

Help activates the online help system.

5.6.8 Layout

Depending on the number of cameras in your system, you have one or more images to show in the image pane. *Layout* configures the image panes and is only available in Service mode. You can make as many panes as you want and choose the images to be shown in each pane.



Layout setup when you have one image in the image pane.

When you select *Layout - New*, a window like the one below opens. You give the pane a name and choose how it shall appear on the screen.

Images- defines the images to be shown in the pane

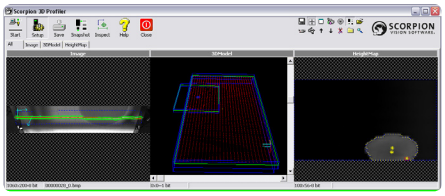
- *Source* - displays available images
- *Show* - sequence of selected images
- ← - removes selected image from the pane
- → - adds selected image to the pane

Options

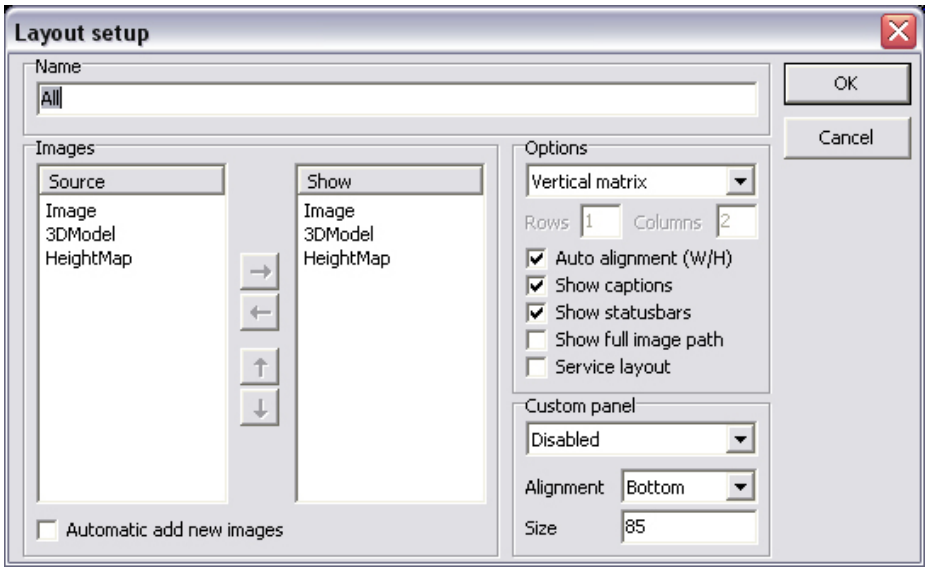
- Image presentation
  - *Horizontal sequence* - displays all images in one row
  - *Vertical sequence* - displays all images in one column
  - *Horizontal matrix* - displays the images in a matrix, more columns than rows
  - *Vertical matrix* - displays the images in a matrix, more rows than columns
- *Auto alignment* - select the best fit image view when changing the image mode
- *Show captions* - display image captions
- *Show statusbar* - show a statusbar with image name and size below each image
- *Show full image path* - display full image name path in the statusbar
- *Service layout* -

Custom panel - here you can decide the layout of an additional custom panel

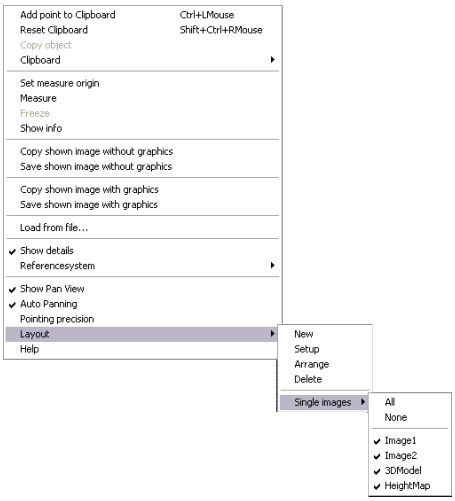
- *Disabled* | *Only in image mode* | *Only in normal mode* | *Always*
- *Alignment* - *Left* | *Top* | *Right* | *Bottom* - alignment relative to the image
- *Size* - given in pixels



Example with four image panes. The All pane shows 3 images.



Layout setup when you have more than one image in the image pane.
In this case three images named Image, 3DModel and HeightMap. You can choose which of the images you will see in the All pane.



The Layout menu

Right click the mouse over an image pane and select *Layout - Setup*. A window with the layout configuration for that pane opens. The window is similar to the one shown above.

Layout - Arrange opens a window where you can change the order of the images in a pane.

In *Layout - Single Images* you can choose which of the images to be shown in separate panes: *all*, *none* or you can select them by name. See the menu example to the left.

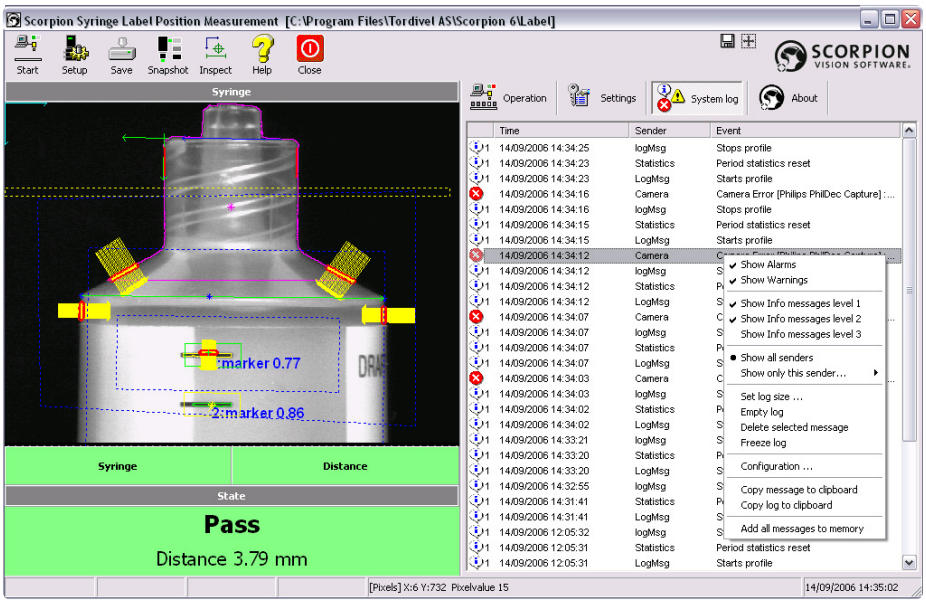
6 System log

The events are classified in five categories:

- *Alarm*
- *Warning*
- *Information level 1*
- *Information level 2*
- *Information level 3*

The system log menu has the following items:

- Show Alarms - activates display of alarm messages
- Show Warnings - activates display of warning messages
- Show Info messages level 1/2/3 - activates display of information message level 1, 2 or 3
- Show All Senders - display all senders - default
- Show only this sender - only selected source is visible
- Set log size - user defined log size is defined
- Empty Log
- Delete selected message
- Freeze log - stops updating system log
- Configuration - opens system log configuration - see below
- Copy message to clipboard
- Copy log to clipboard - copies all message to clipboard
- Add all messages to memory - add all levels to memory - this means that you will be able view message not visible when changing message visibility.



System events like quality alarms are shown in this window. By right clicking in the system log and selecting from the menu, the system log can be configured. The menu is shown in the example above.

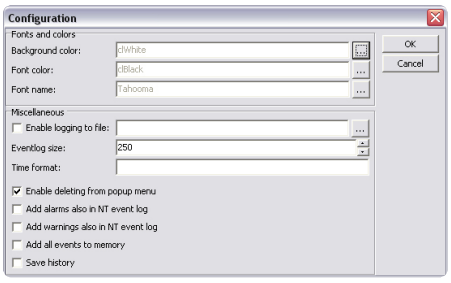
The Scorpion System log is important in verifying correct system operation.

When debugging and developing Scorpion profiles, it is recommended viewing all categories. In a running system all information needed is available in the three first levels. Below a screenshot of the system log is shown.

The first column of the log shows an icon identifying the event category, the next column is the time, the third the source or sender of the message and the fourth the description or actual message.

6.1 Configuration

Under the menu item Configuration, you find the following:



System log configuration

Note: Valid only for some browsers:
Upon closing the browser this dialog may disappear. Press Alt-Tab to locate the hidden dialog.

Fonts and colours

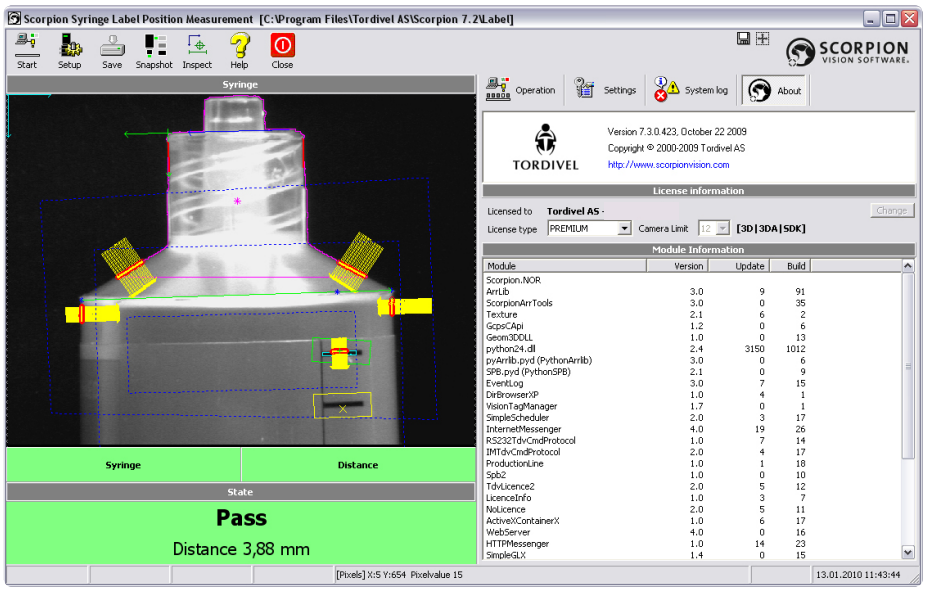
- Background colour - sets the background colour - clWhite is default
- Font colour - sets the foreground font colour
- Font name - sets the font

Miscellaneous

- Enable logging to file - activates file logging of all events - ... activates a file browser
- Event log size - sets the size of the event log file
- Time format - specifies the time-format in the Time column - can be useful to display ms to verify system timing
  - h: hour, m: minute, s: second, z : ms
  - D: day, M: month, Y: year
    - hh:mm:ss.zz DD/MMM/YYYY yields 16:34:52.22 30/JAN/2001
- Enable deleting from popup menu
- Add alarms also in NT event log
- Add warnings also in NT event log
- Add all events to memory
- Save history - will save history when terminating Scorpion making the system log persistent

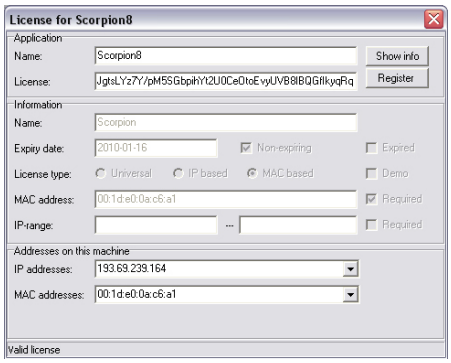
7 About

Here you find information about the Scorpion version and program components contained.



System information

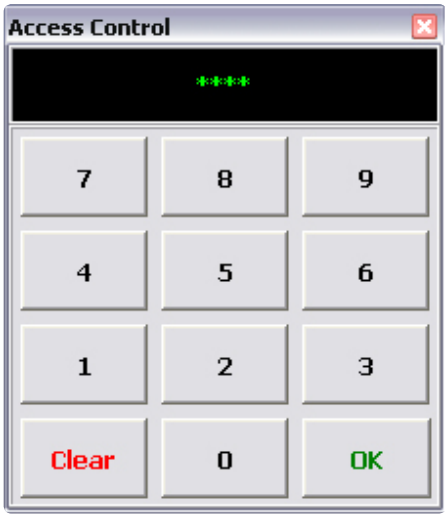
Under *License information* you see the type of license that is valid on your computer. Press the *Change* button when the license needs modification. You see i.e. the license string itself and the expiry data.



License information

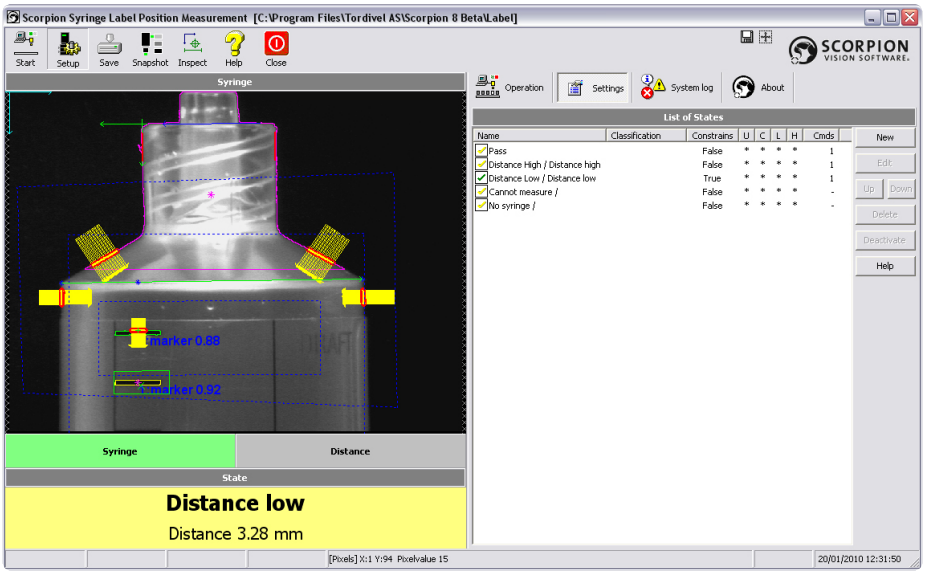
8 Settings

The system has two different pair of settings. One for trained operators and the other for authorized service operators. Select the Setup button on the upper left of the screen, give the PIN code for settings and the first category is shown.



The settings are protected by a PIN code, and cannot be changed until the correct code is given. The settings can however be read without applying the code.

Buttons and menus for system configuration are shown in the main window when selecting Settings. The buttons *Snapshot* and *Inspection* are used to take an image and inspect this for instance at system verification, configuration or on manual operation. After system configuration you can also choose an image in the image history and select *Inspection* to run a new inspection.
By selecting the *Image* button, images are saved. By using these buttons, a set of images for test purposes can easily be generated.



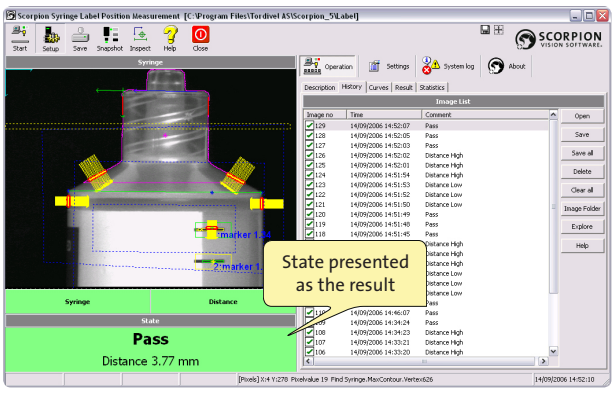
Main window when configuring the settings.

8.1 States

In Scorpion system states are used to classify the result of an inspection. In an identification system the states are typically the identified unit or product. In assembly verification they can for example be pass, fail, no product or cannot measure.

The state is presented as the inspection result in the History - Image list and in the Inspection result panel at the lower left of the screen.

The states themselves are defined in an ordered list. The condition is updated as the inspections are processed.



Each state can be activated or deactivated. You can also copy a state by selecting it in the list, right click the mouse, choose Copy from the menu and Paste it either in another state or as a new one. The Delete button deletes the selected state.

8.1.1 General

The state dialog consists of three pages:

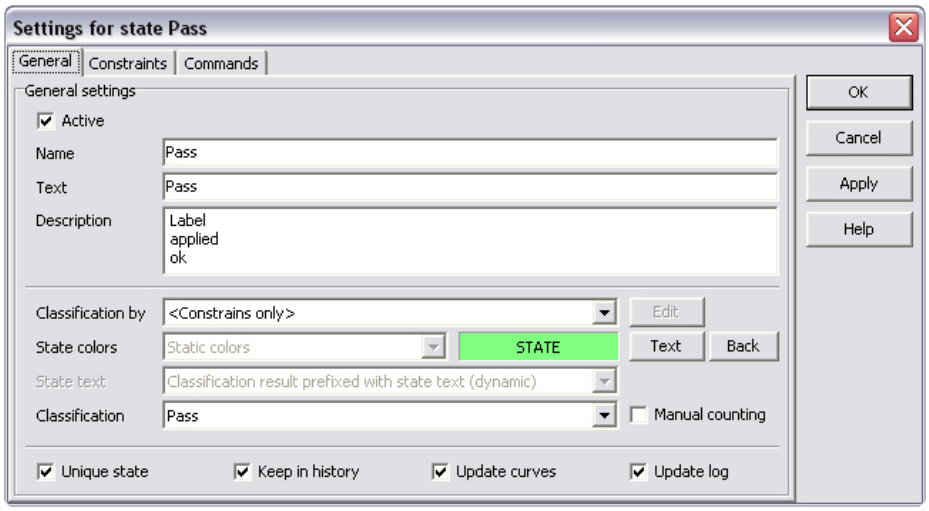
General page
- Name
- Description
- Foreground and background colour
  - Used in the Inspection result panel

Constraints page
- The constraints page defines when a state's condition is true or false

Command sequence page
- The command sequence is executed when the state is true

Click *New* to add a new state and you see the General page. You give the state a name, and it appears in the list. Double click the name or select *Edit* and the *Settings for 'State'* panel is shown. Associate a colour to the state - this will illustrate the inspection result in the *Inspection result* panel. The criteria used to define the state are then given. The combination of these criteria defines a state. These are general properties in addition to state constraints. Commands to give an action if a state occurs can additionally be given.

There are five states defined in our example "Label on Syringe". We will show how the 'Pass' state is defined.



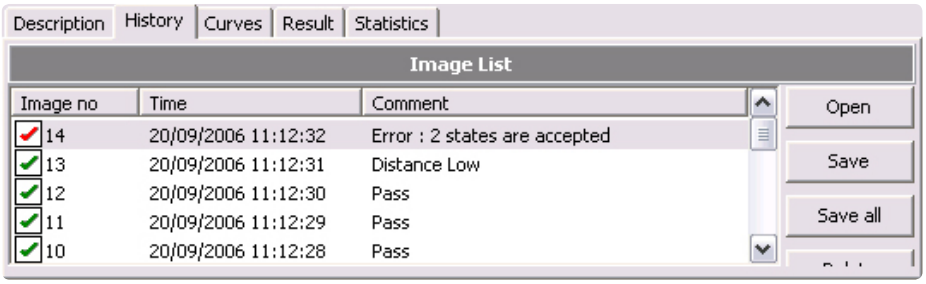
Defining the Pass state

Colour selector

- OK - closes the dialog
- Cancel - closes the dialog and cancels changes
- Apply - applies changes without closing the dialog
- Help - activates the State Help pages

The condition's colour is changed by clicking the colour square. The colour is selected using the Colour selector.

The state is defined to be unique, that means Scorpion will indicate an error if 'Pass' occurs at the same time as another state. If this happens, Scorpion will indicate an error by red in the image list, as shown in the example below.



Two states occur at the same time - there is an error in the state definition.

***Hint**: if you are analysing a huge amount of images it can be wise only to keep the ones showing problems. Select only 'Keep in history' for states classifying errors.*

If the state is not unique, an inspected unit can be accepted by more than one state. The state highest in the state list is shown in the *Inspection result* panel. (The list can be sorted using the *Up* and *Down* buttons.) All commands related to the true states will however be run.

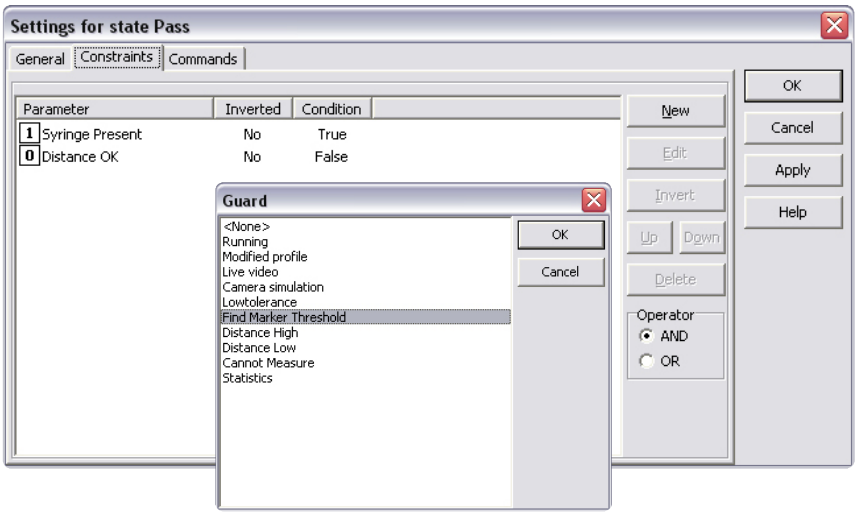
You can choose if logs and curves are to be updated when a state is accepted. In our example the state 'Pass' will update the logs and curves since measured values are relevant when the state occurs. The 'Can not measure' and 'No syringe' states will however not update the logs and curves with any values. A 'Pass' result will be kept in the history list when 'Keep in history' is marked.

8.1.2 Constraints

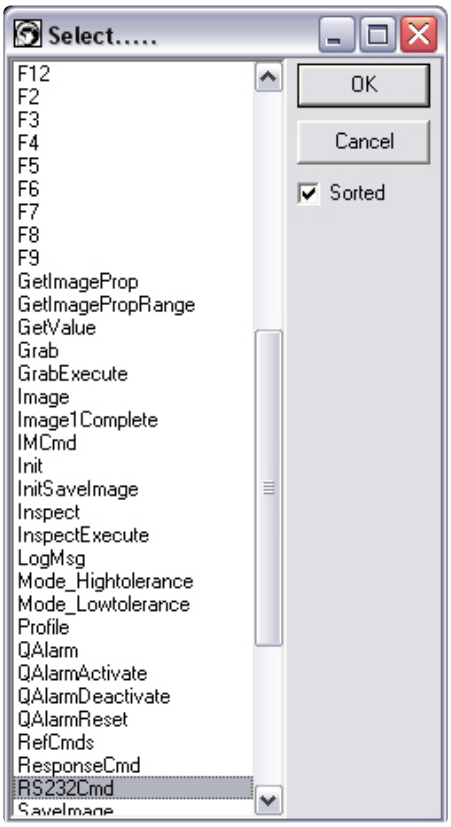
In the Constraints panel you can add logical expressions and combinations of such. All results from logical tools can be used to define constraints.

Add new constraints by pressing *New* and choosing a logical tool.

In our example both the result of the *Syringe present* and *Distance OK* tools have to be true at the same time. The tool constraints are defined in the Toolbox. You can make extensive and complicated expressions by combining results of logical tools and states.



Constraints for state Pass.



The command browser is activated pressing the (…) button in the Command field. The browser contains all system and user defined commands.

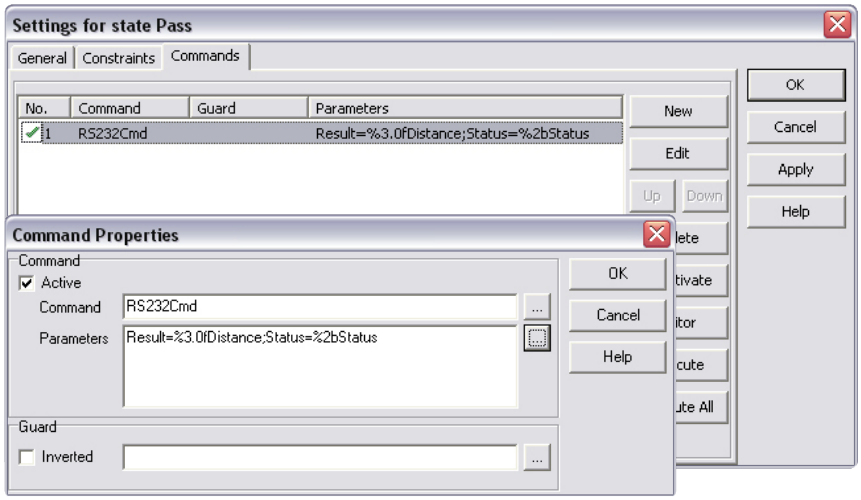
8.1.3 Command sequence

The Command Sequence is executed when the inspection leaves the state true.

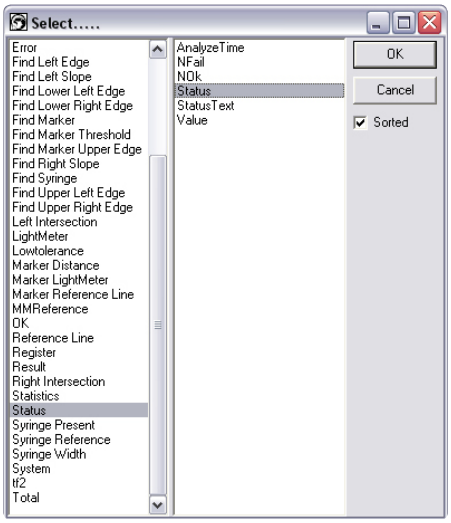
To immediately run the command, select the respective command line and press *Execute*. To run the complete command sequence, press *Execute All*.

In the example Scorpion is configured to send a response over rs-232. The Distance and Status names you see in the parameter strings are new names defined in the Alias manager. They represent the Result.Value and Status.Value tool parameters respectively. (See chapter Service - Alias.)

To define the expression in an editor, select the *Editor* button.



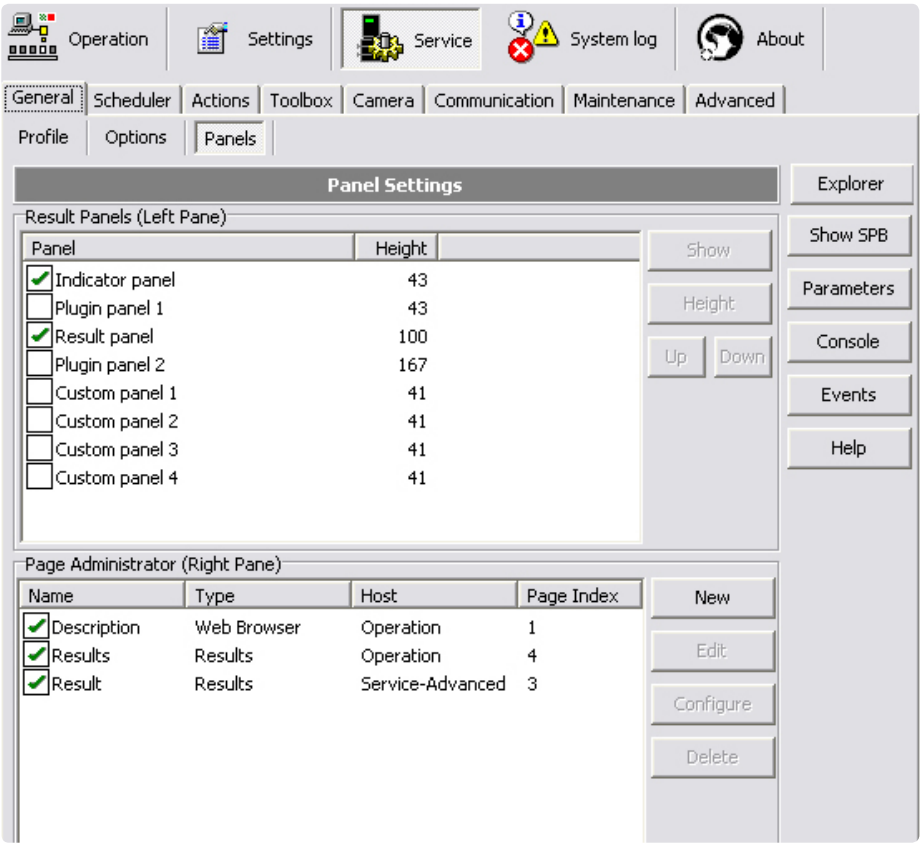
Command sequence for state Pass.



The parameter browser is activated pressing the (…) button in the Parameters field. The parameter browser contains system parameters and the results of all tools defined in the toolbox.

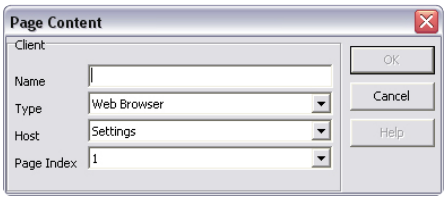
8.2 Web Browser

You can include a web page in the Settings panel with e.g. a description of the settings. Under *Page Administrator* in the Service - General - Panels page you can include a new page.

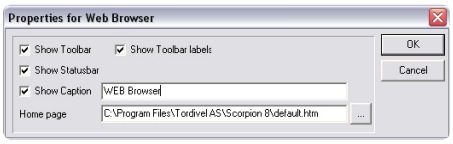


Panel settings

Press *New* and fill in the Page Content panel coming up. Choose *Web Browser* as Type and *Settings* as the Host; give the page a name, press *OK* and a web browser will show up in the Settings panel.



Press the *Configure* button to decide which page to show.



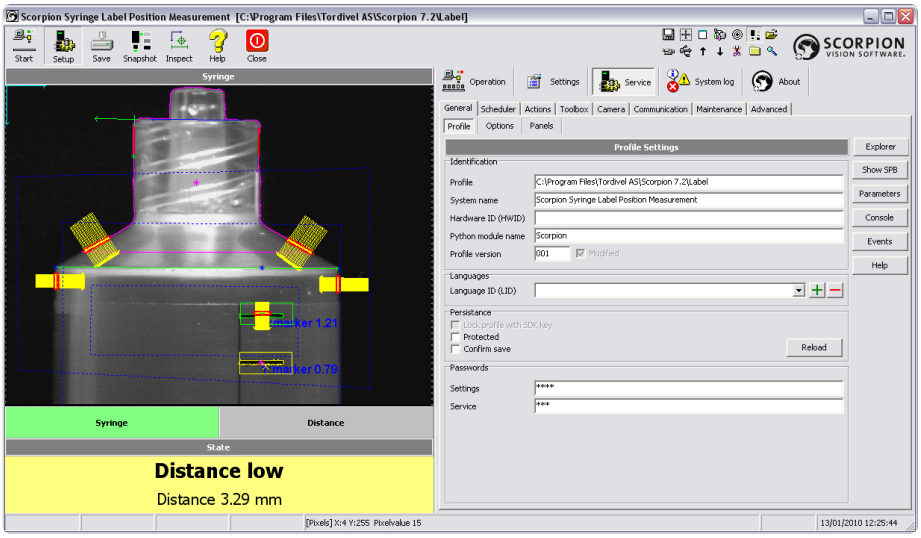
Under *Properties* in the web page's tool panel you can, if you are authorized, change the page setup. You can decide which page to be the home page and if the tool panel with buttons and text, status line and page title shall be shown.

You make the page content with an editor, for example Microsoft FrontPage. You can relate Scorpion's commands and parameters to buttons and boxes on the web page.

9 Service

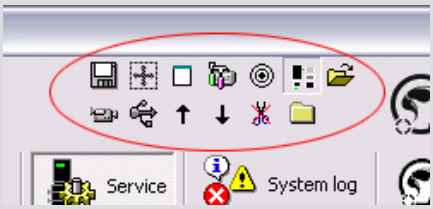
The service settings are only available for authorized service operators and are hidden by a PIN code. The code is different from the Settings code. The Service code unlocks however also the Settings panel. When the PIN code is correctly entered, the Service button is visible on the right side operation panel.

Each service panel is shortly described in this chapter. Use of the panels requires however detailed information and training above the scope of this user manual.



Main window in Service mode

Shortcut Symbols



The Service toolbar provides convenient shortcuts for often used functions.
Move the mouse over the symbol and a descriptive text is shown.

- Save current image to disk
- Full image mode - hides right pane
- Show/Hide console window
- Go to Toolbox
- Go to Central
- Camera simulation
- Open simulation folder
- Activate/Deactivate live video
- Reset Camera list – applicable in service mode
- Previous image in history list
- Next image in history list
- Reset clipboard
- Open Explorer in profile folder

9.1 General

Here you find the general settings for the user interface and system behaviour.

With the buttons to the right you can check the system configuration and status. To get an overview of the file structure, press *Explorer* and the Windows Explorer is opened. To see the system configuration file (SPB) in an editor, press *Show SPB*. The *Parameters* button opens a Browser with all the parameters generated by the system. The *Console* button opens the console window. The *Events* button opens an event tracer window. *Help* activates the help pages.

9.1.1 Profile

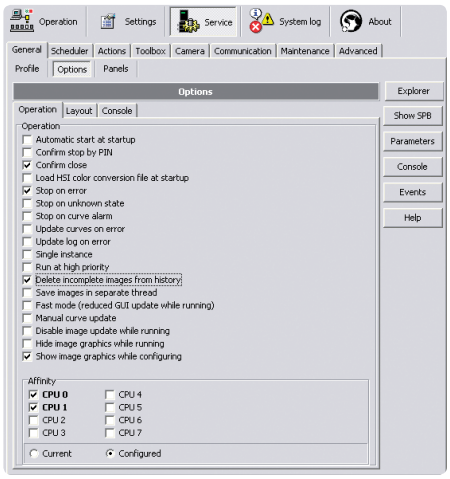
In the Profile panel you can set the profile, system and project names and the profile version. See the image above for an example.

- Profile - the current profile directory path
- System name - profile name
- Project - the project name - note: it is not possible to change this item
- Python module name - note: it is not possible to change the item
- Profile version - the profile version - the version is automatically incremented when a maintenance backup is performed
- Modified - checked when the profile is changed
- Persistance - choose Protected or Confirm save. Reload button available.

You can also change the Settings and Service passwords.

9.1.2 Options

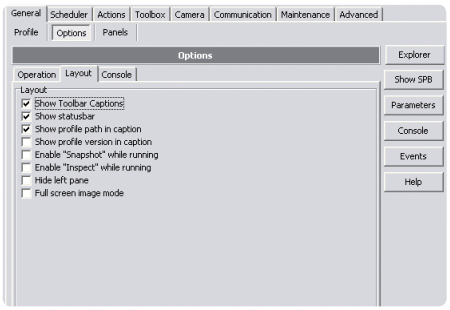
The options are used to configure the application behaviour in detail. The options are divided into three categories.



Operational options

Under *Operation* you can do the following:

- Automatic start at start up - if set, the inspection automatically starts when Scorpion is started.
- Confirm stop by PIN - you are asked to confirm termination by giving the PIN code.
- Confirm close - gives a warning when you terminate the program. You are asked to confirm.
- Load HIS colour conversion file at start up - valid when colour images are used. Loads colour lookup table when starting Scorpion. Loading of this file takes time, thus it is timesaving for the image analysis to load the file at start up.
- Stop on error - if selected, Scorpion stops if an error occurs.
- Stop on unknown state - if selected, Scorpion stops on an unknown state.
- Stop on curve alarm - if selected, Scorpion stops on curve alarm.
- Update curves on error - if selected, curves with inspection data is updated also on processing error. Normally you don't want the curves updated with "noisy" data.
- Update log on error - if selected, the data log is updated with inspection data also on processing error. (Event messages are independently of this put in the system log.)
- Single instance - system global state. When selected only one instance of Scorpion is started. Recommended used in factory environments to avoid multiple Scorpions being started by accident.
- Run at high priority
- Delete incomplete images from histrory
- Save images in separate thread
- Fast mode (reduced GUI update while running)
- Manual curve update
- Disable image update while running
- Hide image graphics while running
- Show image graphics while configuring
- Affinity - current and configured



Layout options

Under *Layout* you can do the following:

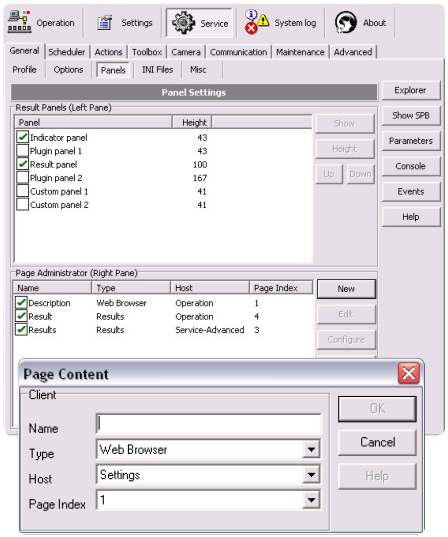
- Show Toolbar captions - if selected, the toolbar captions are shown.
- Show status bar - if selected, the status bar is shown.
- Show profile path in caption - if selected, the path to the profile is shown in the main window caption.
- Show profile version in caption - if selected, the profile version is shown in the main window caption.
- Enable "Snapshot" while running
- Enable "Inspect" while running
- Hide left pane
- Full screen image mode - if selected, only the image part of the screen is seen.

Console window options

Under *Console Window* you can do the following:

- Show console messages in system log
- Show console window at start up
- Always on top - if selected, the console window is always on top on the screen.
- Show Arrlib messages - shows messages from the library of image processing algorithms

9.1.3 Panels



Panel settings and page content definition

Under the *Result Panels* settings you find a list of optional user interface panels. Select them and press the *Show* button to make them available. Select them and press *Hide* to remove them from the screen. Normally you want to see the:

- Indicator panel - if selected, the indicator panel is shown at the lower left side of the screen. The panel indicates which error that has occurred if a unit is rejected.
- Result panel - if selected, the result panel is shown at the lower left side of the screen. Here you see the result of a classification.

Plugin panels are defined under Advanced - Central - Plugins. You include them in the screen by selecting them and pressing *Show*.

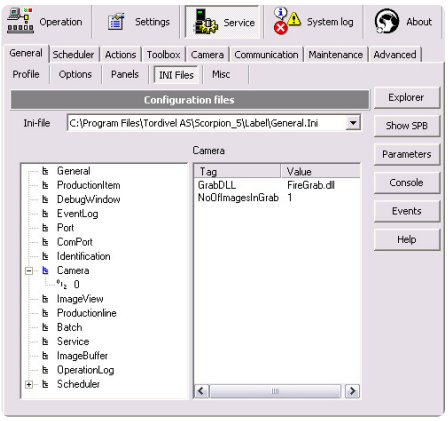
Under *Page Administrator* you can add additional detailed panels, like web browsers and result panels.

Press *New* and fill in the *Page Content* panel coming up.

Host decides where in the panel structure your page will show up; in the Operation, Settings, Service or Service-Advanced panels. The panel *Type* can either be Data Input, Web Browser or Result. Give the page a name, press OK and the page is included.

Press the *Configure* button to set the home for the web browser.

9.1.4 INI files



INI files

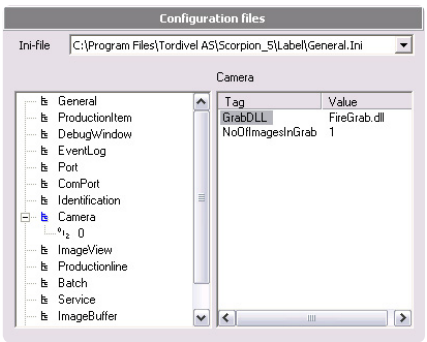
This panel is used to get an overview or tune a large number of different parameters. The parameters are hierarchically structured as seen in the example below. Many of the parameters found here you can also find in other service and settings panels. Operating on the INI (Initialisation) files is an alternative way of configuring and managing the system.

Note: This panel is normally not being used.

Changing the Camera Interface DLL

This panel is used to change the camera interface dll.

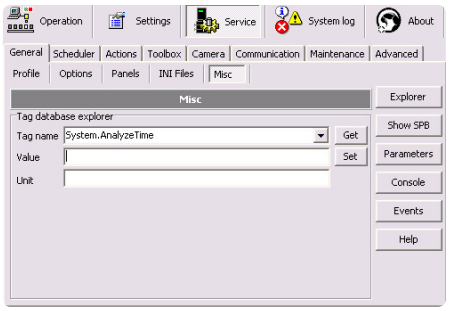
- Select General.Ini
- Select Camera



- Edit the GrabDLL entry by double-clicking

- Change the name of the interface DLL
- After changing the DLL, Scorpion must be restarted.

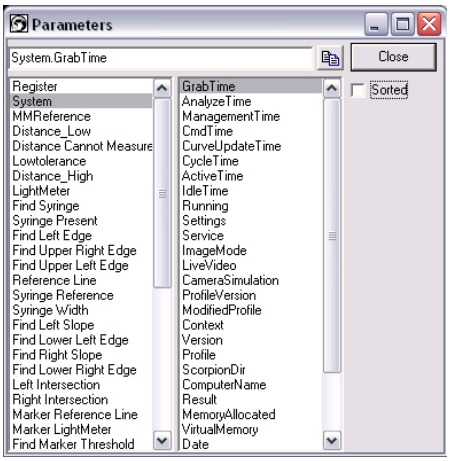
9.1.5 Misc



Tag database explorer

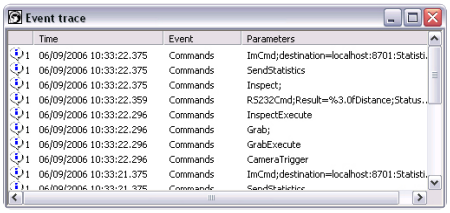
Under Misc (Miscellaneous) you can set or get values in the Scorpion tag database.

Hint: use the *Parameters* button to select the tag name.



List of system parameters

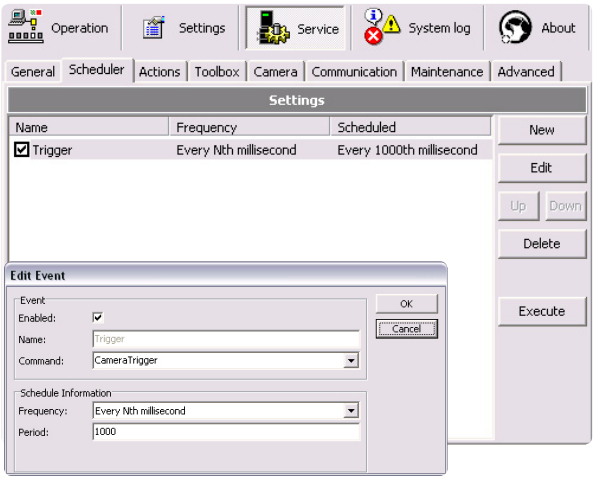
Press *Events* to see an event trace.



Event trace

9.2 Scheduler

Here you find tasks that are to be automatically run at scheduled intervals. The tasks can be activated or deactivated by selecting the box in front of their name.

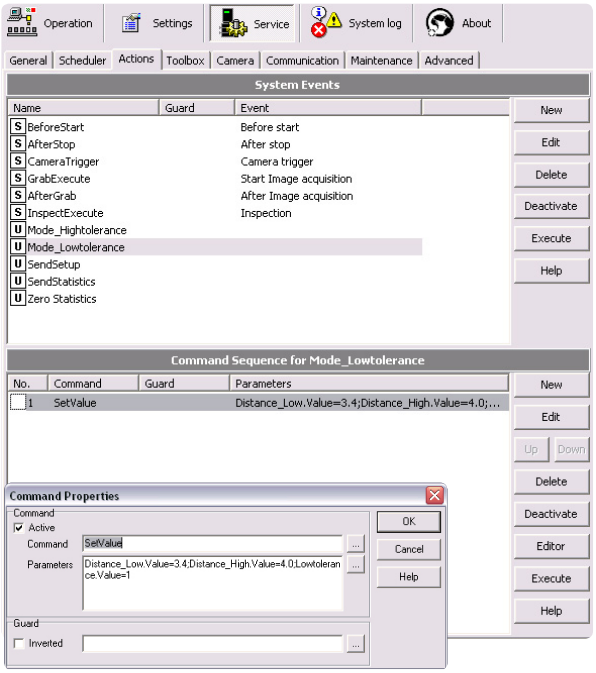


Scheduled task

The commands described in chapter Service-Actions can be used here, thus run repetitively and scheduled.
In the left example the 'CameraTrigger' command is set to trigger the camera every second.

Hint: Use the *SaveImage* command to save an image to file from time to time. These pictures can later be run as live video, thus you can easily see if e.g. the light conditions have changed over time. You find live video under Service-Camera.

9.3 Actions

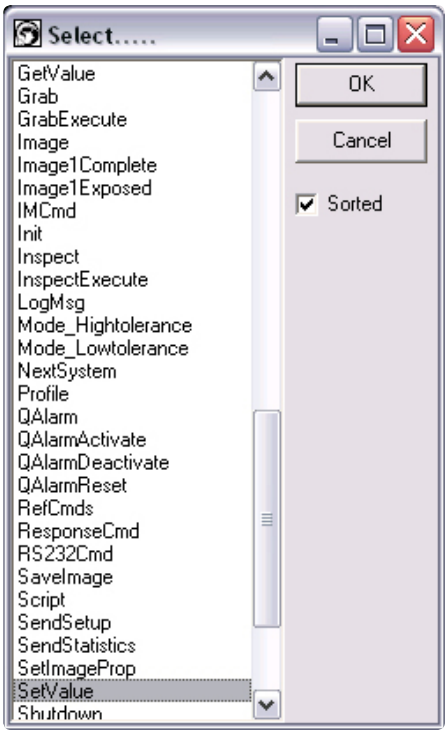


Actions - command sequences at different system events

To adapt the system to your needs, Scorpion has defined a set of commands to be used at system events. For each system event, you can define a command sequence to be run when the event occurs. The commands can also be scheduled to run repetitively. See chapter Service-Scheduler. You find an overview of the Scorpion system events and commands in the System events and Commands chapters.

A system event can be defined by Scorpion (marked with **S**) or by the user (marked with **U**). Commands from external systems are typical examples of user defined system events.
In our example "Label on Syringe", there are two specially defined system events: *Mode_Lowtolerance* and *Mode_Hightolerance*. For each system event a command sequence to be run when the event occurs, is defined. All commands are available from the Scheduler or from the external rs-232 or tcp/ip interface.

Choose a system event, press *New* under *Command sequence for 'system event'* and the small window above to the left will appear. Here you add the command and eventual parameters. In the example above, the system low tolerance limits are given. In the *Guard* field, you can give the name of a logical or script tool. The command will then only be run if the result of this tool is true (=1). If you select the *INV* box, the command is run only if the result is false (=0). Press the (…)-button, and you find available commands (window left below) and guards.



Available commands

You can run a selected command immediately by pressing the *Execute* button. Press the *Execute All* button and the whole command sequence defined for a system event is run.

For information on the rs-232, tcp/ip and Profibus interfaces see the Communication chapter.

Error messages are sent to the System log.

The system defined events may also be called from any other event, either by the localized name or by the internal name. Using the internal name will always work when transferring profiles between computers with different locale settings.

Actions - Label on Syringe

In our example "Label on Syringe" there is two user defined system events: *Mode_Hightolerance* and *Mode_Lowtolerance*. Both *Mode_Hightolerance* and *Mode_Lowtolerance* consist of three SetValue commands. They set tool values in the toolbox.

The command sequence for *Mode_Lowtolerance* consists of the following commands:

1. Set lower limit	SetValue;Distance_Low.Value=3,4
2. Set upper limit	SetValue;Distance_High.Value=4,0
3. Set mode	SetValue;Lowtolerance.Value=1

The command sequence for *Mode_Hightolerance* consists of the following commands:

1. Set lower limit	SetValue; Distance_Low.Value =3,6
2. Set upper limit	SetValue;Distance_High.Value=3,85
3. Set mode	SetValue;Lowtolerance.Value=0

Note that the command sequences use the tool parameters.

9.4Toolbox

See the online help files for details on how to use each tool.

The image analysis in Scorpion is performed by a toolbox of user configurable tools. You find a rich variety of tools in Scorpion. These are image processing tools in addition to mathematical and logical tools. They are grouped in six categories: Basic, Data, Edge, Geometry, Reference, 3D and Advanced tools. The tools are rather simple, but put together they solve very complicated tasks.

An image analysing tool is used to make a calculation. When configuring a vision system, you decide which tools to use and set their parameter values. The parameters define the tool set-up and are typically coordinates, search areas (ROI - Region Of Interest), reference points, min/max values, etc.

When run, a tool generates a result given as one or more values in addition to a set of graphical elements for visualization. The measuring result is used to define the measured objects state or status, which again decides if an action is to be taken. The visualization elements are used to illustrate the Scorpion measurement in the camera image. Each element is given a colour.

Additionally to the image analyses tools, there are other tools used to further process the analyses results. Two tools of importance are the logical and Python tools. The Logic tool classifies results from a set of image analyses tools. Python ensures maximum system flexibility.

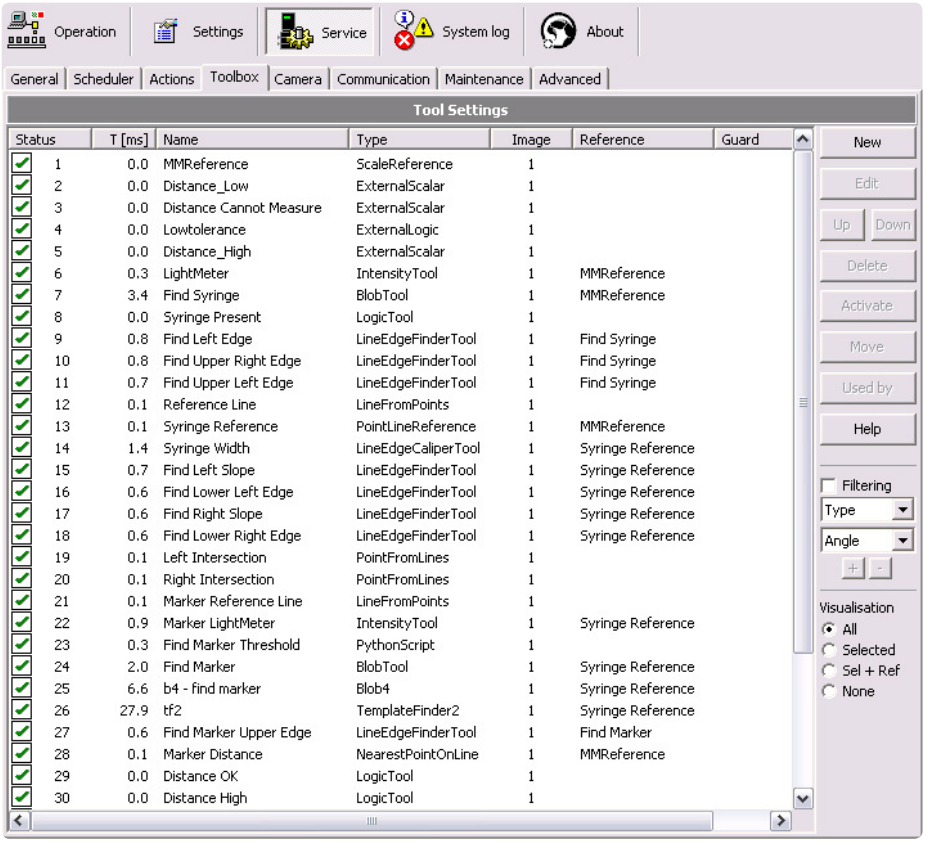
Scorpion Vision Software is a complete 3D machine vision platform. The toolbox has more than 40 tools solving 3D vision tasks.
Important features are:

- *Integrated 3D Visualization and 3D Images - point cloud support*
- *Powerful 3D reference systems - intuitive, convenient and easy to use*
- *Seamless 3D integration enables high precision 3D measurement using 2D image processing tools*
- *Advanced PlaneFit3D and CylinderFit3D establish reference systems based on point clouds*
- *Stereo Vision using from 2 to 4 cameras or images*

See the online help pages for details.

9.4.1 The Tool Settings window

The toolbox consists of an ordered sequence of tools. You can work with a tool by selecting it and using the buttons or you can right click the mouse over the tool and choose operations from the menu.



The toolbox consists of an ordered sequence of named tools. They are connected to an image and an optional reference system.

A system can use several different images in the identification process. The 'Image' column shows which of them the particular tool is operating on. In our example there is only one image.

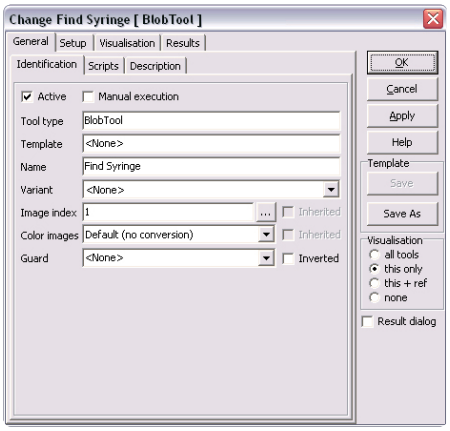
In the 'Reference' column you find the name of a tool used as reference for the selected tool. Under 'Guard' you can name a tool that must be successfully run prior to the execution of the selected one.

The icon in front of a tool's name indicates it's state after an inspection. The icon definition is as follows:

- *Not run*
- *Ok*
- *Blocked by guard or reference*
- *Error or No result*
- *Not active*
- *The license is not covering the use of this tool*
- *Manual execution*

9.4.2 Common tool elements

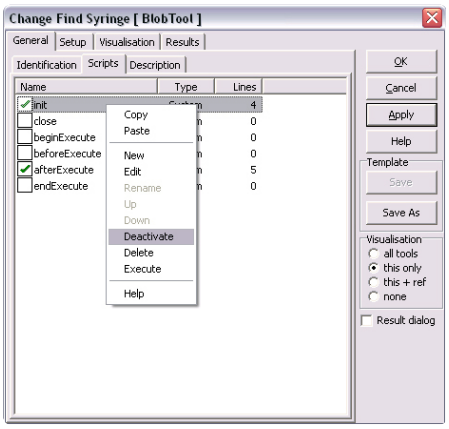
9.4.2.1 General



The General tab consists of the following parts:

- *Active* - activates or deactivates a tool
- *Manual execution* - sets the tool for manual execution from python
- *Tool type* - displays the tool type - read only
- *Name* - the name given by the user of the tool instance
  - The tool can be renamed using the toolbox mouse menu
- *Image Index* - the tool is executed on the selected image index
  - Press the ... button to select a named image
- *Color images* - when working with color images one can select to work on the Hue, Intensity or Saturation Color Plane
- *Guard*
  - Press the ... button to select a named logic tool as a guard
    Note: Expressions can be used as guards: GetValue('Scalar.Value') > 5
- *Description* - a free text tool description
- *Scripts* - System Defined scripts associated with each tool

Tool Scripts



All tools in the toolbox can be customized using system defined tool events and executed by user defined Python scripts. The scripts are available from the General tab of all tools.

The events are called by the tool list object. This is a very powerful feature that can substantially reduce the length of the toolbox. The script's mouse menu is shown to the left.

When a tool activates a script, the tool list instantiates a 'hidden' Python class in the Python namespace for the actual tool. The class definition is given by the activated scripts. This class has a member 'name' which can be used within the scripts to get access to the python tool object, *tool=GetTool(self.name)*.
This 'hidden' class is a standard python object, you can add members and methods as for any standard Python class by using the self argument which always must be the first argument, *self.mymember=10*.

- *Copy* - copies script to clipboard
- *Paste* - pastes script from clipboard to selected script
- *New* - creates a user defined script
- *Edit* - opens script editor
- *Rename* - renames user defined script
- *Up* - moves script up
- *Down* - moves script down
- *Deactivate* - deactivates selected script
- *Delete* - deletes script - removes user defined script | empties system defined script
- *Help* activates scripting help

Method	Description
init	Called at tool creation and when the user applies changes to any script. Useful to customize the python tool instance. The init method is paired with the close method.
close	Called at tool destruction and when the user applies changes to any script. May be used for cleanup. The close method is paired with the init method.
beginExecute	Called once before the actual tool execution. In this method it is possible to write its own processing algorithms, iterate itself or any kind of processing. By returning 1 from this method the default tool execution will not be executed. This method is also useful for setting up internal states and parameters before image processing.
beforeExecute	Called just before tool execution, also when iterating the tool from Python.
afterExecute	Called just after tool execution, also when iterating the tool from Python. By returning 0 the tool will execute again until afterExecute returns 1. Sometimes useful for result validation. Special care should be taken to avoid entering a endless loop (by always returning 0)
endExecute	Called at last in tool execution, enables to collect results, cleanup etc. This method is called only once for each tool in toolbox execution.

A tool normally consists of the following elements:

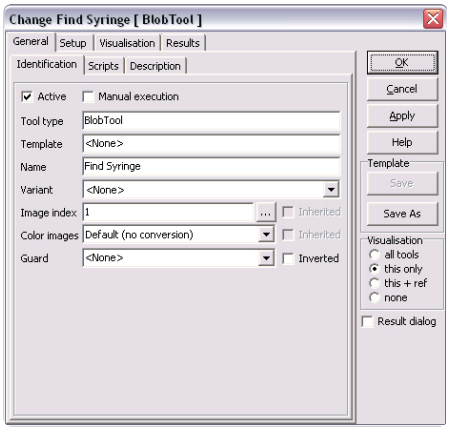
General
Buttons
Setup
Visualisation
Results
ExecuteCmd

Example: Iterate itself

```
def init(self):
  self.count=0                    #define a local counter
  self.max=12                     #define a local max count

def beginExecute(self):
  self.count=0                    #reset counter
  t=GetTool(self.name)            #get the python tool instance for myself
  img=GetImageMatr('Box')         #get the image to process
  ResetStatistics()               #userdefined method for statistics
  for i in range(self.max):
    SetROI()                      #userdefined method for settng ROI
    t.execute(img)                #execute the tool at new location
    UpdateStatistics()            #userdefined method for collect results and update statistics
  return 1                        #abort default processing/execution
```

9.4.2.2 Buttons



These buttons are present in all tools.

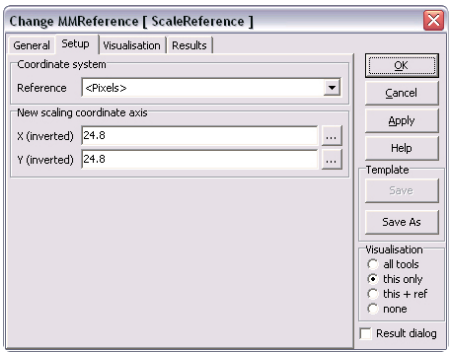
- *OK* - will accept changes and close the tool dialog
- *Cancel* - will cancel changes and close dialog
- *Apply* - will accept changes and perform an inspection while the system is in non-running mode
- *Console* - will toggle the console window
- *Help* - will activate the help pages for this tool
- *Save* or *Save As* - saves the tool as a Template. You can save it either as a local or shared template.

Visualisation - in image viewer

- *all tools* - visualises all tools
- *this only* - visualises only this tool
- *this + ref* - visualises this tool and it's references
- *none* - turns off tool visualisation

Note: turning off tool visualisation is handy when editing polygons in the image.

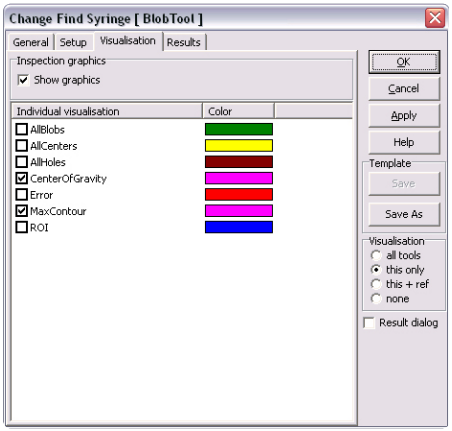
Check the *Result dialog* to open a separate window showing the tool results.



9.4.2.3 Setup

The optional Setup page is present in most tools. The ScaleReference example is a simple but typical Setup page:

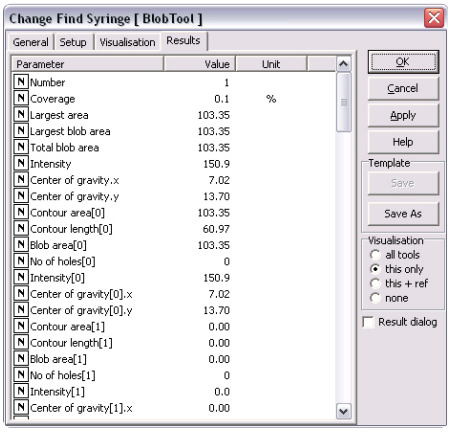
- Reference - user defined reference system
- New scaling coordinate axis - X and Y scale is defined



9.4.2.4 Visualisation

For adding graphics on the image and visualising the operation of the tool. Visualisation of all or this tool only can be selected.

Note: Turning off Show Graphics will hide the tool's visualisation unless the tool is active or explicitly set to visualise in the Visualisation Group box.



9.4.2.5 Results

All tools have a Result page that displays all parameter results of the tool.
Manually it is possible to set the unit and the precision of each parameter.

Hint: Activate the Results menu by right-clicking the parameters to set unit and number of decimals.

Results definition

Parameter	Description
<parameter>	Numeric or text data, depends of the tool type
Status	Tool execution status. 0=not executed, 1=executed, 2=guarded, 3=error, 4=deactivated
Analysis time	Tool execution time in ms

9.4.2.6 ExecuteCmd

The executeCmd tool interface is available in most Scorpion tools. This enables selected operations to be performed from a Python script. An overall description is given here; refer to each tool's help page for details and available commands.

Command format

ok,ret = <tool>.executeCmd('<command>','<parameter>=<value>;...')
ok,ret = <tool>.executeCmd('<command>;<parameter>=<value>;...')

<command> and <parameter>, and also <value> if it contains an object name (e.g., ROI) are case insensitive.

Common commands

The Set and Get commands have a standard format; see the examples below. "SOURCE", "DESTINATION" and "CLIPBOARD" are reserved words, case insensitive.

Return value

This is always a tuple, where the first element indicates success (1) or failure (0). The second element depends of the tool type and command.

Availability

The interface is always available, but note the following: When the tool's configuration dialogue is open, changes made to the tool from the executeCmd interface may be lost when the dialogue is closed. If you press OK or Apply, the changes are overwritten with the configurations dialogue's data. (ROI settings are always remembered.)

Example 1: Set ROI

```
tool = GetTool('Combiner') #Get a handle to an ImageCombiner tool

#Set ROI of tool
tool.executeCmd('Set','object=ROI;value=100,100,50,50')
```

Example 2: Set ROI from clipboard

```
tool = GetTool('Combiner') #Get a handle to an ImageCombiner tool

#Set ROI of tool using user defined points on clipboard
tool.executeCmd('Set','object=roi;source=clipboard')
```

Example 3: Copy Polygon ROI to clipboard

```
tool = GetTool('Combiner') #Get a handle to an ImageCombiner tool

res,ROI=tool.executeCmd('GET','OBJECT=ROI') # get tool's roi
res=tool.executeCmd('Get;OBJECT=ROI;destination=clipboard')
```

Example 4: Execute a tool specific command - Add a reference image

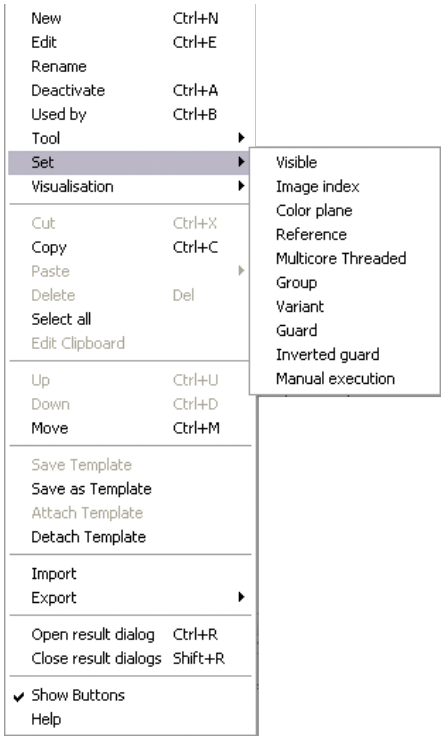
```
tool = GetTool('Combiner') #Get a handle to an ImageCombiner tool

tool.executeCmd('ADDIMAGE')
```

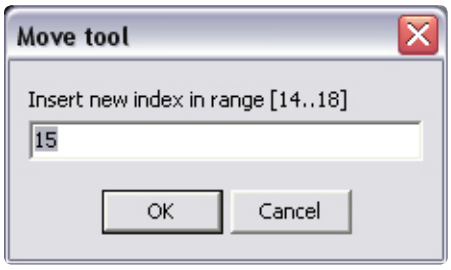
9.4.3 Tool operations

The following menu items are available in the toolbox window:

- *New* - Ctrl+N - creates a new tool
- *Edit* - Ctrl+E - edits selected tool
- *Rename* - renames selected tool
- *Activate/Deactivate* - Ctrl+A
- *Used by* - Ctrl+B - shows all tools referencing this tool
- Tool - Ctrl+Z - sets ROI
- *Set* - sets common properties for a set of selected tools. The following set operations are available:
  - *Visible*
  - *Image index*
  - *Color plane*
  - *Reference*
  - *Multicore Threaded*
  - *Group*
  - *Variant*
  - *Guard*
  - *Inverted guard*
  - *Manual execution*
- *Visualisation* - determines which tool results you want displayed in the image
  - *All* - graphic results from all tools
  - *Selected* - only results from the selected one
  - *Sel+Ref* - results from the selected one and it's references
  - *None* - no graphic results are seen in the image
- *Cut* - Ctrl+X - cuts the selected tools
- *Copy* - Ctrl+C - copies the selected tools
- *Paste* - paste the tools on the clipboard
  - Copy the selected tool
  - Copy the configuration from the selected tool - overwrite the selected tool
  - Copy the configuration from the selected tool without reference - overwrite the selected tools with the exception of the reference
  - When multiple tools are selected paste is not allowed to overwrite existing tools - use Edit clipboard to change the names
  - It is possible to edit the clipboard using Notepad and copy tools from another Scorpion profile running on the same computer
- *Delete* - Del - Deletes the selected tools
  - It is not possible to delete tools that are connected to other tools not being deleted
- *Select all* - selects all tools in the toolbox
- *Edit clipboard* - activates toolbox clipboard editor - can be used to rename tools before pasting into the profile
- *Up* - Ctrl+U - moves the selected tools up
  - it is not legal to move a tool on top of a tool that it depends on
- *Down* - Ctrl+D - moves the selected tools down
- *Move* - Ctrl+M - moves the selected tools, give new position in window coming up
- *Save Template* - saves the tool as a template
- *Save as Template* - saves the tool as a new template
- *Attach Template* - attach a template to the tool
- *Detach Template* - detach the template used
- *Import* - Ctrl+I - import a set of tools save to file
  - Import will not overwrite existing tools - to replace the whole toolbox - delete the tools that shall be imported before using import
- *Export* - Exports the selected tool to a SPB-XML file
- *Export all* - Exports all tool to a SPB-XML file
- *Open result dialog* - Ctrl R
- *Close result dialog* - Shift R
- *Show Buttons* - display the right hand buttons in the toolbox
- *Help* - Activates the html help file



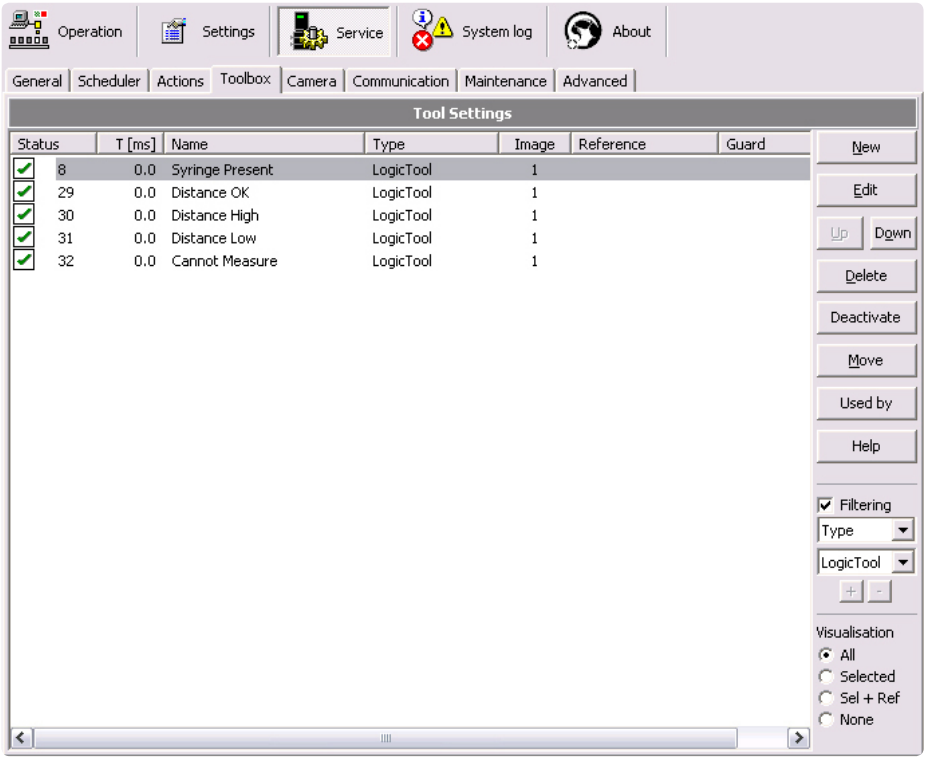
The Set menu



Move tool - give new position in list

9.4.3.1 Filtering

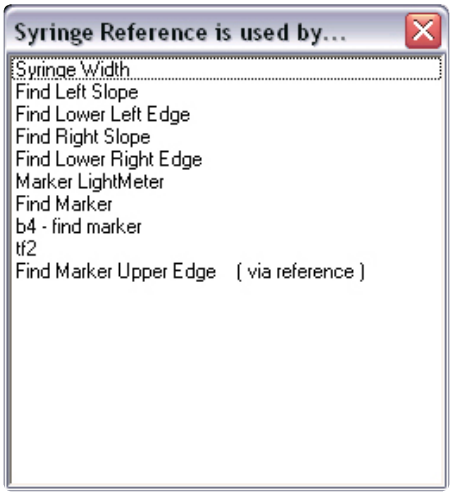
To limit the number of tools shown in the list, you can use filtering – available at the lower right of the Tool Settings window. You can filter based on tool name, type of tool, image, reference or guard. In the below example we show only the tools of type LogicTool.



Filtering the list of tools

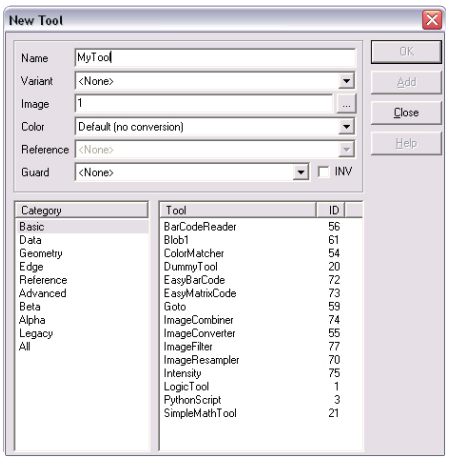
Select a tool and press the *Used by* button, and you get a list of other tools using this tool's results in their calculations. You are e.g. not allowed to delete a tool if other tools base their calculations on it. Below you see the tools using 'Syringe Reference'.

At the lower right of the Tool Settings panel you find the *Visualisation – All/Selected/Sel+Ref/ None* box. Select a tool in the list and choose *Selected* and you see only the results of this tool in the screen image. Choosing *Sel+Ref* you get the results also from the tools used as reference for the selected one. Select *None* and no graphic results are seen in the image.



Tools using the Syringe Reference tool results

9.4.3.2 Add new tool

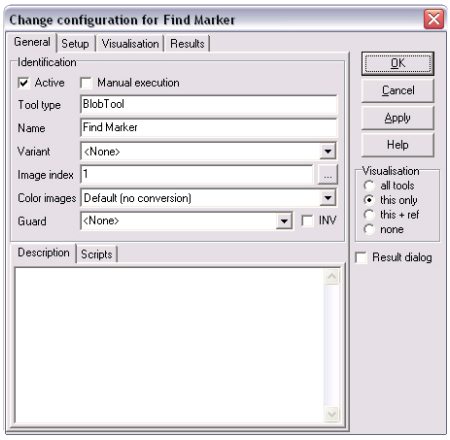


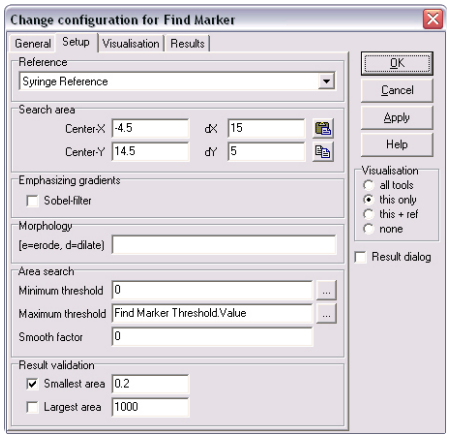
Adding a new tool

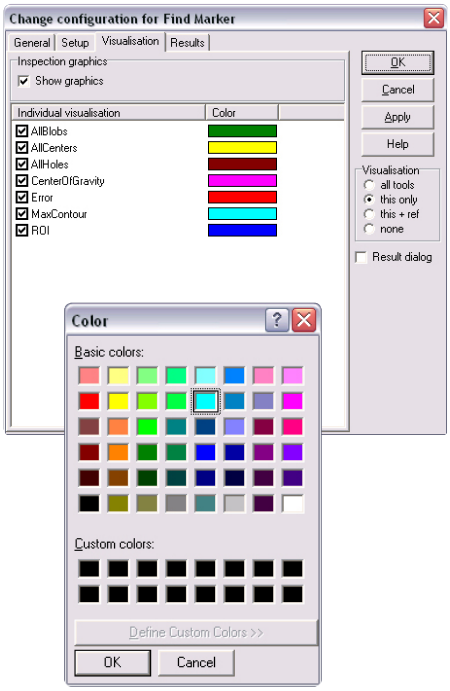
To add a tool in the toolbox, press *New* and a window like the one to the left appears. The tools are grouped in categories. Select the tool category you want to add and choose the tool from the list coming up. Give the tool a name and press *OK*. You will now see the new tool listed in the Tool Settings list.

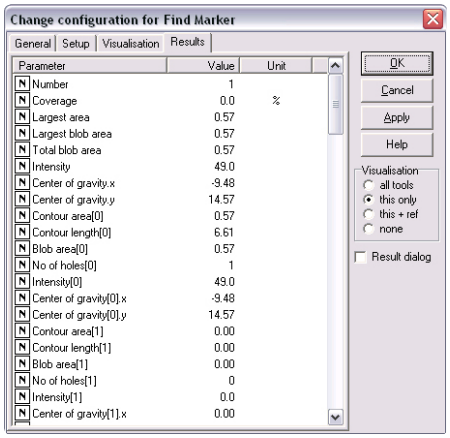
Below you see configuration options for a blob tool finding a label – 'Find Marker'. A blob is a continuous area with the same shading limited by a contour and possibly of a number of internal holes.
Double-click the tool name or select the name and choose the *Edit* button to edit a tool. Under *General* in the window showing up, you find the tool name and type in addition to eventual image index, guard and a description. You can here also activate/deactivate the tool. In the *Guard* field, you can put the name of a logical tool. The tool you are configuring is only run if the result of the logical tool is true (=1). If you select the Guard *INV* (Inverted) box, the tool is run if the result of the logical tool is false (=0).

Under *Setup* you put in relevant values to configure the tool. In this example, the orientation of the top of the product is used as a reference to find the label. This is first found by another tool – 'Syringe Reference'.









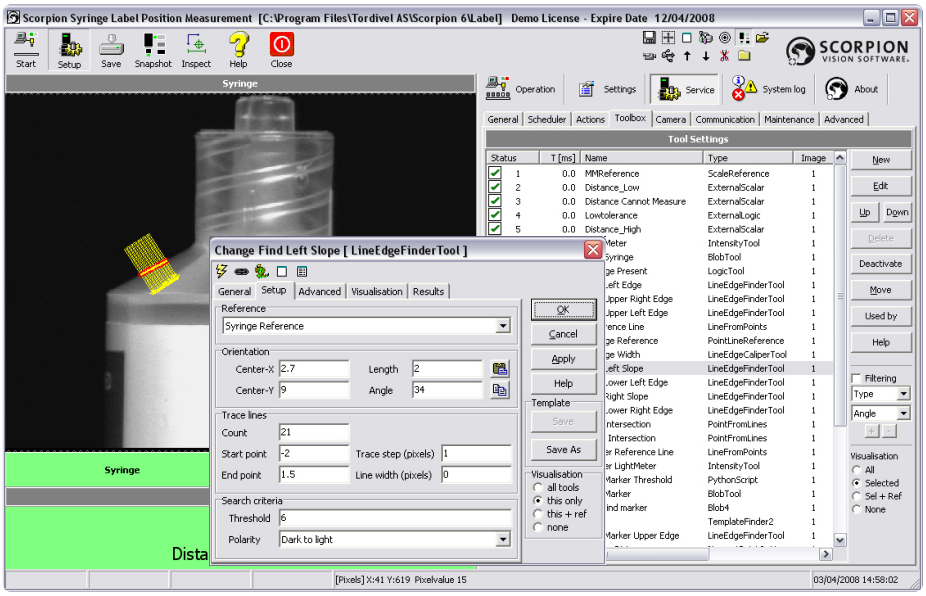
Under *Results* you find the results of a tool operation. Right click when selecting a result parameter and you can set the *precision* and add the *unit* of measurement. This will be reflected when the parameter is shown in Operation - Results and Service - Advanced - Results.

9.4.3.3 Visualise the tool results

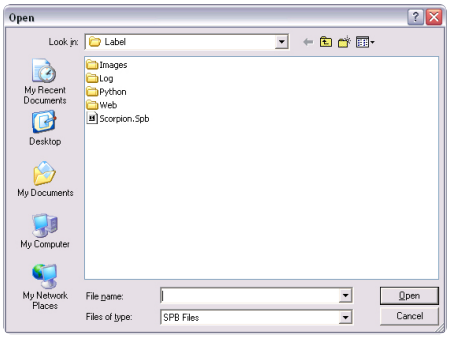
Colours are set to visualise the tool operation in the camera image on the screen. Below you see an example. The result of running the *Find Left Slope* tool is shown with yellow and red.

To help you setting up the tools, use the *Visualising - all tools/this only/this+ref* select box at the lower right of the toolbox configuration panels. Press the *Apply* button, and you immediately see the result of your settings in the camera image. The *all tools* option shows the graphical results of all tools in the image, *this only* shows only the results of the tool you are configuring. *this+ref* shows the results of the tool you are configuring in addition to the results of the tool's reference.

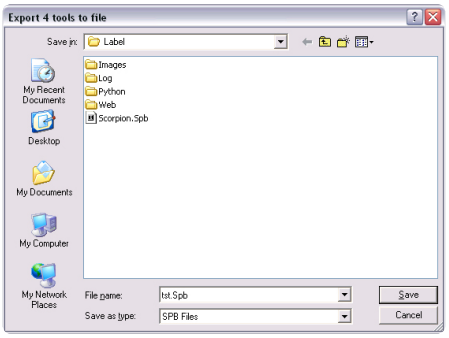
Remember that *Show image graphics when configuring* under Service - General has to be set for seeing the image graphics.



Visualisation of tool operation. Right click a tool's graphic in the image and the tool name is seen in the menu. Select it and the tool configuration window opens.



Importing tools



Exporting tools

9.4.3.4 Copy

You can copy a tool by selecting it in the list of tools, right click the mouse, choose *Copy* from the menu and *Paste* it either in another tool or as a new one. In the first case you can either paste it as an exact copy or only paste the configuration of the tool.

9.4.3.5 Import

The import method allows you to open a configuration file generated by export and select tools to import. Note the import starting position, either after first selected tool if any or at bottom of list.
All tools in the imported file will by default be selected if there are no duplicates.

- a yellow symbol will signal duplicate tools of same type
- a red symbol will indicate duplicates of different type

Note: Importing duplicates can cause broken dependencies if the imported tool is of another type or if the tool sequence is changed.

9.4.3.6 Export

From the popup menu you can export selected or all tools to an external configuration file. The export dialog allows you to save the spb-file to any location.

For more details on Scorpion tools - press the Scorpion *Help* button.

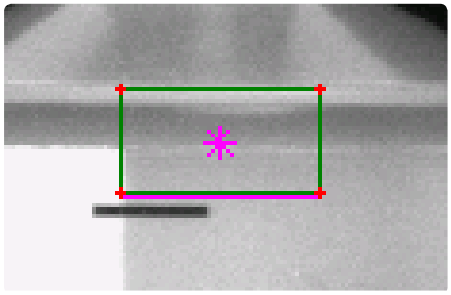
9.4.3.7 Copy and Paste ROIs

Most tools in Scorpion have the option to copy or paste the Region Of Interest (ROI) to or from the clipboard. The clipboard is visualised in the image. Several formats are accepted on the clipboard for paste operations. Below left the Search area of Blob is shown with the paste and copy ROI buttons to the right. Press the *Copy* button and the ROI is graphically shown in the image.

- *cx,cy,dx,dy* - Four numbers separated by commas: these numbers are transferred directly to the *Center-X*, *Center-Y*, *dX* and *dY* controls. The image below right shows a rectangular ROI copied to an image
  - Note: only decimal points "." are accepted, not commas ",". Applicable for regular rectangle ROIs only.
- *cx,cy* - two numbers separated by comma, set center of ROI
- *Polygons* - points stored as a Scorpion polygon. This is the format generated by e.g. clicking in the main Scorpion image.
  - Single point: If the polygon contains one point only this is used as the ROI center, where applicable.
  - Two points: Some line detection tools' ROI may also be set by two points, giving center position, length and direction. In addition, a circle may be defined by two points - center position and radius.
  - Four points: The smallest rectangle containing all the points is found and set as the ROI, or, if the tool accepts an angle, a "best fit" angled rectangle is found.
  - Free-form polygons: Polygon-ROI tools (e.g., PolygonMatch, Blob3) keep their ROI as a set of polygons of any size.

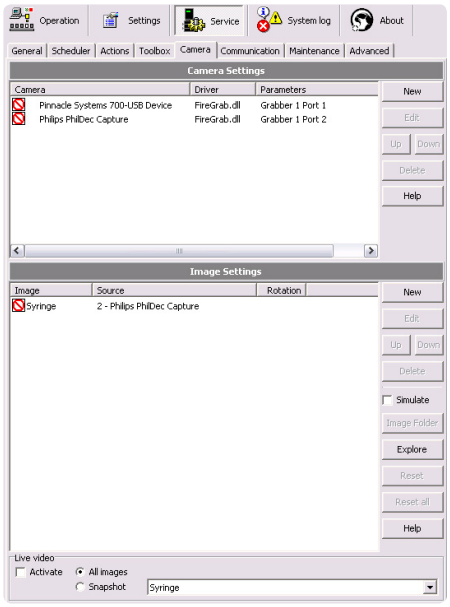
For the *Copy* operation, the polygon format is used. The number of points varies due to the kind of tool.
The operations are also available using *executeCmd*.

ROI of a Blob tool. Press the Copy button and the ROI is shown in the screen image.



Rectangle ROI copied from a tool to an image

9.5 Camera



Camera and image settings

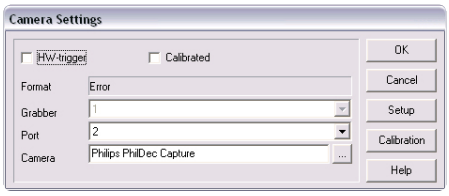
In this window you can change the camera settings. The window is split in configuration of camera and configuration of images. In *Camera Settings* you define the type of camera and the connecting board and port. Here you also define the exposure time, contrast and brightness for the image type(s).

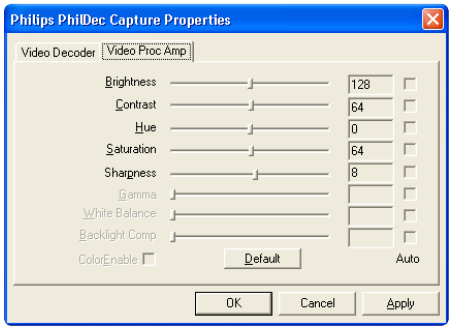
In *Image Settings* you give the image a name and define the image source – the camera or a folder if you are simulating. By choosing *Live video*, the system will take pictures at fastest possible speed. This is useful when adjusting the camera or to get an overview of image variations – if for instance the light conditions have changed. In multi image type systems, you can decide to run live video for one type of image or for all types.

Note: the dialogs described in this section assume that the DirectX driver is used: firegrab.dll.

9.5.1 Camera settings

Double-click the camera or select it and press the Edit button, and you see detailed information. This varies dependant on the camera. An example on camera setting is shown below.





Pressing the Setup button will activate the camera's own property dialogs. Press Calibration to open a calibration dialog.

9.5.2 Image settings

With Scorpion images are generated or captured in three ways:

- Captured from an image source
  - normally a camera connected to Scorpion
- Loaded from file
  - often used to simulate or test a vision system
- Generated from the inside of Scorpion
  - often as a result of processing other images
  - produced by the ImageConverter and ColorSegmentor tools

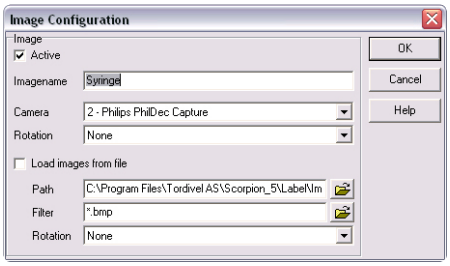


Image configuration example

Double-click the image in the Image Settings dialog or select it and press the Edit button and you see detailed information. The following properties are defined in the Image Configuration dialog coming up.

- *Active* - Used to enable the image. Images generated from the inside of Scorpion shall not be active.
- *Image* - the name of the image specified by the user. The name is displayed as a caption to the image.
- *Camera* - Used to select the camera connected to the image when not in simulation mode
- *Rotation* - defines the rotation
- *Load images from file* - The images are read from file with the given path and filter. This image path is also used as source if you mark the Simulate box in the main image settings window.
- *Path* - specifies the path to load simulation images
- *Filter* - filter to select a subset of images
- *Rotation* - defines rotation for images loaded from files

9.5.3 Installing a camera driver

Before adding a camera, the camera drivers must be installed. A number of drivers are included on the Scorpion CD-ROM. Select Drivers from the CD-ROM window to install selected camera drivers. The DirectX8.1 driver is required for camera operation under Windows 2000.

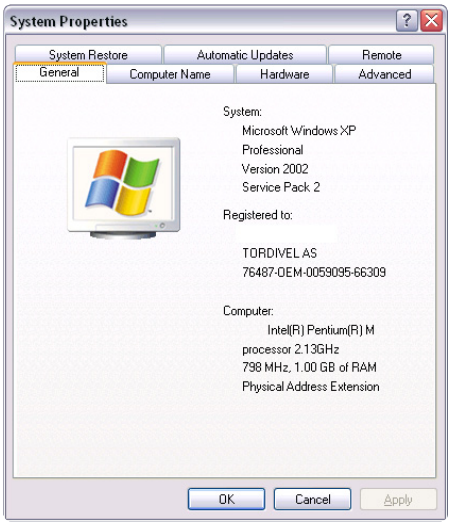
Scorpion support DirectX compatible image sources using the firegrab.dll camera interface. In this section we will outline how wdm-drivers are installed under Windows XP. The information is relevant for installing cameras under Windows 2000. DirectX camera sources are supported by Scorpion under Windows XP and Windows 2000.

Most usb and firewire camera are supported by a wdm-driver. Most likely the driver is supplied by the camera vendor. An alternative is to obtain a specific or generic driver from 3.party companies like Unibrain and Imaging Source. On the Scorpion CD there are available drivers to support cameras from Sony, Allied Vision, Point Grey, Unibrain, Basler, Imaging Source and more. It is recommended to consult the Scorpion support web or the vendor to get the latest and best wdm-driver for the camera you want to use.

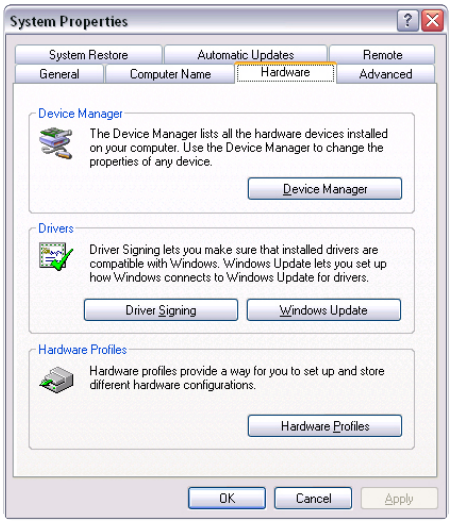
The firegrab.dll support features like hw-triggering, format-7, pausing the graph, dynamic camera commands, image averaging and dropping the first image after graph start.

Some vendors like Unibrain, Imaging Source and Allied Vision have complete installation programs removing the need for the guidelines in this section.

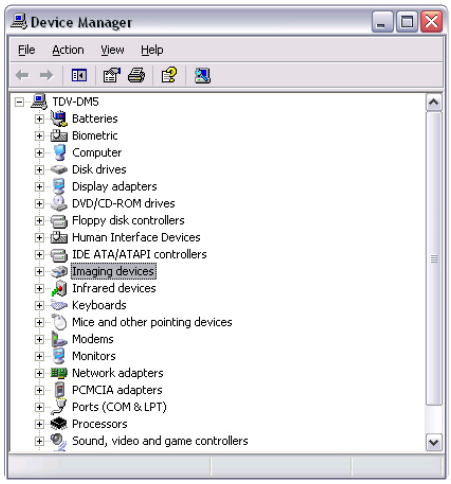
DirectX drivers are installed from the Windows Device Manager. The Device Manager is activated from My Computer - System Information.



Under the Hardware tab activate the Device Manager:



Pressing the Device Manager will activate the Device Manager. When a DirectX camera source is installed, the Imaging devices folder is present.

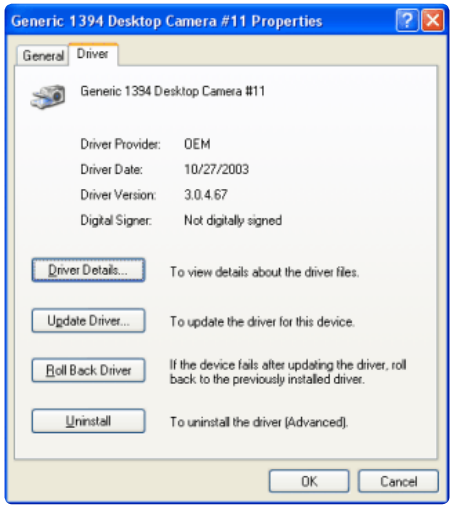


Opening the Imaging devices folder will reveal the active cameras.

Note: The driver for each camera is installed individually. This means that when a new camera is connected to the PC a new driver specific to this camera is installed. This happens even if another instance of this camera is active on the computer.

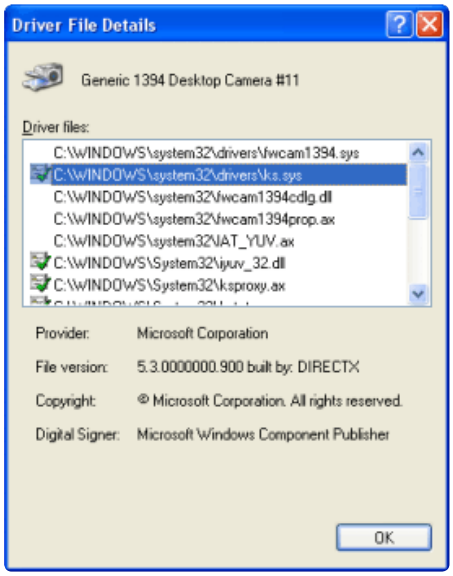
To change the camera driver select the image device and activate the camera property dialog.

Note: The default driver installed by XP is a generic driver that normally should be replaced or updated by the driver recommended by the camera vendor.

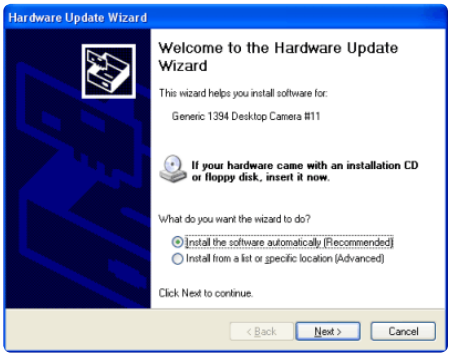


Note: If you have problems with a camera driver, it is wise to completely uninstall the driver, disconnect the camera, cold reboot, connect the camera and manually install the correct driver.

The driver details shows the all the files of the driver.



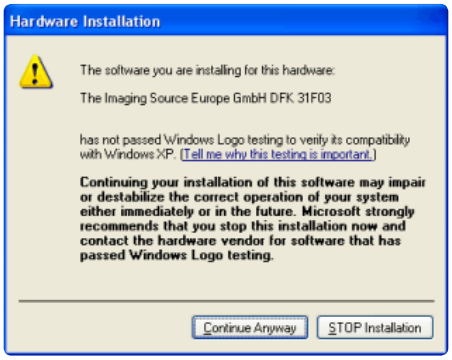
Pressing Update Driver will activate the Hardware Update Wizard.



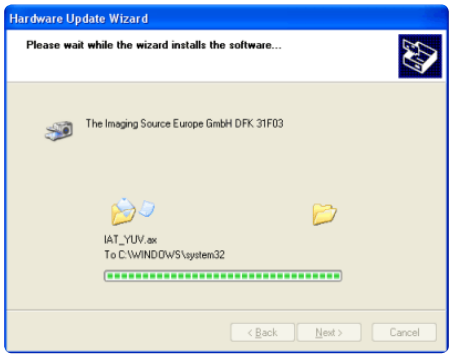
It is not recommended to let XP install a camera driver automatically.

Note: Letting XP handle installation can lead to selection of a generic driver - even if a better driver is present on the computer. XP also mix drivers when different drivers are present on the computer, thus making the image source unavailable.

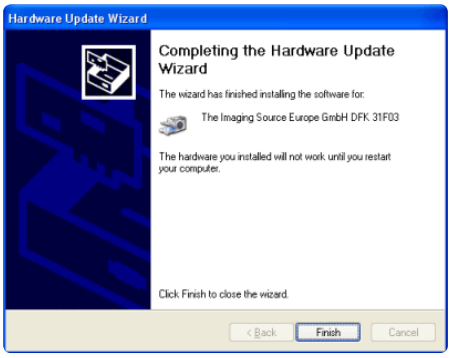
When installing a driver this warning from Microsoft is normal and can be ignored.



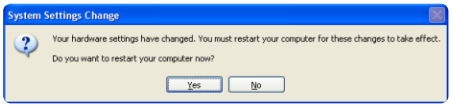
When installing the driver the following progress dialog is shown:



When completed this dialog is shown:



Pressing *Finish* may result in an XP restart requirement.



Press *Yes* to complete the driver installation.

9.5.4 Adding cameras

Any number of cameras can be connected to Scorpion. There is no software limitation.

Before adding a camera, the camera drivers must be installed. A number of drivers are included on the Scorpion CD-ROM.

> *Select Drivers from the CD-ROM window to install selected camera drivers. The DirectX8.1 driver is required for camera operation under Windows 2000.*

Adding cameras involves three steps:

Adding a camera

1. Activate Service (give the password)
2. Select the Service - Camera tab
3. Press *New* under Camera Settings
4. Select the correct camera from the list box
5. If the camera is not present in the camera list, check the Windows device manager. If the camera is present in the device manager, restart Scorpion.
6. Select *Advanced* to set the camera properties.

Adding an image and connecting it to the camera

1. Press *New* under Image settings
2. Set the image name
3. Select the correct camera from the camera list box
4. Press *OK* to close the image property dialog
5. Another image pane is now visible in Scorpion

Adding a camera trigger (a system event) – for software triggered cameras

1. Activate Service - Actions
2. Press *New* to add a system event
3. Activate the name browser
4. Select the system event - *Image 1 complete*
5. A description can be defined for the system events
6. Close the property dialog by pressing OK
7. Press *New* under Command Sequence for *Image 1 complete*
8. Set the name to Trigger Image 2
9. Set command to *Grab*. More information about Grab is found under Help
10. Set parameters to *imageno=2*

Adding a third camera is basically the same procedure.

When this procedure is done you can continue making a complete system:

- User configurable tools perform the image analysis in Scorpion. You must configure the tools in the toolbox to make Scorpion work for you.
- States are used to classify the result of an inspection. After defining the tools in the toolbox, you define the system's states. They are derived from the tool results. See chapter "States".

9.5.5 Saving Images

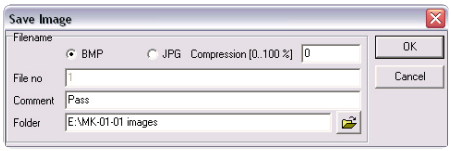
Scorpion has a number of ways to store images. The most important ways are:

Later you can use the images in simulation mode.

- ***Save current image to disk*** *Shortcut*
- ***Save*** *or* ***Save All*** *in Image History List*
- ***Save with or without overlay graphics*** *in the Image menu*

9.5.5.1 Upper right Shortcut

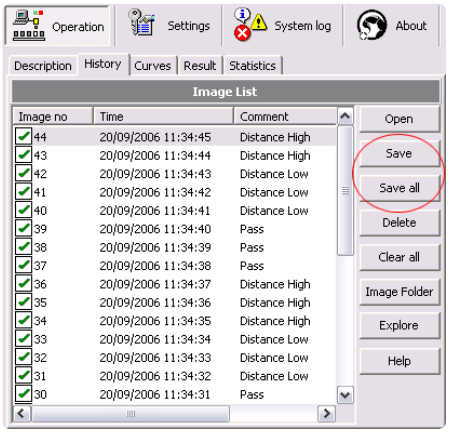
Press the *Save current image to disk* shortcut at the upper right of the screen. This activates the Save Image dialog.

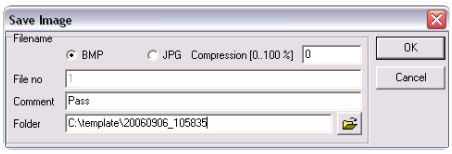


Note: You can choose between BMP and JPG format.

9.5.5.2 From the Image History List

Activate the History list menu - select the *Save* or *Save All* command



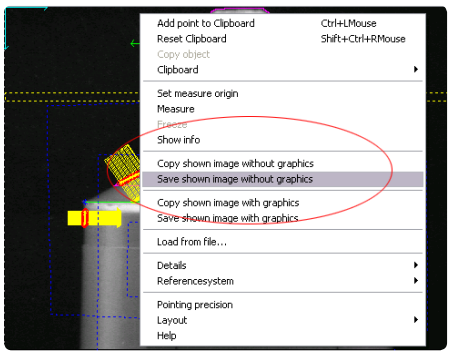


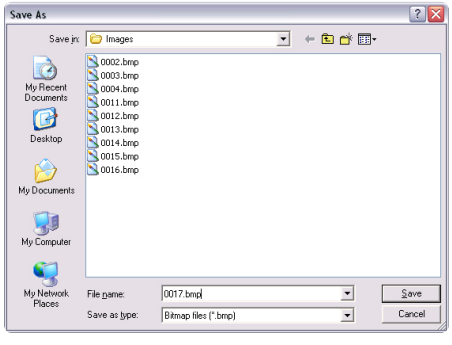
Note: All images are stored with an image sequence number prefix and an image index postfix.

9.5.5.3 From the Image Menu

To build up a set of images to use for example for test purposes, do as follows:

1. Press the right mouse button over the image pane to the left on the screen.
2. Select either *Save shown image with graphics* or *Save shown image without graphics* from the menu. Select a folder to place the images from the window coming up.





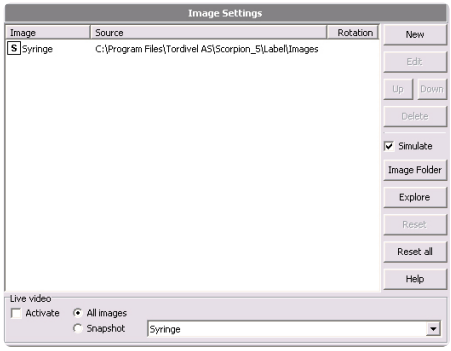
Note: Only images saved with the command *Save shown image without graphics* can be used for image processing.

9.5.6 Simulating

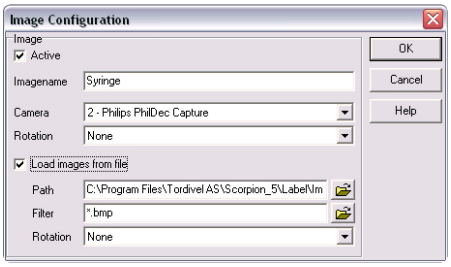
Image simulation plays a very important role in offline verification.

Activating simulation mode changes the image source from the camera to images stored on file.

1. Go to Service - Camera
2. Select *Simulate* in the Image settings panel



3. Double click or select the image in the Image settings list and press *Edit*
4. Set the image path to the folder defined above.



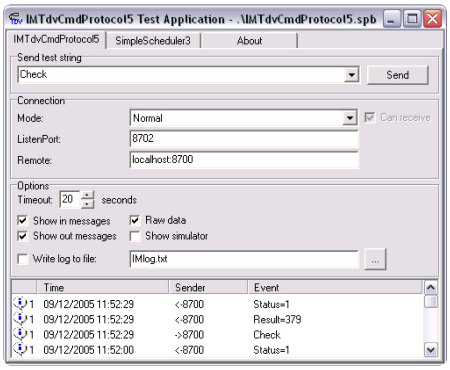
5. Press *OK* and you are ready to start Scorpion in simulation mode.
6. Open the Service – Scheduler tab and add a CameraTrigger command.
7. Click on *Start* and an image will appear in the frequency defined by the Scheduler.

9.6 Communication

9.6.1 RS232 and TCP/IP

Scorpion can send or receive commands and values to and from other applications over RS232 and/or Internet Messenger. Thus Scorpion can be configured and managed from external systems. These can also be other Scorpion systems. An external system may also send commands, set values in or receive values from Scorpion.

The dialogue to set the connection for the RS232 and Internet Messenger command protocol is shown below. The window will continuously show the commands and values transmitted between Scorpion and the external system.



External trigger system sending a check command to Scorpion and receiving the inspection result.

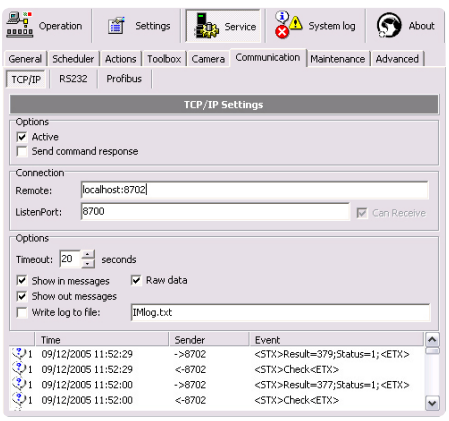
TCP/IP

Options

- Active - Enable / Disable tcp/ip communication
- Send Command Response - default off - activated for debugging purposes

Connection

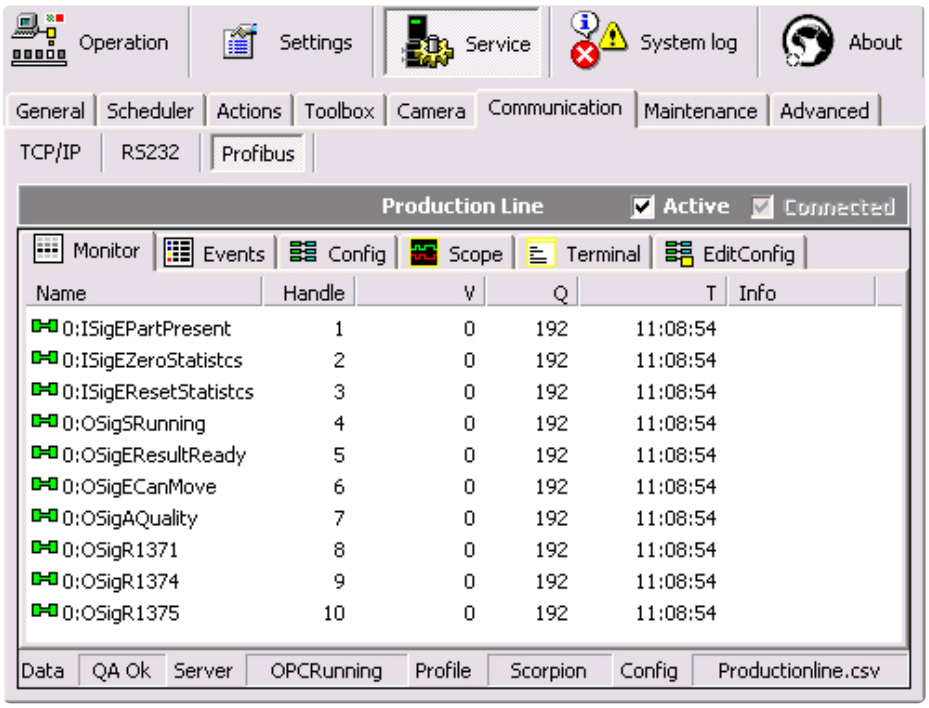
- Remote - tcp/ip and port number - example “localhost:8700’
- Listen port
- Can Receive - checked if listen port is open



Connection to an external system.

9.6.2 Profibus

The signals being sent from Scorpion are defined and described in the Actions chapter. Under Profibus you get an overview of the signals to and from the production line. You see green connection icons to the left in the example screen image below, indicating that the connection is good. You also see the name of the signals, the values and the transfer time.

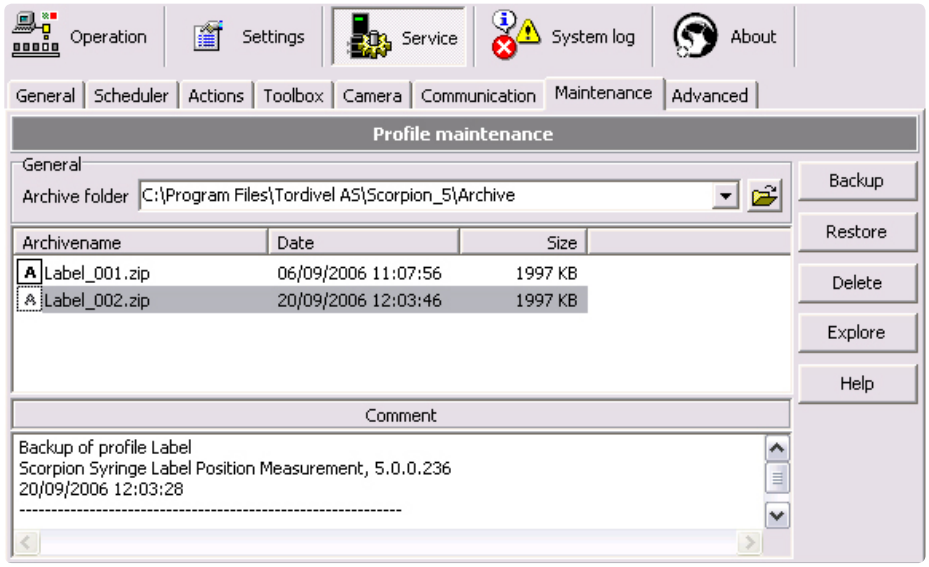


Signals to and from the production line

9.7 Maintenance

The configuration of Scorpion done to perform an inspection task is called a profile. In this window you can maintain the system profiles.

By zipping the profile folder with all its contents, profiles may be copied, moved or sent by email and restored at another computer for further management.



Profile maintenance

Choose a suited directory to place the backup under the Archive folder. Press *Explore* to get an overview of the file structure. This opens the Windows Explorer. Press *Backup* to take a backup of your system. A .zip file with the current configuration is then generated and placed in the archive folder you specified.

To restore a previous edition of the profile, select it in the list and press the *Restore* button.

9.8 Advanced

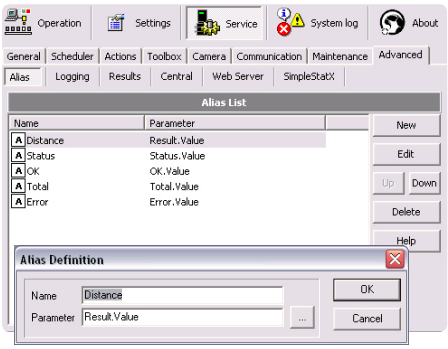
Under Advanced you find settings mainly used when installing and testing the system.

9.8.1 Alias - a new name

You can give a tool parameter an alias - a new name. This is i.e. useful for external communication. The external system thus does not have to relate to Scorpion's internal structure.

New names are defined in our example "Label on Syringe", as seen to the right.

Distance and Status are used in command sequences related to states.



List of aliases - new names for external use

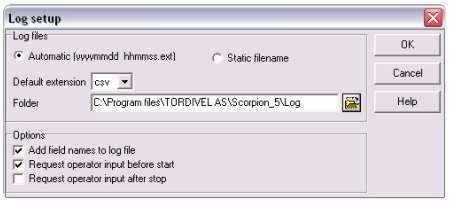
9.8.2 Logging

All measured values can be logged. Press the Setup button before activating logging. A window like the one to the left appears.

Select a folder for the log files and either an Automatic, by date and time, or Static filename.

Each time you start a new inspection batch, a new file is generated. The format of the log file is csv - comma separated values. All standard analysing products and databases like MS Access and MS Excel can then read the measured values. You can also choose log or txt as file extensions.

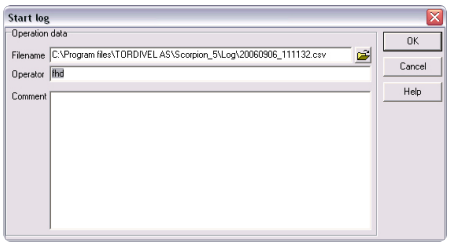
Press the *New* button to add parameters for logging. Select from the list coming up. Activate logging by selecting the *Active* check box. Each inspection cycle the parameter values will be included in the log file.



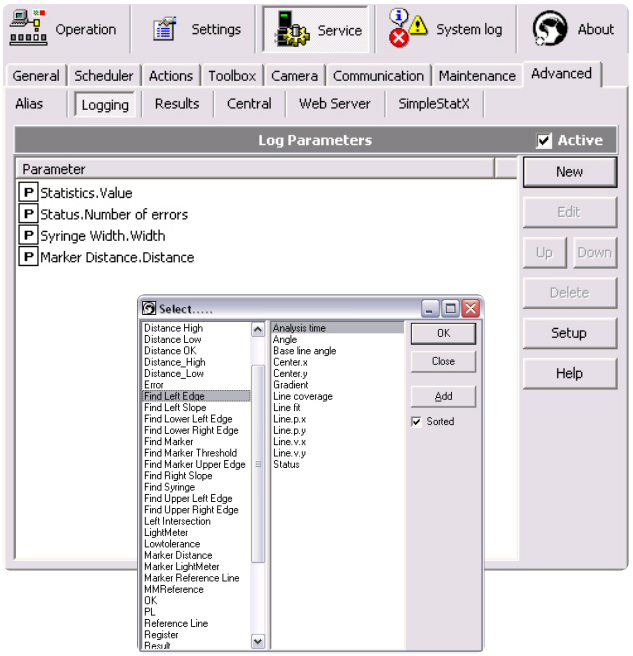
Log setup

If selecting the *Add field names to log file* option, the parameter names are included in the log file to ease readability.

The *Request operator input before start* option opens a window like the one below before inspection (and logging) is started. Here you can change the file name, name the operator and add comments. Likewise the *Request operator input after stop* option opens a window for operator input when logging ends.



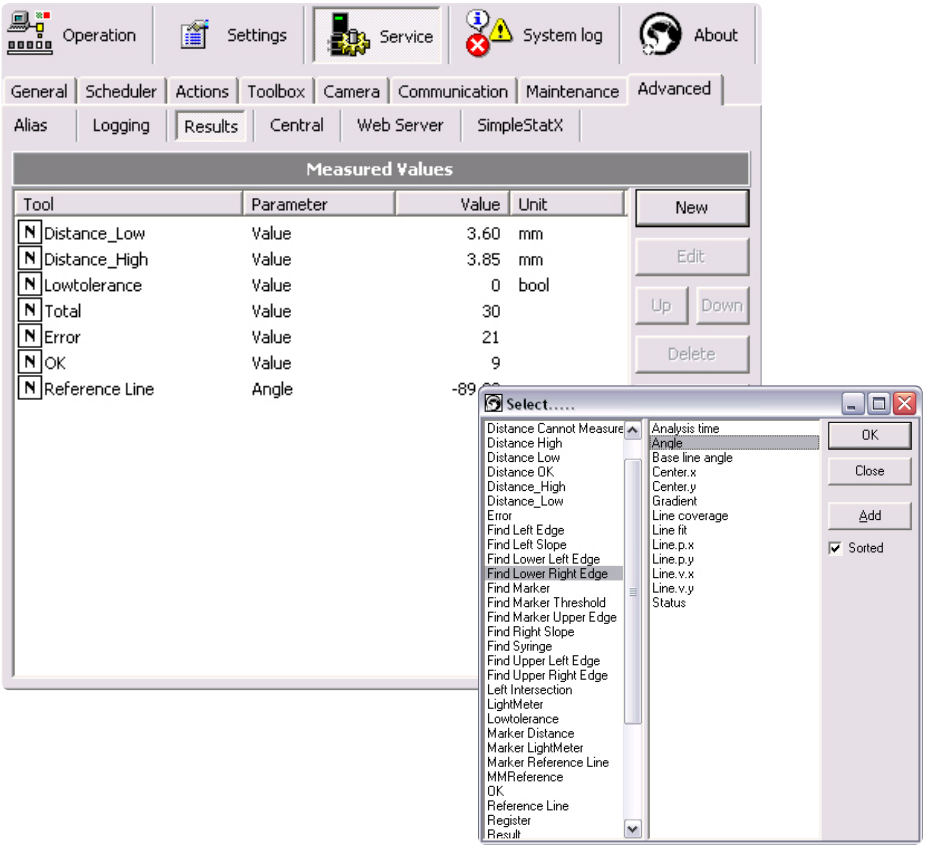
Operator input before logging starts



Adding parameters for logging

9.8.3 Results

Measured values of each inspection are shown in this panel. You can choose which parameters to display by selecting *New* and choose from the list coming up. The left column shows the image tools used in the analyses. The right column shows the values measured by the tool. Choose a value, click *Add* (then the window stays open - smart if you want to add more values) or *Ok*, and the value is included in the overview.



Inspection results

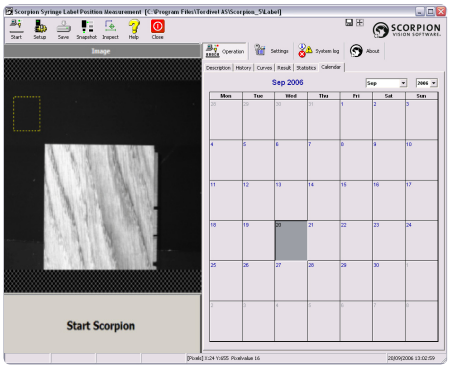
9.8.4 Central

Central provides you with flexibility. In Central plugins and python scripts are managed. Scorpion Plugins are based on ActiveX components. The Plugins are used for:

- Customising the User Interface
  - Configuration
  - Curves
  - Tables
- Adding Custom Interfaces
  - Robot
  - PLC
  - Database
  - Any network protocol

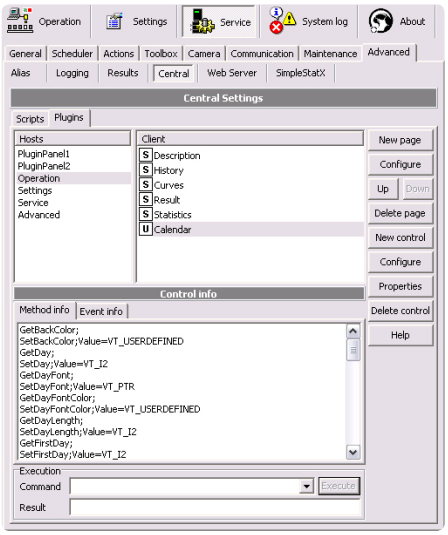
With the Python Scripts you can glue all plug-ins into the Scorpion Application to meet user demands. The plug-ins can be placed in the left hand pane or on any of the tabs in the right hand pane. Placing a plugin under Service, it will automatically be protected by the Scorpion access control.

To the left Scorpion is shown with two visible plugins. In the left pane, a button - Start Scorpion - is added to make it easier to start Scorpion. When the user pushes this button, Scorpion is started and the button changes caption to Show Stopper. The business logic for this is placed in the button onclick event. TDVButtonX is a part of TDVUtilities, a set of easy to use ActiveXs made to enhance a Scorpion application. To the right the Microsoft Calendar ActiveX control is used to set the date.



Scorpion with two visible plugins

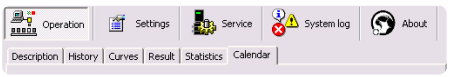
TDVCentral2 is extended using standard ActiveX components. These are managed using the Plugins panel.



The Central - Plugins panel

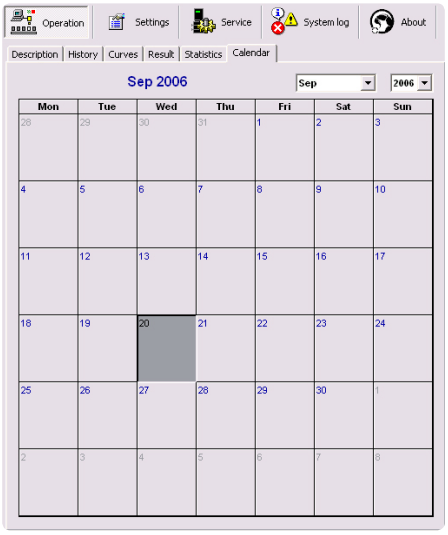
Plugin is the configuration/test panel. Use the *New page* button to insert new named pages, each containing an ActiveXContainerX control that in turn can house any other ActiveX control registered with your system.

When a page is inserted, a new main tab is created with a page for the control.



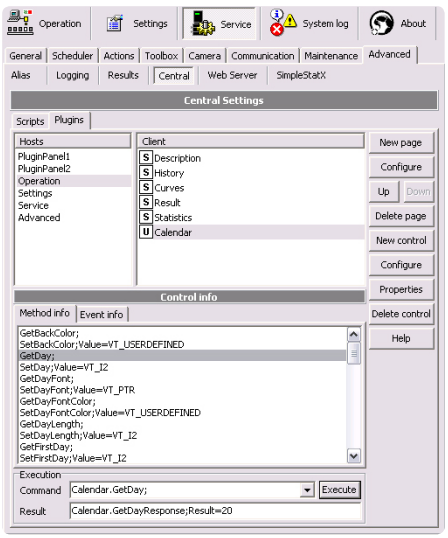
Highlight the new page name in the Pages list and click *New control* to select an ActiveX control to be inserted. The page is immediately displayed with the new control inserted.

Any method the control exposes is listed in the *Method info* list, with parameter names and types. The controls are identified by the name of the inserted page, with a period (‘.’) separating the control name from the command name, for example, “Calendar. AboutBox”. See the ActiveXContainerX help file for syntax information.



The Calendar is inserted

Note: the Python script engine is accessible via the pagename script. See the Scripts page for more information.



Method info list in the Central panel

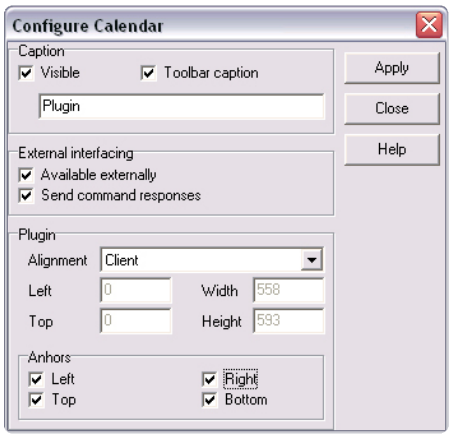
If you double-click a message in the Method list, the text is copied to the *Command* edit field. By clicking *Execute* the command is sent to the named control, and the result is shown in the *Result* field.

Note that all string handling is case insensitive: “Calendar.GetYear” and “calendar.getyear” are considered equal.

Responses set the behaviour for each inserted control - see the General page for more information.

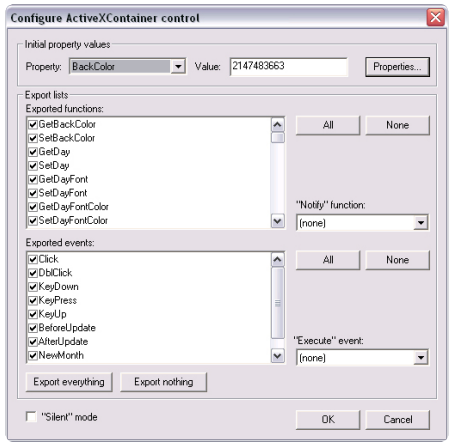
You can set the relative order of the pages with the up and down buttons.

Double click Calendar in the list of plugins to configure the control.



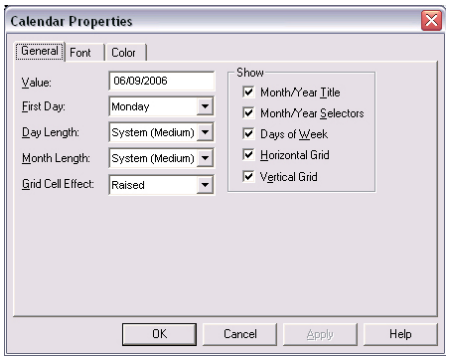
Double click Calendar in the Plugins list

Click the Configure button to open the configure ActiveXcontainer control dialog.



Calendar configuration

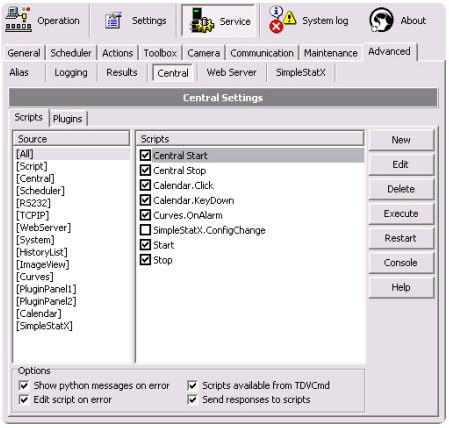
Click the Properties button to get a dialog where the calendar apperance is set.



Calendar properties

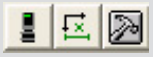
9.8.4.2 Scripts

The Scripts panel manages all Python scripts defined.



The Central - Scripts panel

In the script editor window you find these buttons:



Use them to directly take a snapshot, inspect the image or restart Central.

There are several kinds of scripts involved on this panel, listed under Source:

- [Script] – these are general named functions that you can define. See the Script details chapter below.
- [Central] – these are the Start and Stop scripts, executed on start up and shutdown, respectively. Note that these scripts are not function definitions like all the other scripts, but rather lines that are executed directly.
- (Page names) – similar, but for the configured ActiveX controls.

The [All] source lists all defined scripts.

Each script also has a checkbox associated with it - you can temporarily (or permanently) disable a script by unticking it. The scripts themselves are retained until you delete them manually, even if the associated control page is deleted.

You can execute a script at any time by pressing *Execute*. If the selected script needs parameters, you are prompted to supply them manually. See the Script details chapter below for more information.

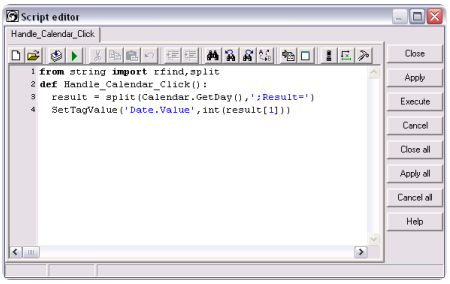
Show python messages on error - for debugging purposes, the Python log can appear automatically if an error occurs in a script. To have any effect, the user must be logged in.

Edit script on error - the Python script editor can appear automatically if an error occurs in a script, with the cursor positioned at the point of error. The user must be logged in for this to have any effect.

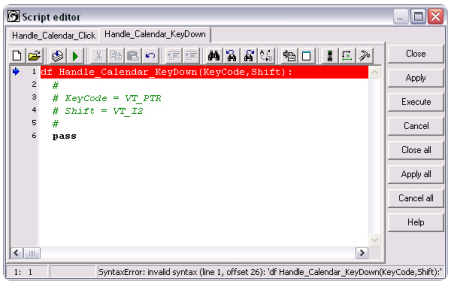
Scripts available from TDVCmd /*Send responses to scripts* set the behaviour for the general ([Script]) scripts - see the General page for more information.

Editing Scripts

Select a script in the Scripts box and press *Edit*, or double-click the script name. The script editor is shown with the current content of the script.



Each script being edited is given a tabbed page on its own in the modeless editor. Pressing Cancel will abandon the current script; pressing Apply will register the script with the Python engine, but keeping the editor open. Close will first apply, and then close the editor tab. Run will first apply, then actually run the script. If the script needs parameters, you will be prompted to give them manually. The … all buttons will handle all open scripts. In case of syntax errors, you will not be allowed to close the editor, and both the Apply and Close buttons will inform you of the error, letting you re-edit it as shown below.



Note also in this example how the scripts are auto-generated:

- The page Calendar has an event called KeyDown – the generated script is called Handle_Calendar_Keydown. This script is then executed when the event is signalled. See the “Script details” page (below) for more information on this.
- The parameters are the ones from the event, with type information given as comments. In this case the first parameter is of type VT_PTR, which is a generic pointer that is unfortunately not handled by ActiveXContainerX. Parameter two is a two-byte integer, passed by value.

We have done our best to decode the Python error log, and find that in most cases, we pinpoint the error correctly. However, there might be cases we do not handle correctly, so please refer to the Python log if in doubt.

For detailed information on scripts, see ‘Script details’ below.

Script Details

Event handlers

All events exported from the contained ActiveX controls, and most events from the built-in components, get a wrapper function defined in Python with the (Handle_) name defined in the Scripts panel. Normally you will not call these scripts yourself (they are really event handlers), but you are of course free to do so. Parameters are typed, but everything is converted to strings before the function is called. Even so, things work mostly as expected – for example, you can do maths on numeric variables without converting anything.

The built-in components also raise events and have callable functions; these are slightly different and in some cases more powerful - see the description below.

ActiveX dispatch functions

Each exported function from an ActiveX control is also wrapped in a Python function. The real name of such a function is for example Calendar_SetYear, but for convenience, an alias name of Calendar. SetYear is also made. To access any of these from a Python script, simply follow this convention:

- The page name is converted to a python class (case sensitive in Python)
- The function name is unaltered (but also case sensitive in Python)
- The function name is therefore Page. Function

All parameters are sent as a long string with Name=Value;…

For example:

```
Calendar.SetMonth("Value=6")
```

This Python call is transformed to a call to the Delphi callback Central.Execute. It should be of little importance here, but you can also call the method directly yourself. The function shall be given a single parameter, specifying a TDVCmdProtocol string. To do the same as above, you may call:

```
ExecuteCmd("Script","Calendar.
SetMonth;Value=6")
```

The return value from such a call is a TDVCmdProtocol formatted string. The examples above would both result in this string returned to your Python function:

```
Calendar.SetMonthResponse;Value=6;
Result=OK
```

This may be changed. For example, the following call

```
Calendar.GetMonth()
```

returns this

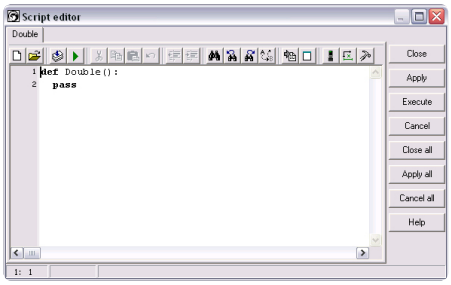
```
Calendar.GetMonthResponse;Result=6
```

General scripts

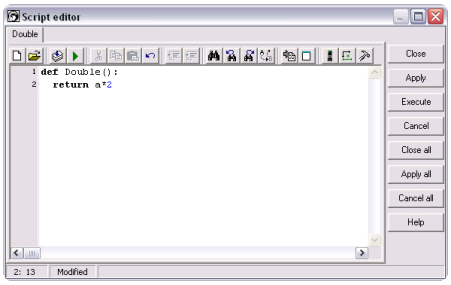
You can generate generic scripts from the Scripts page by pressing *New*.

A new script skeleton is generated.

The script does nothing by default, but is syntactically legal:



You can add parameters to this function, for example:



Remember that you have access to all the other generated scripts, including event handlers and control functions, and any global variables you may have defined, e.g., in the Start script (below). Upon Applying this script, it is also available to any other scripts you have.

Note: All scripts generated this way should just be "def" statements – that way, the model works very well. There is nothing stopping you from writing any Python code you want in these scripts, possibly apart from your own confusion.

When the *Apply* or *Close* button is pressed, the contents of the editor are simply executed by the Python engine, notifying you of any syntax (or exec) errors. When you press the *Run* button, the script source is inspected, and if the "def" line has any parameters, you will be prompted to supply them before the function is actually executed.

All the scripts defined this way are also available from TDVCmdProtocol using the special "Page" name Script. For example, you may call the script defined above by sending this string to TDVCentral2:

```
Script.Double(100)
```

Note: This is the only place the TDVCmd-Protocol commands are case sensitive. You may not call

```
Script.double(100)
```

although

```
script.Double(100)
```

is quite ok.

Note also: The string sent here does not confirm to the TDVCmdProtocol. You can not enter parameters following a semicolon.

Start and stop scripts

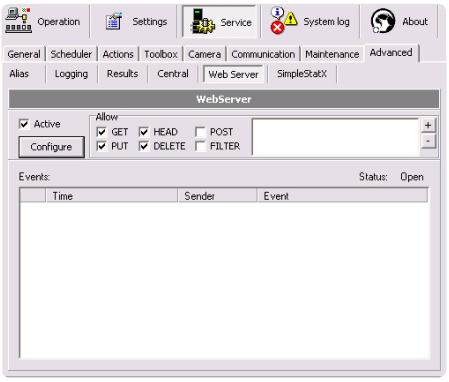
There is a script run at start up, as well as a script run just before shutdown. We like defining global variables and setting up system parameters in the start up script. When the start up script is run, all control pages have been initiated; so all functions (like Calendar.SetMonth) are available.

Note: These scripts are special. They are NOT run when the editor's *Apply* or *Close* buttons are pressed.

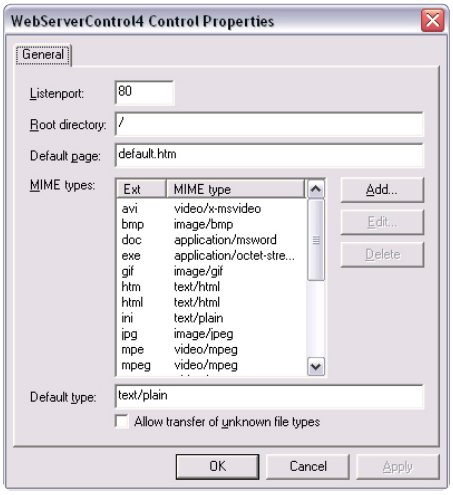
Script executed errors

If a Python error occurs when a script is run during runtime, two things may happen depending on your settings. First, the Python error log may pop up; second, an editor may pop up showing the point of error in your script(s). Again, we have done as much as we could to decode the Python messages, and if you follow the conventions with scripts, the correct script should show up in the editor with the error, even in a nested Python function call. No guarantees, though…

9.8.5 Web Server



Web Server



Configure opens the Web Server control properties.

This is a web server component that can be customised by a set of Python scripts.
The following events are available via Python event handlers:

- GET(Requester,URL,Header,Body)
- GETCompleted(Requester,URL,Code)
- PUT(Requester,URL,Header,Body)
- PUTCompleted(Requester,URL,Code)
- DELETE(Requester,URL,Header,Body)
- DELETECompleted(Requester,URL,Code)
- HEAD(Requester,URL,Header,Body)
- HEADCompleted(Requester,URL,Code)
- POST(Id,Requester,URL,Header,Body)

These event handlers are very special. All apart from the "--Completed" ones also return values to the web server, controlling the responses to requests. When you generate event handlers, you will be given comments to get you going. For details, see Scorpion's online Help. Here is a short summary of the options:

- The GET handler can accept or deny the transaction, or compose the return value itself (either as text or as a file)
- The PUT, DELETE and HEAD handlers can accept or deny the transaction
- The POST handler must be implemented for any POST calls to be accepted. The script is responsible for handling this, either synchronously or asynchronously.

The Web Server configuration is accessed here. The event handlers are edited and available under Service - Advanced - Central.

Only one function is available to the WebServer via the TDVCmdProtocol interface (details here):

```
WebServer.AsyncPostResponse;ID=<id>;ContentType=<type>;ExHeaderLines=<lines>;Body=<body>;Code=<code>
```

To call this function from a Python script, you use the abbreviated form:

```
WebServer.AsyncPostResponse(<id>,<type>,<exheaderlines>,<body>,<code>)
```

This is a related reply to a POST event, needed if the POST handler specifies asyncroneous reply.

10 System events

System event	Comment
After close service	run after service mode is left
After close settings	run after settings mode is left
After image acquisition	run when all images are complete
After inspect	run after inspection
After stop	run after stop
Before open service	run before the service mode is entered
Before open settings	run before the settings mode is entered
Before inspect	run before inspection
Before start	run before start
Close	run before a profile is closed
Error	run on inspection error, i.e. if more than one reference is accepted
Image Grab	run when the Snapshot button in the toolbar is pressed
Image N complete	run when image N is in Scorpion's memory
Image N exposed	run when an image is exposed
Inspection	run when the Inspect button in the toolbar is pressed
Start up	run after the profile is loaded: Start up
Quality alarm activate	run when a quality alarm is activated
Quality alarm deactivate	run when a quality alarm is deactivated
Unknown unit	run when the system detects an unknown unit
<User defined>	can be called from other command sequences or from an external system.

11 Commands

11.1 System Commands

The system commands perform actions on the Scorpion application.

Command	Parameters	Comment
AccessControl		Command for opening and closing the Scorpion Services and Settings. Syntax AccessControl;<service|settings>=<0|1> Command response AccessControlResponse;Result=<OK|Error> Example Open service and settings AccessControl;service=1;settings=1 Close service and settings AccessControl;service=0 The command is not protected with password. The command is suited to close Service in the system event Start. Scorpion can then not be run unprotected under normal operation.
ActiveSystem	name=profile name	Sets a specified system/profile active in a multi profile system. The system commands can be used to decide the active profile in a multi profile system. The command can e.g. be connected to io-signals to avoid changing systems with mouse and keyboard. Activating an external system to catch the operator's attention in case of faults can be another use. Syntax ActiveSystem;name=<profil> Command response ActiveSystem;<name=OK/Error;>Result=OK/Error; Example Set a named active profile: ActiveSystem;name=TI-01-01
Backup		Backups current profile configuration. The generated backup file is WinZip 8 compatible. Syntax Backup;<filename=name;><comment=text> Example Backup;comment=Automatic backup Default filename is profilename_nnn.zip, where nnn is increased by one for each backup, starting at 1. If filename contains no path infomation the backup will be stored at the configured Archive directory. Comment is appended to the default comment. The default comment contains key values for backup creation time, Scorpion version and profile name. The comment may be viewed/edited with WinZip.
Console		Controls the console window. Syntax Console;<show=0|1>;<clear=0|1>;<length=n>;<msg=text>;<save=filename> Examples Console;show=1 - shows the console Console;msg='This is a console message';show=1 Console;clear=FirstRun.Value; - clears the console if the value of parameter FirstRun.Value<>0 Console;save=console.log - saves the console messages to file console.log **Note**. When using the msg keyword, the tdvcmd separator ';' may not be used as part of the text.

Command	Parameters	Comment
Curves	cmd=zero	The Curves command resets all Scorpion graph objects. Syntax Curves;cmd=<zero> Command response CurvesResponse;<cmd=OK/Error;>Result=OK/Error; Exsample Curves;cmd=zero
Delay	value=value	The Delay command stops the program in a given number of milliseconds. Syntax Delay;value=<duration_ms> Command response DelayResponse;<value=OK|Error;>Result=OK|Error; Example Delay;value=100; - stops Scorpion for 100 ms You can also substitute scale values with aliases or result : Delay;value=ExScalar.Value Delay is only ment for short periods, <500ms, since the command **stops the program, not only the profile** (mouse, IO, screen update, etc.). Delay can be useful in some cases e.g. if an io-signal has switched on a light and you want the light to stabilise before a new image is taken. If a longer pause is needed, you should consider io-signalling used as sequence control.

LogMsg	Level=-1..3 [1];
	msg=freetext
		Writes a message in the Scorpion event log.
		
		Syntax
		LogMsg;Level=[-1|0|1|2|3];<Msg=<freetext>>
		where
		• Level - default is 1 = info1
		  • -1: error
		  • 0: warning
		  • 1: info1
		  • 2: info2
		  • 3: info3
		• Msg
		  • The message cannot include ';' or '=', they are the protocol's separation signs.
		
		Example
		LogMsg; msg='This is an Info1 message'
		LogMsg; level=-1; msg=This is an error message
		LogMsg; level=0; msg=This is a warning
		LogMsg; level=1; msg=This is an info1 message
		LogMsg; level=2; msg=This is an info2 message
		LogMsg; level=3; msg=This is an info3 message
		
		These can be useful as 'debug' messages to control command sequences and program flow.
		Guard used on the command can be useful. Put several messages in a command sequence and use Guards to control which message to write in the event log.

Profile		Profile maintenance Syntax Profile;Cmd=Save|Backup;<filename=name;><comment=text> Example Profile;Cmd=Save Profile;Cmd=Backup;comment=Backup after configuration changed **Note** Default filename is profilename_nnn.zip, where nnn is increased by one for each backup, starting at 1. If filename contains no path infomation the backup will be stored at the configured Archive directory. Comment is appended to the default comment. The default comment contains key values for backup creation time, Scorpion version and profile name. The comment may be viewed/edited with WinZip. Backup is also available by Backup-command

Command	Parameters	Comment
RefCmds	name=refname	Used to refer and run another command sequence for a given state. The command takes the name of a system state as a parameter. Syntax RefCmds;name=<tilstand> Example run the command sequence for the state 'System OK'. RefCmds;name=System OK

SaveImage	<path=filepath;>
	<filename=filename;>
	<imageno=n;>
	
	See table below for more details
		Saves the shown images in Scorpion.
		The command has two modes, fixed filename or auto saving. Fixed filename is normally used when communicating with other systems, while auto saving is meant for image logging. An increasing number gives the file name when auto saving.
		
		Syntax
		SaveImage;imageno=<1..n>;[filename=<filename>];[path=<path>]
		where
		• imageno
		  • -1 means that all images are saved
		• filename
		  • if path is not given, file name can contain full path. File name shall not be given if auto saving is set.
		• path
		  • full or relative path to image catalogue. Used if file name does not include path or path is not given (auto saving).
		
		Filename when auto saving:
		Single image configuration
		0000.bmp, 0001.bmp, 0002.bmp--> on given catalogue
		
		Multi image configuration
		[0000_0.bmp,0000_1.bmp..0000_n-1.bmp],
		[0001_0.bmp,0001_1.bmp..0001_n-1.bmp],
		[0002_0.bmp,0002_1.bmp..0002_n-1.bmp]-->
		
		Save images with fixed filename:
		SaveImage;path=c:\Images;filename=test;
		SaveImage;imageno=0;filename=c:\images\bilde1.bmp
		SaveImage;imageno=1;filename=c:\images\bilde2.bmp
		
		Automatic saving of all images, can typically be a command for a given state/reference:
		SaveImage;path=c:\images\log

Script		The Script command is used to run scripts defined in the Service-Advanced panel Script list. Syntax Script;Name=ScriptName Note that the Draw function of the PythonScript tool cannot be run in these scripts.
Shutdown		The Shutdown command saves the profiles and exits the program. Syntax Shutdown
Start		The Start command starts a profile - starts the inspection. The command can i.e. be used for automatic start managed by an io-line. Syntax Start Command response StartResponse;Result=OK/Error;
Statistics	cmd=zero cmd=reset cmd=save	The Statistics command performs actions on the Scorpion statistics. Use it to reset last period, delete all statistics or save the statistics to file. Syntax Statistics; cmd=<zero|reset|save> Examples Reset period: Statistics; cmd=zero Reset statistics: Statistics; cmd=reset Save statistics: Statistics; cmd=save (The statistics is anyhow saved at program termination.)

Command	Parameters	Comment
Stop		The Stop command stops a profile - stops the inspection. The command can i.e. be used for automatic stop managed by an io-line. The stop command can also be used to stop the profile e.g. when searching for errors. Syntax Stop Command response StopResponse;Result=OK/Error;
QAlarm	name=system/inspection/curve/command; value=0/1	Sets one of the system's quality alarms. Syntax QAlarm;name=<System|Inspection|Curve|Command> Scorpion activates quality alarms based on defined rules. QAlarmReset turns off all quality alarms.
QAlarmReset		Resets all quality alarms. Syntax QAlarmReset Scorpion activates quality alarms based on defined rules. QAlarm turns on individual quality alarms.
WebBrowser	name=Operation/Settings url=xxxx homeurl=xxxx cmd=GoBack/GoForward/Stop/Refresh/GoHome	Controls the Web Browser under operation and setup. Syntax WebBrowser;Name=<Operation|Settings>;<url=<url>>; WebBrowser;Name=<Operation|Settings>;<homeurl=<url>>; WebBrowser;Name=<Operation|Settings>; <cmd=<GoBack|GoForward|Stop|Refresh|GoHome>>; Examples WebBrowser;Name=Settings;homeurl=http://www.tordivel.com WebBrowser;Name=Settings;url=..\Etikett\web\description.htm WebBrowser;Name=Settings;cmd=GoHome Remember to activate WebBrowser under Service-General. You can use paths relative to the Scorpion working directory.

Parameters used in SaveImage	Value	Description
imageno	-1..n-1 [-1]	Image number to be saved, indexed from 0 to n-1. In multi image systems, all images will be saved if imageno is –1. The file name is then x.bmp, where x is the image number. Valid only if file name is given (not auto saving).
filename	filename [none]	If path is not given, file name can contain full path. File name shall not be given if auto saving is set.
path	path [none]	Full or relative path to image catalogue. Used if file name does not include path or path is not given (auto saving).

11.2IO Commands

The IO commands Set and GetValues access the Scorpion tagdatabase.

Command	Parameters	Comment
GetValue	name=value; <name=tagvalue;> <name=tagvalue;>	GetValue gets values from the Scorpion tagdatabase. SYNTAX GetValue;name=<value>;<value>;<value> value is an alias or tagname The GetValue result is returned in a GetValueResponse message GetValueResponse;<value>=result;<value>=result The values are the parameters given in the GetValue message EXAMPLES Request GetValue;name=ExScalaro.Value;Alias Response SetValueResponse;ExScalaro.Value=12;Alias=3
SetValue	name=value; <name=tagvalue;> <name=tagvalue;>	SetValue sets values in the Scorpion tagdatabase. SYNTAX SetValue;<name>=<value>;<name>=<value> The name is an alias or a tagname. The value is a value, an alias or a tagname A message is generated on a SetValue command - SetValueResponse. This response is normally not used. EXAMPLES Request SetValue;ExScalaro.Value=10;Alias=ExScalaro.Value; Response SetValueResponse;ExScalar.Value=OK;Result=OK

11.3 Camera Commands

The camera commands are used to capture images and set camera properties. The camera properties are only available using IEEE-1394 cameras.

Command	Parameters	Comment
Camera		Command to operate on cameras. SYNTAX Camera;CameraNo=<1,n>;cmd=<Setup>/<Reload> • Setup - Activate camera settings for selected camera • Reload - reloads camera configuration for the selected camera **Note** CameraNo is optional, default camera #1 EXAMPLE Camera;CameraNo=1;Cmd=Setup • Activates camera settings dialog for camera number 1 • Used to activate camera selection without entering service mode Camera;CameraNo=2;Cmd=Reload • reloads configuration for camera #2
Grab	See separate table below for details	Command for taking images with Scorpion. SYNTAX Grab;ImageNo=<n>; Grab;Name=<imagename> Grab;Filename=<name>;convert=<none|hsi|bw> EXAMPLES Individual image sequence where 2 images are taken, no backlight. The backlight is handled by an io-signal: Status;0:OSigSBackLight=0 Grab;imageno=1 Status;0:OSigSBackLight=1 Delay;value=100 Grab;imageno=2 Inspect 2 colour images generated by an external application, convert the first image to black and white, the other to hsi (4 images totally). The command sequence either comes complete over TdvCmd or it can be set up as a command handler in Scorpion. The external application calls this handler over TdvCmd: Grab;imageno=1;convert=bw;filename=tank1.bmp; imageno=2;convert=hsi;filename=c:\images\tank2.bmp;Inspect; Exposure, Contrast and Brightness is only supported by the camera interfaces CVLGrab.dll and CVLGrab55.dll. An eventual io to be run between each image is not run if imageno is -1 or not given. Then Scorpion will take pictures at largest possible speed.

Parameters used in Grab	Value	Description
ImageNo	1..n	Image index - if the parameter is omitted or -1, all images will be taken as fast as possible
Name	image name	Name is an alternative to imageNo - avoids dependence to index in the image list
Exposure	0..MaxInt [-1]	Exposure time in ms
Contrast	0..1 [-1]	
Brightness	0..1 [-1]	
Convert	none/bw/hsi [none]	Conversion method if the image is a colour image. HSI requires that Scorpion is setup with at least 3 images.
Filename	<path>name [blank]	If filename is given, the image is read from file instead of taken by the camera.

Command	Parameters	Comment

GetImageProp		Reads selected properties for a Firewire camera.
		• GetImageProp;
		  • <image>=<name>;<imageno>=<index>
		  • name=property; <name=property>
		
		EXAMPLE
		GetImageProp;imageNo=1;name=Shutter
		GetImageProp;image=Valve;name=Shutter
		
		Command response
		GetImagePropResponse;Shutter=3100;Result=OK
		
		POSSIBLE PROPERTIES
		Brightness, Exposure, Sharpness, Contrast, Hue, Saturation, UB, VR, WhiteBalance, Gamma, Shutter, Gain, Iris, Focus, Temperature, Trigger, Zoom, Pan, Tilt, Roll, OpticalFilter, ColorEnable, BacklightCompensation
		
		The property value is given in driver units. Scorpion gets the register value from the camera. The camera driver gives available properties. If a property is not supported, the command fails. To see the properties supported by your driver/camera, open the camera format dialog and get a list of the properties.

GetImagePropRange		Reads the value range of selected camera properties.
		• GetImagePropRange;
		  • <image>=<name>;<imageno>=<index>
		  • name=property; <name=property>
		
		EXAMPLE
		GetImagePropRange;imageNo=1;name=Shutter
		GetImagePropRange;image=Valve;name=Shutter
		
		POSSIBLE PROPERTIES
		Brightness, Exposure, Sharpness, Contrast, Hue, Saturation, UB, VR, WhiteBalance, Gamma, Shutter, Gain, Iris, Focus, Temperature, Trigger, Zoom, Pan, Tilt, Roll, OpticalFilter, ColorEnable, BacklightCompensation
		
		The property value is given in driver units. Scorpion gets the register value from the camera. The camera driver gives available properties. If a property is not supported, the command fails. Commands are supported for Firewire cameras.
		To see the properties supported by your driver/camera, open the camera format dialog and get a list of the properties.

Command	Parameters	Comment
Image		Operate on Scorpion Images Syntax Image;<imageNo=1..n>;<image=name>:<Show=<0|1|name>;<Active=0|1>;<Path=value>;<Simulate=0|1>;<Cmd=setup> Show parameter • - show image overview • 1..n - activates selected image tab number • name - activates image by tab name Active parameter Image;<Active>=<0|1>;<Image>=<name> Image;<Active>=<0|1>;<imageno>=<1..n> • Enables image by imageno or name Path parameter • sets image folder for specified image (relative to current path if not full path given) Simulate parameter • sets simulate state of given image Cmd parameter • setup - shows configuration dialog Example Image;Show=0 • Activate image overview Image;Show=1 • Activate image number 1 Image;Show=Height • Activate image named Height Image;Active=1;Image=Height • Enables Image with name Height Image;imageNo=0;path=d:\log\images • sets imagefolder for all images to d:\log\images Image;imageNo=2;cmd=setup • shows configuration dialog for image no. 2
Inspect		Inspects image(s). Syntax Inspect This command runs the image tools on the shown image(s), then runs a classification.
SetImageProp		Sets image properties in Firewire cameras. Syntax SetImageProp;<image>=<name>;<imageno>=<index>; property=value; <property=value> Example SetImageProp;Shutter=3100 Possible properties Brightness Exposure Sharpness Contrast Hue Saturation UB VR WhiteBalance Gamma Shutter Gain Iris Focus Temperature Trigger Zoom Pan Tilt Roll OpticalFilter ColorEnable BacklightCompensation The property value is given in driver units. Scorpion sets the value directly in the driver. The camera driver gives available properties. If a property is not supported, the command fails. To see the properties supported by your driver/camera, open the camera format dialog and get a list of the properties.

11.4 Communication Commands

The communication commands RS232Cmd, IMCmd and ResponseCmd are used to send TDVCmdProtocol messages over RS-232 or tcp/ip to PLCs and other host systems.

These commands combined with the RS232CmdProtocol and IMTDVCmdProtocol located on Service-Advanced-Communication provide an easy and standard way to control a Scorpion application.

Command	Parameters	Comment
IMCmd	TdvCmdProtocol format (see appendix)	Sends TdvCmdProtocol commands to external systems. Parameters to RS232Cmd and IMCmd are free text, but by the use of special formatting strings, you can replace a tag name with a value in the tag database. SYNTAX IMCmd;<command> EXAMPLES IMCmd destination=localhost:8704:Setup;Lowtolerance=%0.ofLowtolerance; IMCmd SetValue;Alias=12 With IMCmd, data is sent via the IMTDVCmdProtocol. The command is routed to a configured address : <ip>:<port>. The addressing can be overruled by prefixing the command with <destination=193.69.239.12:7002:> Parameters to IMCmd are free text, but by the use of special formatting strings, you can send commands with tagdatabase values over tcp/ip to another Scorpion application.
ResponseCmd		Sends TdvCmdProtocol commands to external systems. The parameters are equal to IMCmd and RS232Cmd, but the response is sent to origin requester, either the RS232 port or to an tcp/ip address. SYNTAX ResponseCmd;<command> EXAMPLES ResponseCmd Setup;Lowtolerance=%.1fLowtolerance sends the value of parameter back to requester ResponseCmd %sRobot.Text sends the value of Robot.Text back to requester **Note** This command is used as a a general method for sending response to any RS232 or tcp/ip requester. **Important: Execute on this command will fail - the command is sent as a response to a command**
RS232Cmd	TdvCmdProtocol format (see appendix)	Used to send data on standard TdvCmdProtocolformat over RS-232 to external systems. Parameters to RS232Cmd are free text, but by the use of special formatting strings, you can replace a tag name with a value in the tag database. SYNTAX RS232Cmd;<command> EXAMPLE RS232Cmd SetValue;Alias=12;Mode=1

11.5 Profibus Commands

The profbus commands are available on systems equipped with Siemens Profibus DP hardware.

Command	Parameters	Comment
Alarm	name=0|1|tagname; <name=<0|1|tagname>	Sets Alarm IO-signal Syntax Alarm;<PulseLen=value;>name=<0|1|tagname>; <PulseLen=value;> <name=<0|1|tagname>;> Command response AlarmResponse;<name=OK/Error;><name=OK/Error;> Result=OK/Error; Example Alarm; 0:OSigEResultReady=0 Results from the tagdatabase can be used as on/off values in Alarm. Numeric value=0 is off, numeric value<>0 is on. Multiple signals can be used in the same command separated with semi colon ';'. To pulse one or more signals, use PulseLen=nn ms. Note that PulseLen is activated only for IO-signals given after this. Alarm assumes that Scorpion has installed an OPC interface. The Status and Event commands are also used towards OPC interfaces.
Event	name=0/1/tagname; <pulselen=value>	Triggers an event (on or off). Syntax Event;<PulseLen=value;>name=0/1/tagname;<PulseLen=value;> <name=0/1/tagname;> Command response EventResponse;<PulseLen=OK/Error;><name=OK/Error;> <PulseLen=OK/Error;> <name=OK/Error;>Result=OK/Error; Example Event; 0:OSigEResultReady=0 Event;pulselen=100;0:OSigEResultReady=1 Results from the tagdatabase can be used as on/off value in Event. Numeric value=0 is off, numeric value<>0 is on. Multiple signals can be used in the same command separated with semi colon ';'. To pulse one or more signals, use PulseLen=nn ms. Note that PulseLen is activated only for IO-signals given after this. Event assumes that Scorpion has installed an OPC interface. The Status and Alarm commands are also used towards OPC interfaces.
Status	name=value; <name=value;> <name=value;>	Sets Status IO-signal Syntax Status;name=value;<name=value;><name=value;> Command response StatusResponse;name=value;<name=value;><name=value;>Result=OK|Error; Example Status; 0:OSigSProduct=Product.Value;0:OSigSResultOK=ResultOK.Value Results from the tagdatabase can be used as values in Status. The numeric value is directly set. An alias manager value can also be used to set a value. Multiple signals can be used in the same command separated with semi colon ';'. To pulse one or more signals, use PulseLen=nn ms. Note that PulseLen is activated **only** for IO-signals given after this. Status assumes that Scorpion has installed an OPC interface.

12 Terms

Term	Description
Inspection	Term for the combined operations picture(s) taking and classification of a unit.
Inspection series	Series of inspections to identify a unit. The number of inspections in a series is configurable. k of n inspections in a series must conclude similarly to qualify an identification.
Camera trigger	Digital signal that activates a new inspection series.
Profile	Scorpion is a general inspection system. A profile makes it dedicated and special for an inspection task. The configuration done to perform an inspection task makes a profile.
ROI	Region Of Interest – the image area of interest for the analysing process.
Tool	A tool is used to make a calculation. When configuring a system the tools to use are decided and their parameters are given values. The parameters set the tool limits and are typically coordinates, search areas (ROI), reference points, min/max values, logical expressions, etc. When running the tool calculates a result in the form of one or more values in addition to (in most cases) a set of graphical elements for image visualization. The result is used to define the measured objects state. This again decides if any action is to be taken. The graphical elements are used to illustrate the result in the camera image on the screen.
Icon symbol	Icon symbols are often used in detailed panels to indicate the state of for example an inspection, a tool or a system operation. Their meaning is as follows:  *Not run*  *Ok*  *Blocked by guard or reference*  *Error or No result*  *Not active*  *The license is not covering the use of this tool*  *Manual execution*

Appendix 1, TdvCmdProtocol format

In TdvCmdProtocol commands it is possible to substitute text with parameter values by specifying a formatting sequence, '%[len]type', before the parameter name.

Each formatting sequence may contain a length specifier. In case of length specifier is given the result is left padded with space or zero if the replaced parameter value contains fewer characters than the given length.

Multiple parameters MUST be separated by ';' or ','.

Format	Padding	Description
%[n]d %[n]D	'0'	Integer number format. If the parameter value is not an integer value the resulting string is the truncated integer value, there is no rounding of the value.
%[n]b %[n]B	'0'	Binary number format. The resulting string is the binary representation of the parameter value, a sequence of '0' or '1'.
%[n]h %[n]H	'0'	Hexadecimal number format. The resulting string is the hexadecimal representation of the parameter value, a sequence of '0'..'9','A'..'F'.
%[n.n]f %[n.n]F	SPACE	Floating point format. The resulting string is the floating point representation of the parameter value. If no length specifier is given, the resulting string holds up to
%[n]s %[n]S	SPACE	Text format. This is used for text parameters.
%	SPACE	Any number format, the resulting string depends of the numeric inpuit value (floating point or integer)

Examples

A.Value = 12.134
B.Value = 11.77
C.Value = 1.0
T.Text = 'OK'

Format string	Resulting string
I=%dA.Value	I=12
B=%8bA.Value	B=00001100
H=%hA.Value	H=C
H=%hA.Value	F=12.134
F=%.2fA.Value	F=12.13
F=%8.1fA.Value	F= 12.1
T=%sResult.Text	T=OK
A=%dA.Value;B=%.1fB.Value	A=12;B=11.8
X=%.3fB.Value;OK=%dC.Value	X=11.770;OK=1

Appendix 2, Test of the Scorpion communication

In the ”Label on Syringe” task, Scorpion communicates with a PLC over rs-232 and the TDVCmdProtocol. You have three alternatives when testing the communication. You can directly connect Scorpion to the PLC, establish a PLC simulator on a separate computer or you can configure the IMTDVCmdProtocol4 program as a protocol simulator. The latter communicates with Scorpion over tcp/ip, but with the same messages as the PLC transmits over rs-232. If you prefer simulating the PLC on a separate computer, use the RS232TDVCmdProtocol.

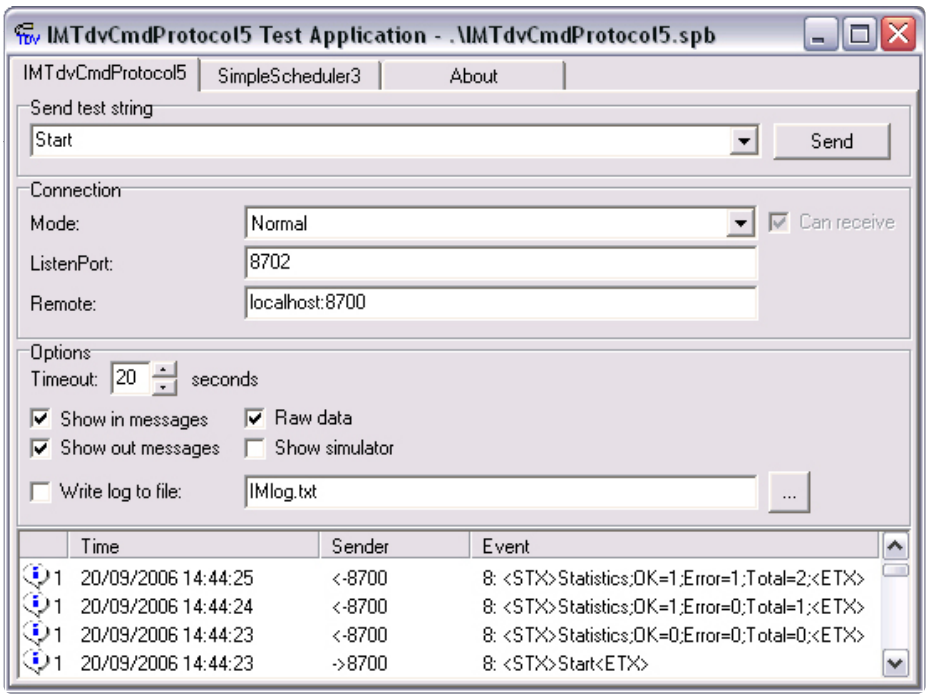
The IMTDVCmdProtocol and RS232TDVCmdProtocol are distributed together with the Scorpion software.

In this chapter we show how you can configure Scorpion and the IMTDVCmdProtocol to communicate and how to test the communication over rs-232 with the PLC. We make use of the fact that Scorpion does not saa any difference in commands coming over rs-232 and tcp/ip as long as they are in TDVCmdProtocol format.

First you establish a configuration file for the simulator. We have called it label.spb. The IMTDVCmdProtocol automatically loads this file if it is given in the program’s parameter line at start-up.

You find more information on the IMTDVCmdProtocol in the program’s own help file.

Setup of tcp/ip ports



Test program

When you have configured the communication, write Start in the Send test string field and press the Send button. Select tcp/ip under Communication and you will now see that Scorpion receives the Start string.

Establishing a command interface

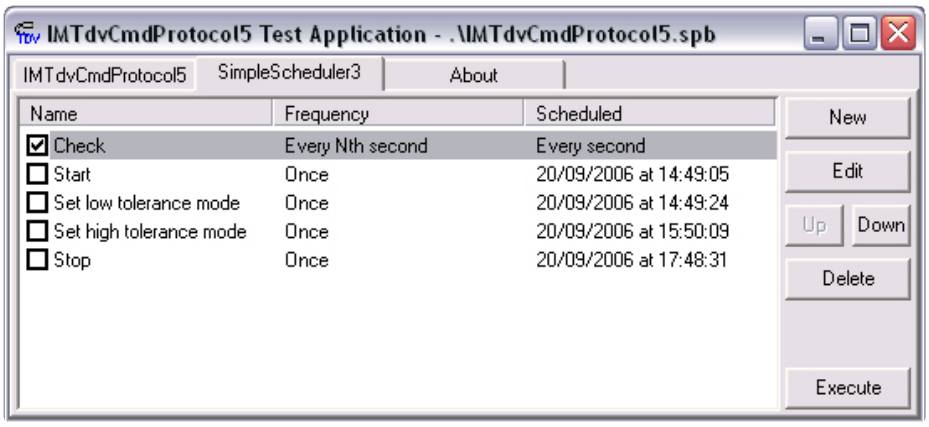
You can now define the commands from the PLC to Scorpion. They are the general commands:

- Start - Sets Scorpion in inspection mode
- Stop - Stops Scorpion

Additionally the following commands are defined under Actions in “Label on Syringe”:

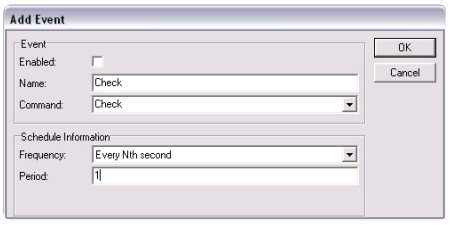
- Mode_LowTolerance - sets the mode to low tolerance
- Mode_HighTolerance - sets the mode to high tolerance

Do this under the Scheduler tab. Below you see the commands entered.



Test program - command setup

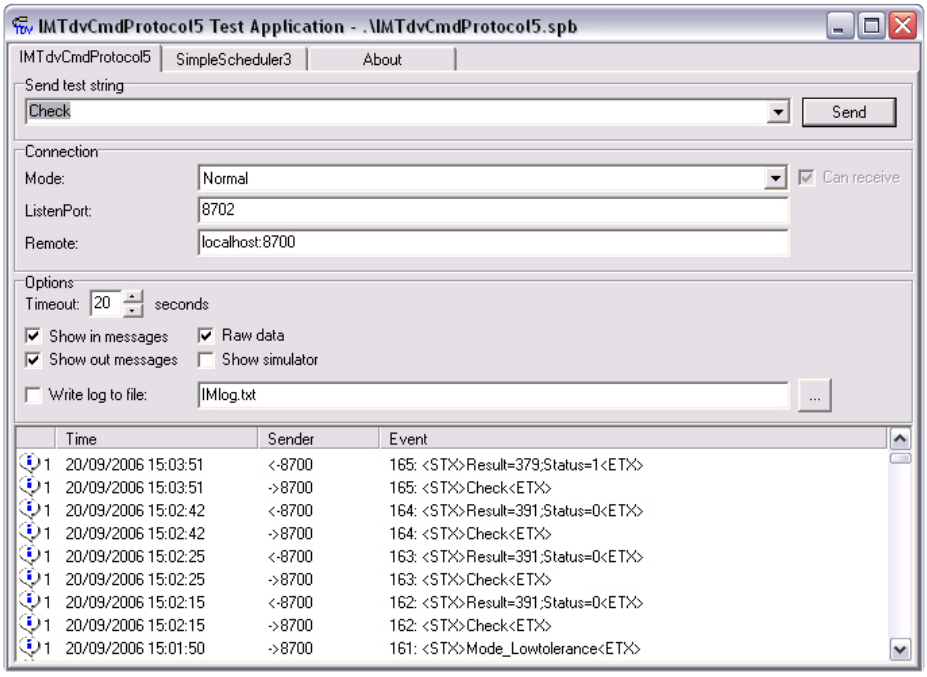
To test the interface, select a line and press Execute and you send a message to Scorpion. The Check command can be run every second when it’s activated. Below you see the setup of this command.



Setup of the Check command

Test of the command interface

When testing the command interface, have Scorpion and the IMTDVCmdProtocol running and have the following IMTDVCmdProtocol setup available on the screen:

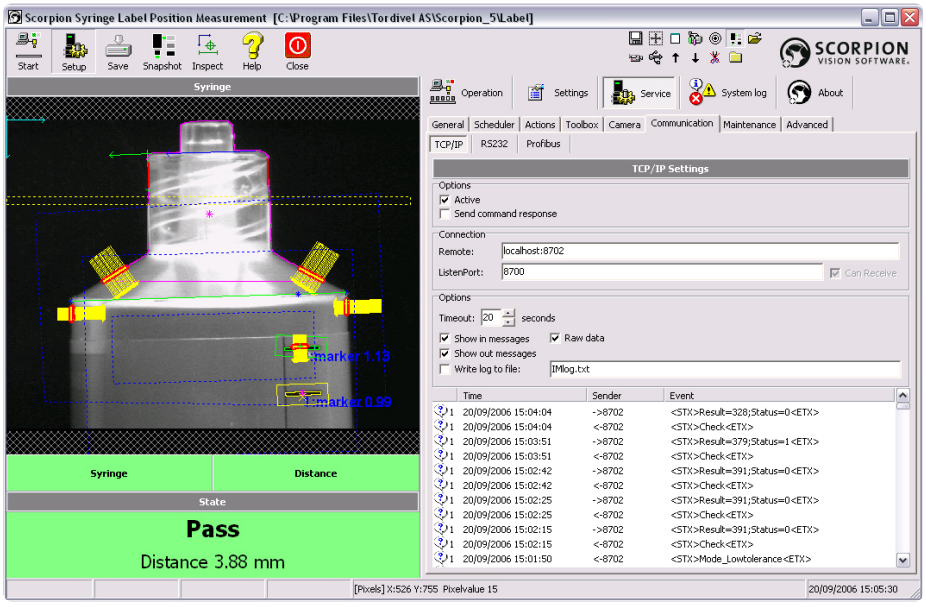


Test message log example

To start Scorpion, select the Start command in Send test string. Then select Check a number of times and see that the message is received by Scorpion and answers returned.
If you want a long lasting test, activate Check to run every second or four times a second. You can change the mode by running the Mode_LowTolerance or Mode_HighTolerance.

Document the test by copying the message list to the clipboard. Below you see an example:

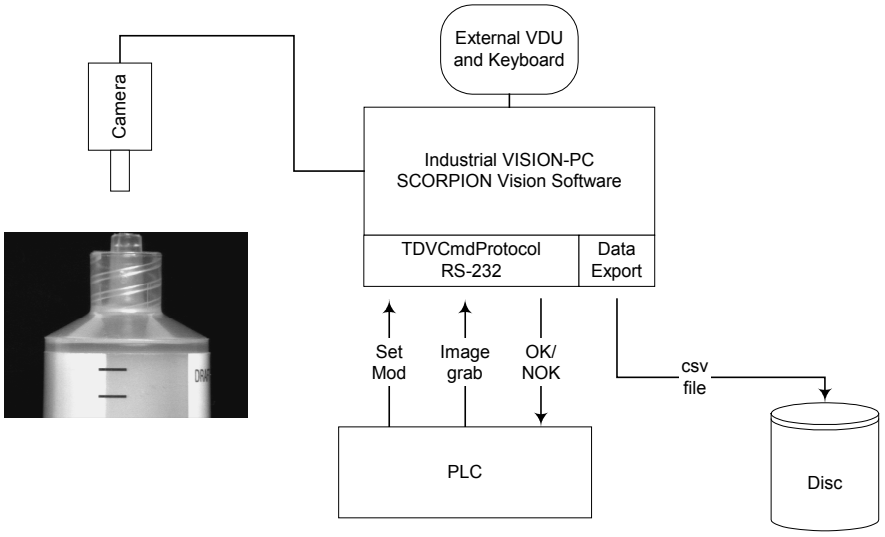
```
30.03.2002 15:36:36 ->8700 <STX>Check<ETX>
30.03.2002 15:36:37 <-8700 <STX>result=381;status=11<ETX>
30.03.2002 15:36:38 ->8700 <STX>Check<ETX>
30.03.2002 15:36:38 <-8700 <STX>result=377;status=11<ETX>
30.03.2002 15:36:39 ->8700 <STX>Check<ETX>
30.03.2002 15:36:39 <-8700 <STX>result=382;status=11<ETX>
30.03.2002 15:39:08 ->8700 <STX>Mode_LowTolerance<ETX>
30.03.2002 15:39:11 ->8700 <STX>Stop<ETX>
30.03.2002 15:39:14 ->8700 <STX>Start<ETX>
30.03.2002 15:39:17 ->8700 <STX>Check<ETX>
30.03.2002 15:39:18 <-8700 <STX>result=370;status=11<ETX>
```



TCP/IP setup. You see the Scorpion setup and how the messages are filling the command log. You can see that mode is set to low tolerance and that Scorpion communicates on port 8700.

Appendix 3, Block diagram - Label on syringe

Below you see a block diagram of our example ”Label on Syringe”. This is a typical setup of a Scorpion based system. One camera is connected to Scorpion.



The production process communication is with a PLC over rs-232. Scorpion can flexible be configured and adjusted to the production process requirements by using the TDVCmdprotocol text protocol. Scorpion is a slave of the PLC, and executes only tasks on command.

The following command protocol is defined in ”Label on Syringe”:

1. Check - sent from the PLC to Scorpion
   a. <stx>Check<etx>
2. Inspection result – sent as response to the Check command
   a. <stx>result=<xxx>; status=<xx><etx>
      - result=<xxx>
        - distance is given in 1/100 mm
      - status=<xx>
        - 11 - Ok
        - 10 - Rejected
        - 00 – Cannot measure
3. Set Mode - PLS selects operating mode
   a. <stx>Mode_LowTolerance<etx>
   b. <stx>Mode_HighTolerance<etx>

<stx> and <etx> are ascii control signs and define start and stop of a command packet in the TdvCmdProtocol.

With this text protocol defined, the PLC can do the following tasks:

- Set operating mode
  - There are two operating modes defined, one with high and one with low tolerance values
- Check syringe
  - Asks Scorpion to take an image and do an image analyses with the selected operating mode
  - Scorpion returns the inspection result with
    - Distance given in 1/100 mm
    - Status telling if values are found and within given limits

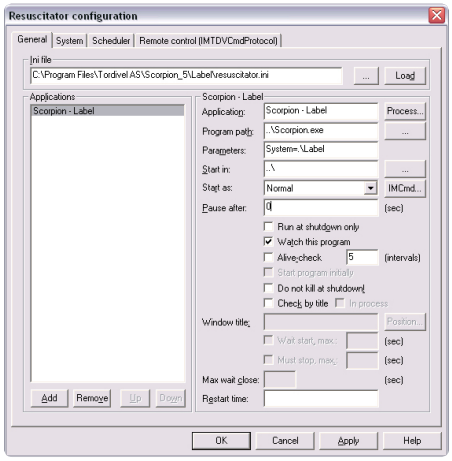
Appendix 4, Scorpion Watchdog

Resuscitator for NT, developed by Tordivel AS, can be used to start and supervise Scorpion. Resuscitator guarantees that your services are available 24 hours a day.

Resuscitator is a highly configurable and unique tool designed to do program management on standalone Windows 2000/NT computers. Resuscitator's primary tasks are to ensure program availability, force strict and controlled startup and shutdown sequences, and offer flexible program scheduling and remote control. You can restart your computer over the Internet using the Remote Client program.

Connecting Scorpion Watchdog and Scorpion Vision Software®

Resuscitator for NT – the Scorpion Watchdog – can make many programs behave like one single system started with a shortcut. Below you see an example based on test of the communication in "Label on Syringe".



The configuration dialog for setting up the Scorpion - Label system.

On the "Label" profile you find two configuration files:

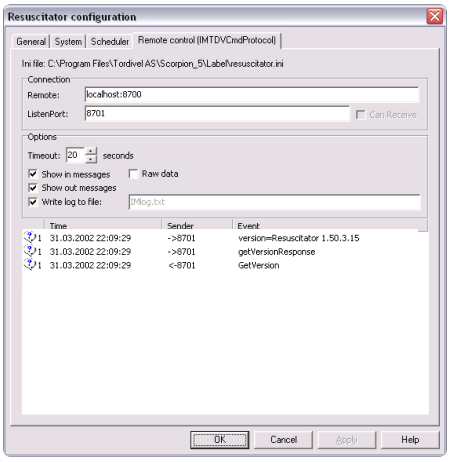
- Label.spb - Setup of IMTDVCmdProtocol4 for test of Label
- Resuscitator.ini - Setup of Resuscitator for NT that starts
  - Scorpion - Label
  - IMTDVCmdProtocol Label.spb

You also find a shortcut - Resuscitator - Test of Label on Syringe. When starting this shortcut, Resuscitator for NT will start in the background. Use the FindResuscitator program to make Resuscitator visible to inspect the setup of resuscitator.ini. Resuscitator starts the IMTDVCmdProtocol and Scorpion, thus you can test the profile's external interface.
Test of this interface is thoroughly described in the Test of Scorpion communication chapter.

Setup of Resuscitator for NT is easy. You find all the details in Resuscitator's own user manual.

Observe that the program path is relatively given: .\Scorpion.exe and that the command line parameters are entered as normal: System=.\Label.

To terminate Resuscitator, IMTDVCmdProtocol and Scorpion with the press of a button, you must configure Resuscitator to receive commands in TDVCmdProtocol format over tcp/ip and configure a new system event in Scorpion. Terminate. This sends a request to Resuscitator to terminate. Resuscitator will then stop all programs under supervision and finally terminate itself.



Setup of remote interface - tcp/ip

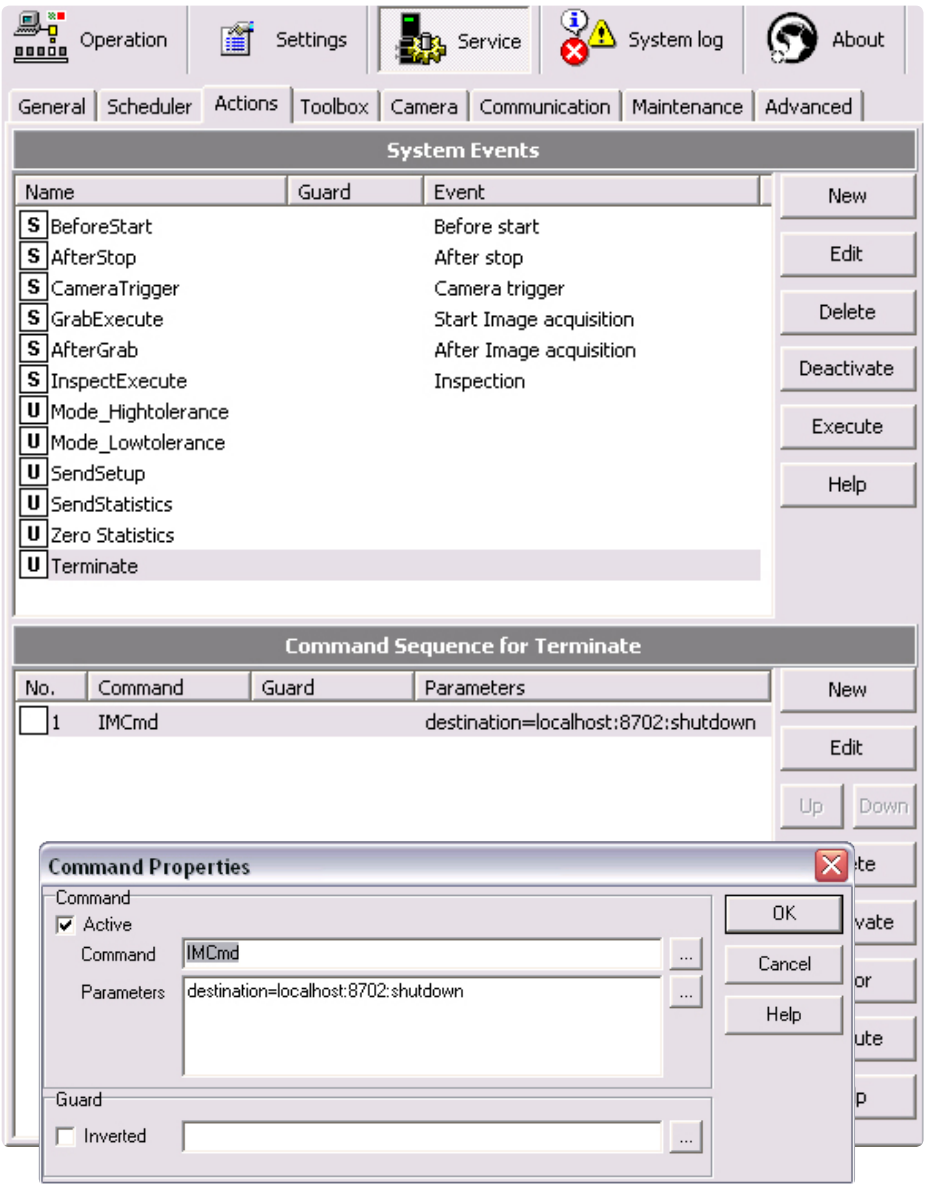
To prepare Resuscitator for remote control, do as follows:
Under the System tab, activate the Enable Remote Control in the IMTDVCmdProtocol group box. Find a free port number for the Resuscitator to receive commands.

In the example Resuscitator is set up to send to localhost:8700, the Scorpion - Label listen port. Resuscitator itself listens on port 8702 (Can Receive is marked). In the message log you see that IMTDVCmdProtocol has sent the GetVersion command to Resuscitator and has received a response.

Check that the communication over tcp/ip is activated in Scorpion and set up a listen port - 8700. Additionally define a new system event Terminate under Service - Actions. Terminate is run when the user presses the button, but before Scorpion terminates. Terminate sends a Shutdown command to Resuscitator:

```
IMCmd;destination=localhost:8702:shutdown
```

With the 'destination=localhost:8702' prefix, the command is sent to Resuscitator independent of the receiver setup in Communication - tcp/ip.



Command to Resuscitator when Terminate

Scorpion Vision Software® from Tordivel AS is an independent and open software tool for industrial vision. It is the best choice for the production engineer wanting to save cost, automate or secure quality. The system gives the user the choice of a small form factor with the Sony SmartCam or the power of a standard PC system.

Scorpion Vision Software is used in a vast variety of industries; automotive, wood, furniture manufacturing, food, pharmaceutical, robotics, packaging, energy and more. It solves tasks within robot vision, label and surface inspection, assembly verification, quality control, color identification and high precision gauging.

The system is founded on top of a standard Windows PC platform - thus a flexible alternative to proprietary vision sensors or custom vision systems. It is cost effective benefitting from the processing power of the Intel processors, low cost and high quality firewire cameras and the possibility of connecting multiple cameras to one PC.

Scorpion Vision Software is packed with features and details making it easy to develop and maintain robust industrial vision systems. Every single feature is specified, implemented, tested and verified based on experience obtained on the factory floor.

Scorpion Vision Software offers large reduction in the development time and maintenance cost for machine vision systems.

TORDIVEL