



OmniView Application Users Guide

8/19/2011
Version 6.0

The software described in this document is furnished under license, and may be used or copied only in accordance with the terms of such license and with the inclusion of the copyright notice shown on this page. Neither the software, this document, nor any copies thereof may be provided to, or otherwise made available to, anyone other than the licensee. Title to, and ownership of, this software remains with Cognex Corporation or its licensor. Cognex Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Cognex Corporation. Cognex Corporation makes no warranties, either express or implied, regarding the described software, its merchantability, non-infringement or its fitness for any particular purpose.

The information in this document is subject to change without notice and should not be construed as a commitment by Cognex Corporation. Cognex Corporation is not responsible for any errors that may be present in either this document or the associated software.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, nor transferred to any other media or language without the written permission of Cognex Corporation.

Copyright © 2011 Cognex Corporation. All Rights Reserved.

Portions of the hardware and software provided by Cognex may be covered by one or more of the U.S. and foreign patents listed below as well as pending U.S. and foreign patents. Such pending U.S. and foreign patents issued after the date of this document are listed on Cognex web site at <http://www.cognex.com/patents>.

VisionPro

5481712, 5495537, 5548326, 5583954, 5602937, 5640200, 5751853, 5768443, 5825913, 5850466, 5872870, 5901241, 5943441, 5978080, 5978521, 5987172, 6005978, 6039254, 6064388, 6075881, 6137893, 6141033, 6167150, 6215915, 6240208, 6324299, 6381366, 6381375, 6411734, 6421458, 6459820, 6490375, 6516092, 6563324, 6658145, 6687402, 6690842, 6697535, 6718074, 6748110, 6771808, 6804416, 6836567, 6850646, 6856698, 6920241, 6959112, 6963338, 6973207, 6975764, 6985625, 6993177, 6993192, 7006712, 7016539, 7043081, 7058225, 7065262, 7088862, 7164796, 7190834, 7242801, 7251366, 7313761, EP0713593, JP3522280, JP3927239

DataMan

5742037, 5943441, 6215915, 6236769, 6282328, 6381375, 6408109, 6457032, 6690842, 6941026, 7175090, 7181066, 7412106, 7427028, 7549582, 7604174, 7614563, 7617984, US-2005-0087601-A1, US-2006-0131418-A1, US-2006-0131419-A1, US-2006-0133757-A1, US-2007-0090193-A1, US-2007-0091332-A1, US-2007-0152064-A1, US-2007-0170259-A1, US-2008-0004822-A1, US-2008-0011855-A1, US-2008-0142604-A1, US-2008-0143838-A1, US-2008-0158365-A1, US-2009-0090781-A1, US-2009-0108073, US-2009-0121027-A1, US-2009-0166424-A1, US-2009-0294541-A1, WO06065619A1, EP1687752

CVL

5495537, 5548326, 5583954, 5602937, 5640200, 5717785, 5751853, 5768443, 5825483, 5825913, 5850466, 5859923, 5872870, 5901241, 5943441, 5949905, 5978080, 5987172, 5995648, 6002793, 6005978, 6064388, 6067379, 6075881, 6137893, 6141033, 6157732, 6167150, 6215915, 6240208, 6240218, 6324299, 6381366, 6381375, 6408109, 6411734, 6421458, 6457032, 6459820, 6490375, 6516092, 6563324, 6658145, 6687402, 6690842, 6718074, 6748110, 6751361, 6771808, 6798925, 6804416, 6836567, 6850646, 6856698, 6920241, 6959112, 6975764, 6985625, 6993177, 6993192, 7006712, 7016539, 7043081, 7058225, 7065262, 7088862, 7164796, 7190834, 7242801, 7251366, EP0713593, JP3522280, JP3927239

VGR

5495537, 5602937, 5640200, 5768443, 5825483, 5850466, 5859923, 5949905, 5978080, 5995648, 6002793, 6005978, 6075881, 6137893, 6141033, 6157732, 6167150, 6215915, 6240208, 6748110, 6751361, 6771808, 6804416, 6836567, 6850646, 6856698, 6959112, 6975764, 6985625, 6993192, 7006712, 7016539, 7043081, 7058225, 7065262, 7088862, 7164796, 7190834, 7242801, 7251366

OMNIVIEW

6215915, 6381375, 6408109, 6421458, 6457032, 6459820, 6594623, 6804416, 6959112, 7383536

CVL Vision Library

5495537, 5548326, 5583954, 5602937, 5640200, 5717785, 5751853, 5768443, 5825483, 5825913, 5850466, 5859923, 5872870, 5901241, 5943441, 5949905, 5978080, 5987172, 5995648, 6002793, 6005978, 6064388, 6067379, 6075881, 6137893, 6141033, 6157732, 6167150, 6215915, 6240208, 6240218, 6324299, 6381366, 6381375, 6408109, 6411734, 6421458, 6457032, 6459820, 6490375, 6516092, 6563324, 6658145, 6687402, 6690842, 6718074, 6748110, 6751361, 6771808, 6798925, 6804416, 6836567, 6850646, 6856698, 6920241, 6959112, 6975764, 6985625, 6993177, 6993192, 7006712, 7016539, 7043081, 7058225, 7065262, 7088862, 7164796, 7190834, 7242801, 7251366, EP0713593, JP3522280, JP3927239

SMD 4

5995648, 5850466, 6751361, 6690842, 6563324, 6490375, 5949905, 5978080, 6137893, 6167150, 6075881, 6748110, 5859923, 6411734, 6324299, 6516092, 7190834, 6658145, 6836567, 6850646, 6975764, 6985625, 6993192, 7006712, 7043081, 7058225, 7065262, 7088862, 7164796, 7251366, 6856698, 6002793, 6005978, 6771808, 6804416, 7016539, 6959112, 5602937, 7242801, 5640200, 5495537, 5768443, 5825483, 6421458, 6459820,

6215915, 6381375, 6457032, 6157732, 6408109, 6141033, 6026176, 6442291, 6151406, 6396942, 6614926, 5371690, 5845007, 5943441, 6963338, 5805722, 5909504, 5933523, 5964844, 5974169, 5987172, 6078700, 6252986, 6278796, 6307210, 6408429, 6424734, 6526165, 6571006, 6639624, 6681039, 6748104, 6813377, 6853751, 6898333, 6950548, 6993177, 7139421, 5757956

BGA II and BGA III

5495537, 5602937, 5640200, 5768443, 5801966, 5825483, 5850466, 5859923, 5949905, 5978080, 5995648, 6002793, 6005978, 6026176, 6055328, 6075881, 6115042, 6118893, 6130959, 6137893, 6141009, 6141033, 6151406, 6157732, 6167150, 6215915, 6289117, 6324299, 6353676, 6381375, 6396942, 6408109, 6411734, 6421458, 6442291, 6457032, 6459820, 6490375, 6516092, 6563324, 6577775, 6614926, 6658145, 6690842, 6748110, 6751361, 6771808, 6804416, 6836567, 6850646, 6856698, 6959112, 6975764, 6985625, 6993192, 7006712, 7016539, 7043081, 7058225, 7065262, 7088862, 7164796, 7190834, 7242801, 7251366

Wire Bonder

5495537, 5532739, 5581632, 5602937, 5640199, 5640200, 5642158, 5676302, 5754679, 5757956, 5768443, 5825483, 5835622, 5850466, 5859923, 5861909, 5949905, 5978080, 5991436, 5995648, 6002793, 6005978, 6035066, 6061467, 6075881, 6137893, 6141033, 6157732, 6167150, 6215915, 6289492, 6324299, 6381375, 6408109, 6411734, 6421458, 6457032, 6459820, 6490375, 6516092, 6563324, 6658145, 6690842, 6748110, 6751361, 6771808, 6804416, 6836567, 6850646, 6856698, 6959112, 6975764, 6985625, 6993192, 7006712, 7016539, 7043081, 7058225, 7065262, 7088862, 7164796, 7171036, 7190834, 7242801, 7251366

The following are registered trademarks of Cognex Corporation:

acuReader® BGAI® Check it with Checker® Checker® Cognex Vision for Industry CVC-1000® CVL® DataMan® DisplayInspect® DVT® EasyBuilder® IDMax® In-SightIn-Sight 2000® In-Sight® (insignia with cross-hairs) MVS-8000® OmniView® PatFind® PatFlex® PatInspect® PatMax® PatQuick® SensorView® SmartLearn® SmartView® SMD4® UltraLight® Vision Solutions® VisionPro® VisionView®

The following are trademarks of Cognex Corporation:

3D-Locate™ 3DMax™ CheckPoint™ Cognex VSoC™ FFD™ iLearn™ InspectEdge™ Legend™ LineMax™ NotchMax™ ProofRead™ SmartAdvisor™ SmartSync™ SmartSystem™

Other product and company names mentioned herein are the trademarks, or registered trademarks, of their respective owners.

Preface.....	7
Audience	7
Organization	7
Conventions Used in This Guide.....	7
Introduction.....	8
OmniView Setup Wizard	9
OmniView Application and OmniView SDK.....	9
OmniView User Interface	9
Operator Panel	9
Event Counters	10
Job Details.....	11
Inspection Times	11
Supervisor Panel	11
Display Live Video	12
View an Image Archive.....	12
Start Shape Training	12
Configure Vision Tools.....	12
Loading Jobs	13
System Panel	13
Setting Passwords	13
Acquiring Images.....	14
View a Display Histogram	14
Perform Camera Calibration	15
Configure I/O Signals.....	15
Set Miscellaneous Parameters	15
Exit OmniView.....	15
OmniView Images.....	16
Image Acquisition.....	17
Image Databases	17
Saving Images to an Image Database	17
Loading Existing Image Databases	17
Saving Images Automatically.....	17
Continuous Acquisition.....	18
Calibration	20
Choosing a Calibration Plate.....	20
Accessing the Calibration Panel	21

Capturing Camera Calibration Images	22
Correcting a Failed Calibration Attempt	26
Performing a Validation on an Existing Calibration	26
Using Other Calibration Options.....	27
Performing a White Balance	27
Shape Training	30
Acquiring a Training Image	30
Starting Shape Training.....	31
Adding Control Points.....	32
Copying Shape Information	34
Training the Shape	34
Specifying a Trigger/Reject Index	35
Adjust Alignment Parameters	36
Use a Model Mask.....	38
Using Advanced Parameters.....	40
Train Parameters	40
Run Parameters	40
Unwrapping the Image.....	41
Adding Sub-align Regions	45
Using Multiple Unwrap Regions.....	46
Changing the Reference Image	47
Vision Tool Training	48
Accessing the Cognex OmniBuild Window	48
Choosing Additional Image Information	50
Adding Vision Tools.....	51
Analyzing Color Images.....	54
Enabling Critical Failures	56
Enabling Critical Failure Conditions	56
Using a Result Analysis Tool for Critical Failures	57
Collecting Statistics on Vision Tools.....	57
Testing the Application	59
Using a Custom Job Editor	60
Enabling a Custom Job Editor	61
Launching through the Operator Panel	62
Launching through the Supervisor Panel	62
Using Globally Accessible Variables	62

CogToolData.....	62
CogJobData.....	63
CogNamedDouble	64
CogNamedInt32.....	65
CogNamedString.....	65
Global Variable Sample Application.....	66
I/O Configuration	67
Configuring I/O Signals.....	69
Using Multiple Values for Trigger Delay and Rejector Positions	69
Wiring the I/O Board	70
External Control Interface	71
Supported Functions	71
Extending the Support Functionality.....	72
Data Logging.....	73
Logging Inspection Results	74
Logging Images.....	74
Frequently Asked Questions.....	75
The Configuration File.....	76

Preface

OmniView is a machine vision, cylindrical product verification system that can accurately inspect randomly oriented cylindrical containers.

Audience

This document is written for engineers who have been trained to use OmniView software, such as Original Equipment Manufacturers (OEMs) and Partner System Integrators (PSIs). The guide can also serve as a checklist for engineers who have previously performed these steps in a training environment..

Organization

This document describes how to create an inspection application using OmniView.

Conventions Used in This Guide

This document uses the following style conventions:

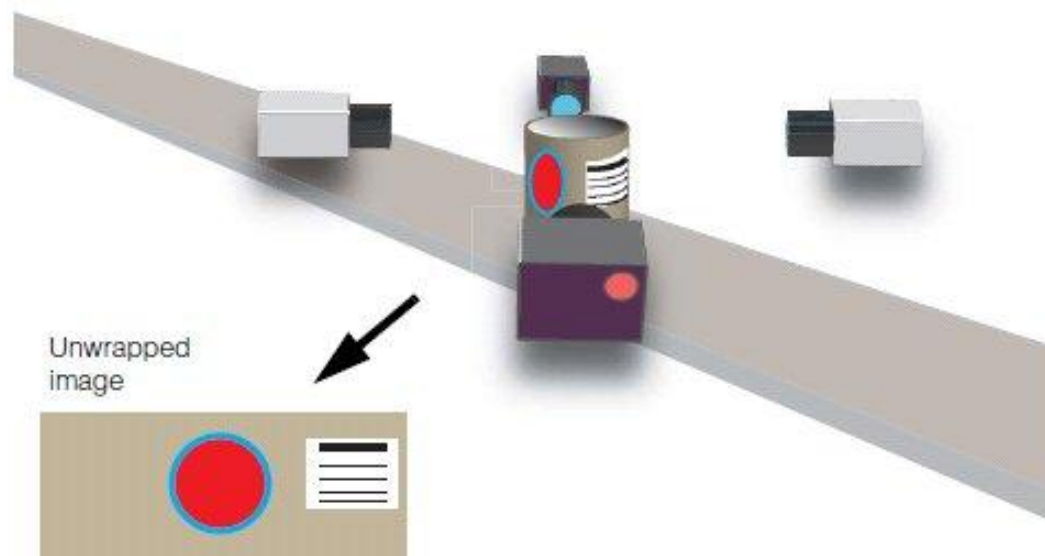
boldface	Indicates a specific component of the OmniView graphics user interface.
<i>italic</i>	Used for variables, arguments, file names, program names, book section names, and for emphasis.
<code>courier</code>	Used for program output.
bold courier	Used in illustrations to show what you would type on a keyboard.
<i><italic></i>	When enclosed in brackets, used to indicate keyboard keys such as <i><Tab></i> or <i><Enter></i> .

Introduction

This document describes how to create an OmniView Application job for inspecting randomly oriented cylindrical containers, which includes the following tasks:

- Calibrating your verification station
- Acquiring images of the containers you want to inspect
- Adding and training the vision tools you want to use to inspect each container
- (Optional) Logging results

Use the following diagram as a reference when reading this document:



A conveyor carries containers to be inspected as they move horizontally through the verification station, using four cameras to capture simultaneous images from all sides. OmniView blends the four images together to create a single unwrapped image containing an entire 360 degree view of the container, then registers the unwrapped image to a common starting point so that verification can be defined for relatively fixed regions of interest. This new aligned unwrapped image (or any of the four original images) can be analyzed with Cognex vision tools to verify any logos, dates, and product codes to ensure the container has been manufactured and labeled correctly.

An optional fifth camera can be used to capture an image directly above the container for an independent verification of the top.

OmniView is designed to operate on sufficiently rigid containers. Parts that suffer geometric deformation compared to the trained shape, or parts that have embossing, protrusions, or intrusions on the surface to be verified will show noticeable irregularities in the unwrapped image, especially in the horizontal direction. Moreover, labels with lumps, texture, or rough surface features will be unwrapped at noticeably lower fidelity than labels with smooth surfaces, even if the container itself is sufficiently rigid. This will adversely affect the ability for vision tools to run on the images.

Cognex requires you mount the side-view cameras counter-clockwise as viewed from the top. It is critical that cameras are mounted rigidly, especially with respect to each other.

OmniView Setup Wizard

Before you launch OmniView for the first time, you must run the OmniView Setup Wizard. The Setup Wizard establishes a configuration file that OmniView uses to store initialization parameters for your job. This configuration file stores information such as the make and serial numbers of any GigE Vision cameras you are using and the size of calibration plate you will use to convert the results generated by the vision tools you use into real-world measurements.

See the *OmniView Application Getting Started* for details on how to use the OmniView Setup Wizard.

OmniView Application and OmniView SDK

The OmniView software can be used in either discrete form:

- The *OmniView Application*, an interactive user interface for setup, management and operation of a vision application.

The term OmniView in this User's Guide is specific to the OmniView Application user interface.

- The *OmniView Software Development Kit (SDK)* provides a sophisticated .NET programming interface that provides access to camera calibration, acquisition, training, fixturing and unwrapping capabilities.

The OmniView kit ships with all the software for both OmniView Application and OmniView SDK. The two forms, however, cannot be used together to produce a single inspection application. You must choose between the interactive interface or the SDK before you start development.

For more information on the OmniView SDK, see the *Cognex OmniView SDK Programmer's Guide*.

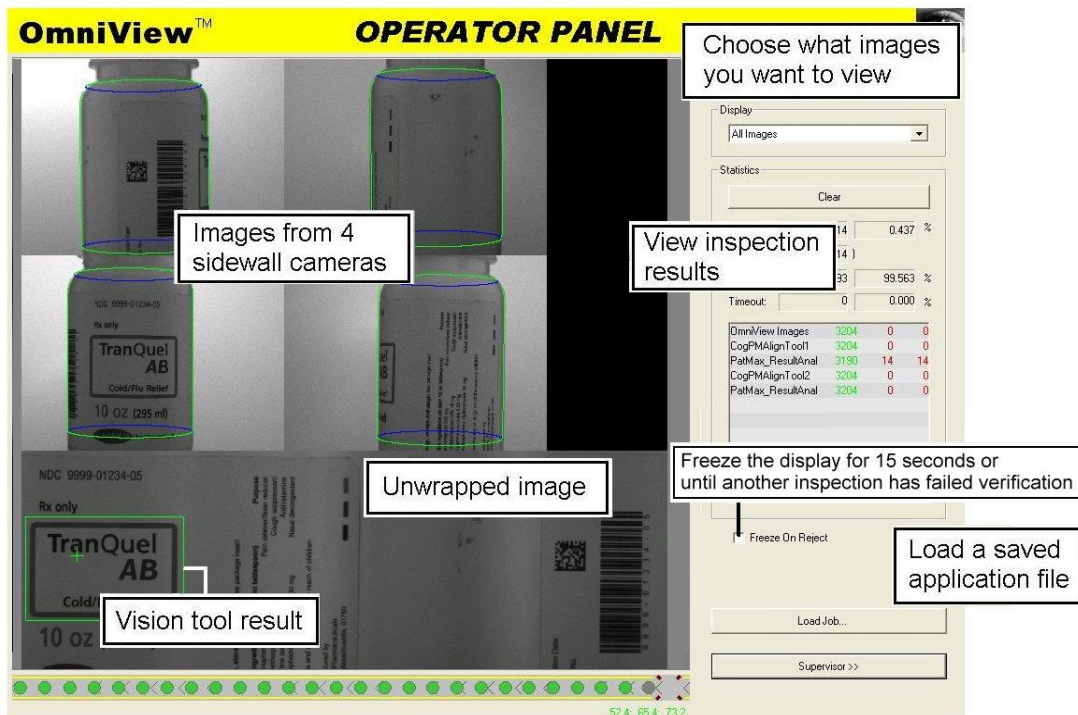
OmniView User Interface

OmniView Application uses a limited set of displays, or *panels*, that allow you to create your vision application and view the results it generates as it executes.

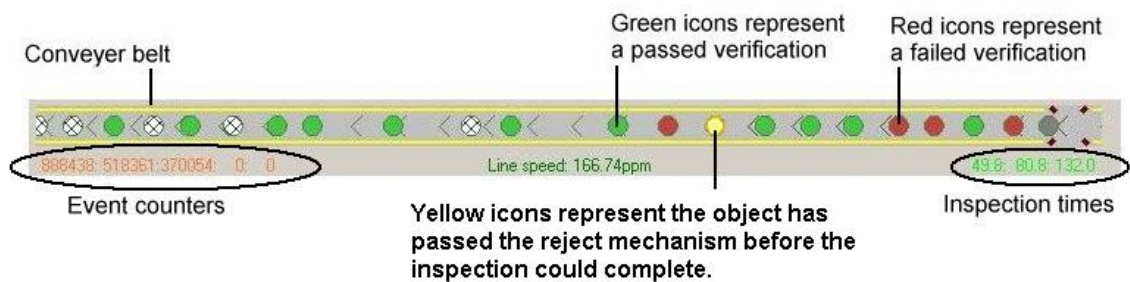
Operator Panel

The Operator Panel is the first panel OmniView displays each time it launches. OmniView automatically loads the last known job, so if your verification station has been prepared, your cameras have been calibrated and your verification station is receiving acquisition triggers from a part presence sensor, OmniView will start to acquire images immediately and perform verifications.

The following figure highlights major components of an example Operator Panel as an application executes:



The graphics along the bottom display information about the current job:



Event Counters

The Operator Panel displays four I/O trigger event counters from left to right:



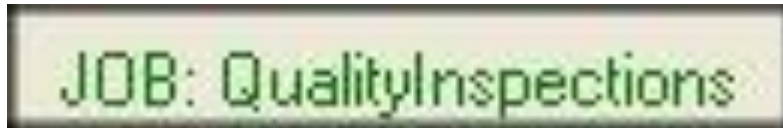
1. Valid part triggers received
2. Verification cycles passed
3. Rejection signals generated
4. Critical Failures generated

The fifth counter is not used in this release.

These counters roll over after 10 million counts, and restart from 0 each time you restart OmniView.

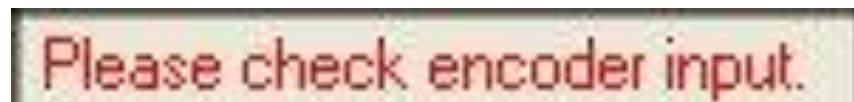
Job Details

During normal operation the bottom center of the panel displays information relating to the current job:



The display changes, cycling through the job name, line speed, and encoder speed.

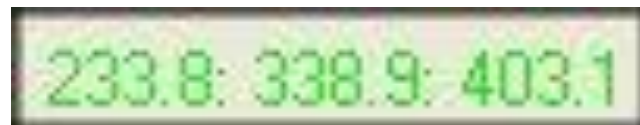
If OmniView detects an error that will interfere with verifications it will begin to display diagnostic messages in red, in place of the job information, as shown in the following example:



Refer to the *OmniView Application Getting Started* for a list of potential error messages and actions you can take to remedy the problem.

Inspection Times

The lower-right corner of the Operator Panel displays the minimum, latest, and maximum inspection time for past inspections, in milliseconds.

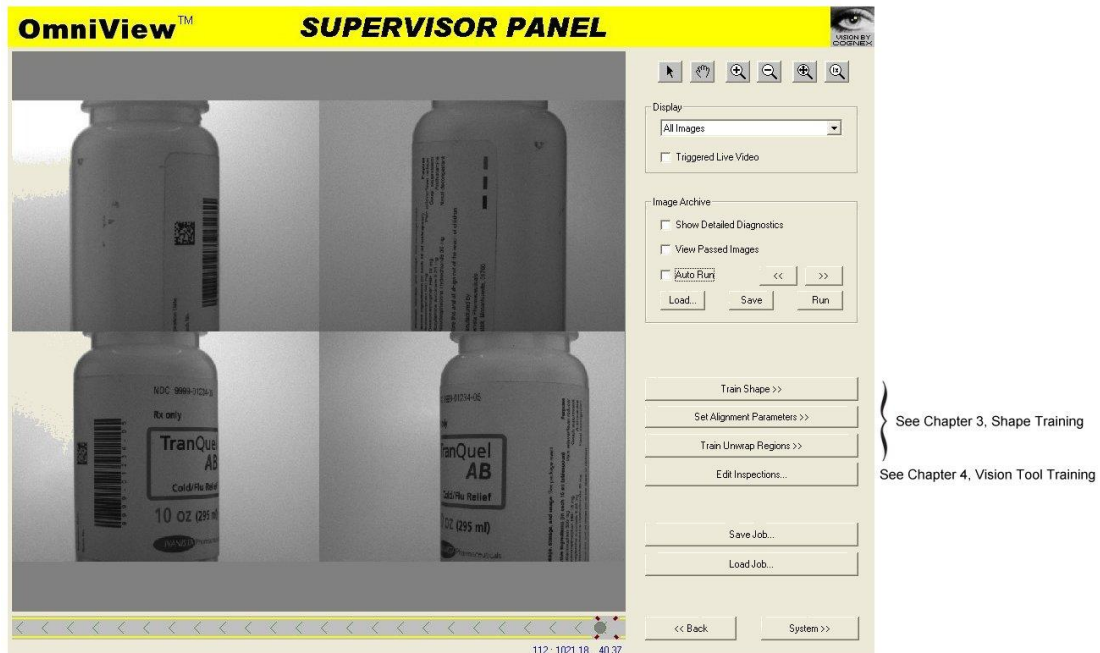


If no application is running, it displays the grey value followed by the column and row location image of the pixel currently under the pointer, relative to the upper left hand corner of the image.

Supervisor Panel

Click **Supervisor** from the Operator Panel to access the Supervisor Panel. Use the Supervisor Panel to create your vision application.

The following figure shows an example Supervisor Panel after the four side-view cameras have successfully acquired images of a container to be inspected:



Display Live Video

Check **Triggered Live Video** on the Supervisor Panel to display live images from the side-view camera(s) selected in the **Display** pull down list, based on triggers received by the part presence sensor.

View an Image Archive

Use the Supervisor Panel to view several types of saved images:

- Use **View Passed Images** to view the most recent 16 images that have passed (when checked) or failed (when unchecked) using the << and >> buttons.
- Enable **Auto Run** to perform an inspection on set of images you load with the << and >> buttons.
- Use the **Load** and **Save** buttons to view and save images. See the section *Saving Images to an Image Database* for details.
- Use the **Run** button to run an inspection on the current set of images.
- Use **Show Detailed Diagnostics** to view detailed graphics over images that fail an inspection. Be aware that this option is available only when you launch OmniView with the following command line: "omniview -okayToDiag"

Start Shape Training

Use the Supervisor Panel to perform shape training, where you define the edges of the containers you want to inspect.

See the section *Shape Training* on page 28 for details on shape training.

Configure Vision Tools

Use the Supervisor Panel to configure the vision tools you will use to analyze the unwrapped image of each container.

See the section *Vision Tool Training* on page 46 for details on vision tool training.

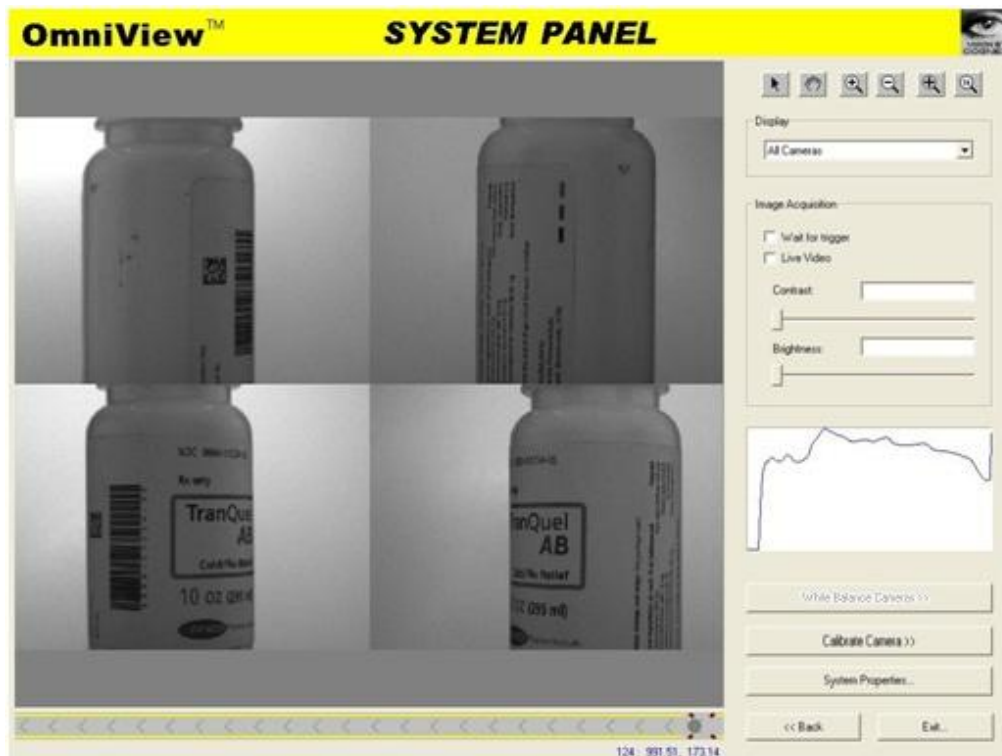
Loading Jobs

Use the Supervisor Panel to load an existing application or save the current application, which contains the buttons

Save Job and **Load Job**. OmniView saves all applications in the %OMNIVIEW_DATA%\jobs directory. Each time you launch OmniView, it automatically loads the last known job.

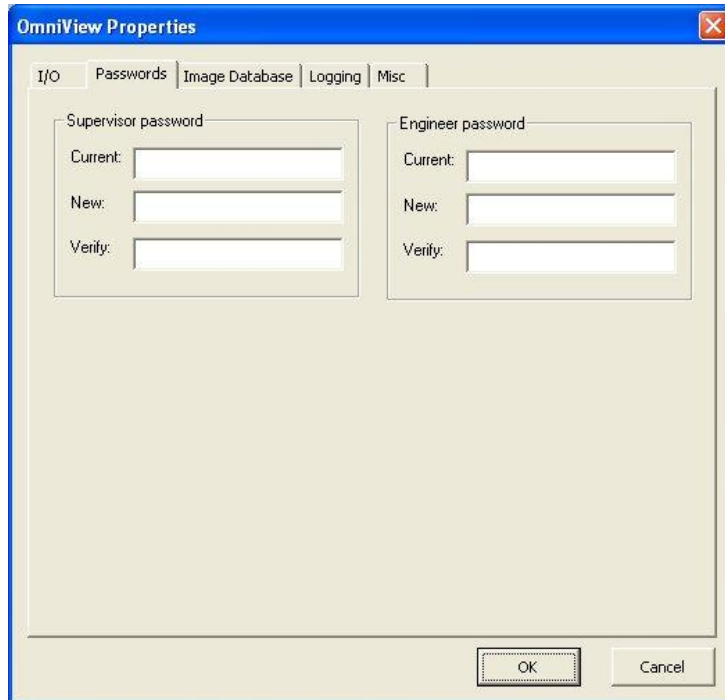
System Panel

Press **System** from the Supervisor Panel to access the System Panel, which you use to configure your I/O signals, calibrate your application, and configure other system properties. The following image shows a sample System Panel:



Setting Passwords

Access to the Supervisor Panels and System Panels can be controlled through password protection. By default, OmniView does not set any passwords. Click **System Properties**, using the **Passwords** tab:



The Supervisor password allows access to the Supervisor panel (shape training and vision tool configuration), while the Engineer password allows access to the System panel (calibration and data logging options).

Acquiring Images

Use the System Panel to acquire images using either method:

- Use **Wait for trigger** to perform an image acquisition when the part presence sensor has been triggered.
- Use **Live Video** to display live video from all four cameras.

Use the **Contrast** and **Brightness** controls for altering these values in your acquired images. OmniView represents these values using a numeric range of [0, 1.0] and are active only when individual camera views are selected for display. Any settings you specify apart from the default are saved as part of the job and override the settings in the OmniView configuration file.

The optimal verification stations require no adjustments using these controls. Cognex recommends that you use the aperture of the camera itself to perform most adjustments, using the controls for **Contrast** and **Brightness** only for fine tuning. Once your station has been calibrated, do not adjust the aperture settings any further.

View a Display Histogram

The System Panel displays a relative histogram of individual camera images, and will display cumulative histogram information for all side cameras if no individual camera is selected.

The horizontal axis represents pixel values starting at 0 (left) through 255 (right). The vertical axis represents the number of pixels in the image for each pixel value. A flat histogram showing balanced brightness is desired.

Perform Camera Calibration

Use the System Panel to calibration your verification station, which must be calibrated in order to accurately blend the four separate images from the side-view cameras into one continuous image.

See the section *Calibration* on page 20, for details on calibration.

Configure I/O Signals

Use the System Panel to access **System Properties** and configure the I/O signals your application will use to communicate with external equipment, such as the part presence sensor and the reject mechanism.

See the section *I/O Configuration* on page 64 for details on I/O signals.

Set Miscellaneous Parameters

Use **System Properties** to set miscellaneous parameters such as password protection and the ability to inspect images in an image-database file.

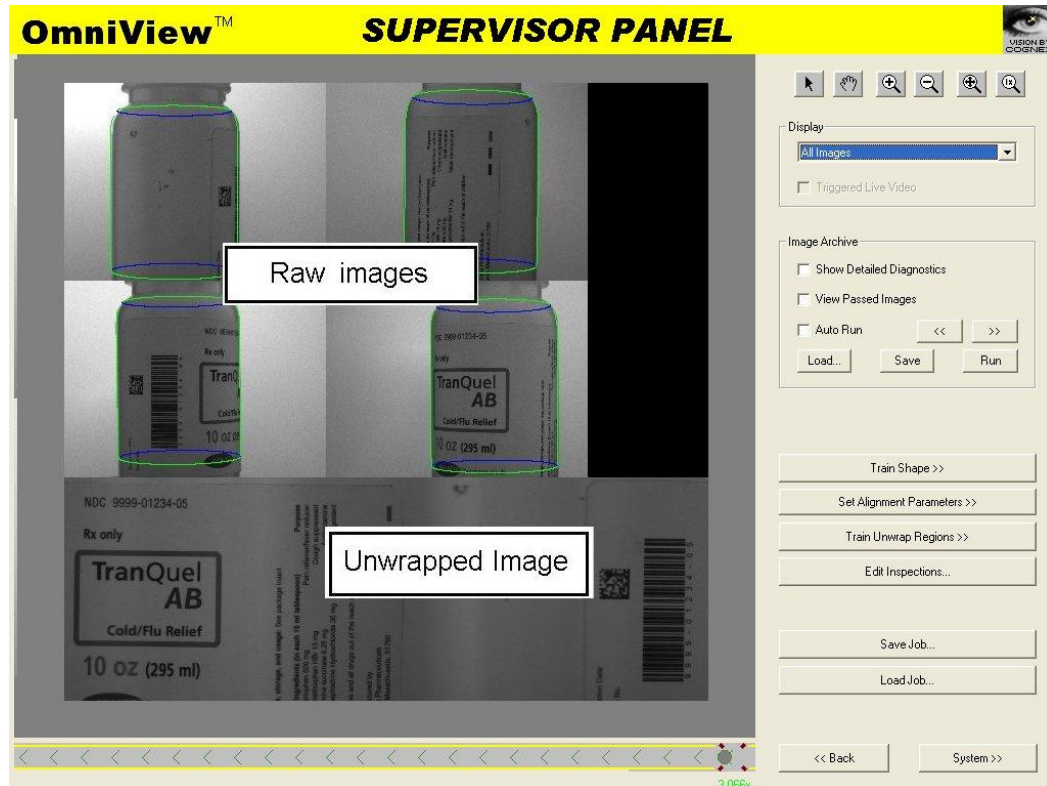
Exit OmniView

The only way to exit OmniView is through the System Panel.

When exiting OmniView or its Setup Wizard, allow a few seconds for the application to exit completely. If you attempt to launch another instance of OmniView or a different OmniView application (such as the Setup Wizard), you might receive an error message and the program you are trying to launch will terminate.

OmniView Images

Each inspection requires a simultaneous acquisition from four (or five cameras) within the verification station. By default, OmniView displays all images it generates for each inspection, as shown in this example of a Supervisor Panel:



Use the **Display** pull down list, available from any panel, to choose which images you want to view at any time.

Option	Displays
Camera 1, 2, 3, 4 or 5	The most recent image from the camera you select: 1, 2, 3, 4 or 5
All Cameras	The most recent image from each camera on your system
Region Unwrap	The unwrapped label regions you have trained If you trained more than one region, all trained regions are displayed.
All Images	All camera images plus the unwrapped image across the display bottom.
All VPI Cameras	All of the four side-view cameras (cameras 1 through 4), excluding camera 5 if it is present

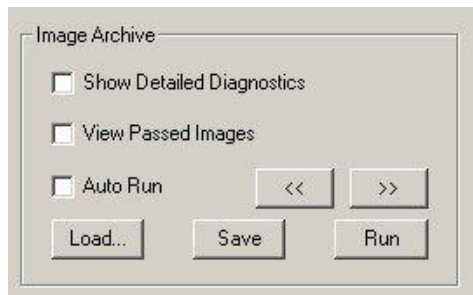
Image Acquisition

Capture simultaneous images from all the connected cameras using either of these methods:

- Check the **Triggered Live Video** checkbox on the Supervisor Panel.
OmniView will capture an image when the part presence sensor detects a container.
- Use the **Wait for trigger** and **Live Video** options on the System Panel.
Check **Live Video** for a real-time display of images from all connected cameras. If you also use the **Wait for trigger** option, OmniView will wait for an external trigger before displaying live images.

Image Databases

As your job executes, OmniView allows you to save acquired images and then load them again later. This can be useful for testing and developing your application before you deploy it to a production environment. Use the **Image Archive** section of the Supervisor Panel, highlighted in the following figure, to save and load images:



Saving Images to an Image Database

For each inspection, OmniView acquires a viewset of four (or five) images of each container. Click **Save** to save the last 16 viewsets to image database files. OmniView differentiates between passed and failed verifications when saving viewsets, storing passed viewsets and failed viewsets in different image-database files. By default, OmniView saves the image database files in the directory *C:\Program Files\Cognex\OmniView\data\Images*, and uses a "date" + "time" stamp to label each file. For example, clicking **Save** can generate the following files:

090810_113707_passed.cdb
090810_113707_rejected.cdb

These filenames indicate the files were created on August 10, 2009 at 11:37 AM.

Loading Existing Image Databases

You can load saved image-database files at any time and test the images they contain against the current job, which can be helpful if you are analyzing why failed viewsets did not pass inspection or to verify that previous results continue to pass verification.

Click **Load** to load existing passed and rejected image-database files. Once loaded, you can navigate through the images the file contains using the << and >> buttons.

Saving Images Automatically

You can configure OmniView to create image-database files of passed and failed verifications automatically by performing the following steps:

1. Go to the System Panel.
2. Click **System Properties**.
3. Select the **Image Database** tab.
4. Check the **Enabled** checkbox under **Image Database Archiving**.
5. (Optional) Choose a different interval for how frequently the images are saved.

By default the images are stored in the directory *C:\Program Files\Cognex\OmniView\data\images*, but you can specify a different archive directory.

Continuous Acquisition

In some instances, you might want to simulate an application running in production mode by cycling through all the images in an image-database file automatically. This requires an OmniView .scp configuration file, which lists the image databases you want OmniView to open and scan. You can specify as many image-database files as you want in the .scp file.

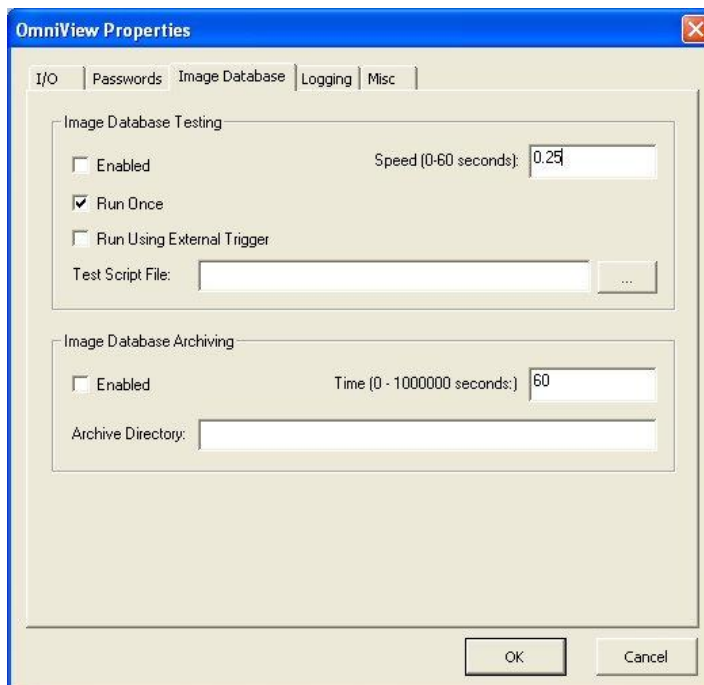
Each line in the .scp file contains a directory path and image file name, as shown in the following example lines:

```
C:\Program Files\Cognex\OmniView\data\Images\070515-4oz-jar1.cdb  
C:\Program Files\Cognex\OmniView\data\Images\070515-4oz-jar2.cdb
```

To configure OmniView to acquire images from image-database files specified in an .scp file, perform the following steps:

1. Go to the System Panel.
2. Click **System Properties**.
3. Select the **Image Database** tab.

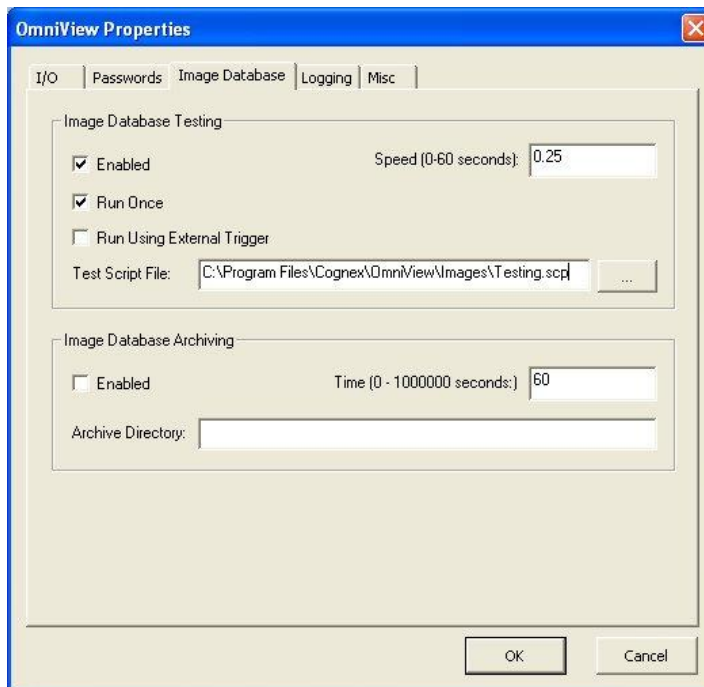
The tab appears as shown:



4. Check **Enabled** under **Image Database Testing** and use the **Speed** value to control how fast OmniView cycles through the images.

5. Select the .scp configuration file that lists the image-database files you want to use.

The following figure shows an example **Image Database** tab configured to run image-database testing:



6. Keep the **Run Once** option checked to run each image in the database only once, or clear the checkbox to allow OmniView **to cycle** through the images in an endless loop.
7. Click **OK** and return to the Operator Panel.

Once you return to the Operator Panel, OmniView should automatically begin to acquire images from the image database file. If you have performed unwrap training and vision tool training, OmniView will simulate a complete application on the images in the image file.

Calibration

Before you can capture multiple images of your container and allow OmniView to blend them together to form a single image, you must calibrate OmniView for the cameras and verification station you are using. The calibration process allows OmniView to create a mathematical transform that can be applied to raw camera images and correct any distortion created by the intrinsic nature of the cameras. Once calibrated, you can begin to train OmniView with the contours of the containers you want to analyze, and then add vision tools to analyze the unwrapped image OmniView will generate.

It can become necessary to perform the calibration process again later. In fact, you must recalibrate your vision application whenever you change an existing camera in your verification station, refocus a lens, change an aperture setting or when recommended by the calibration validation option.

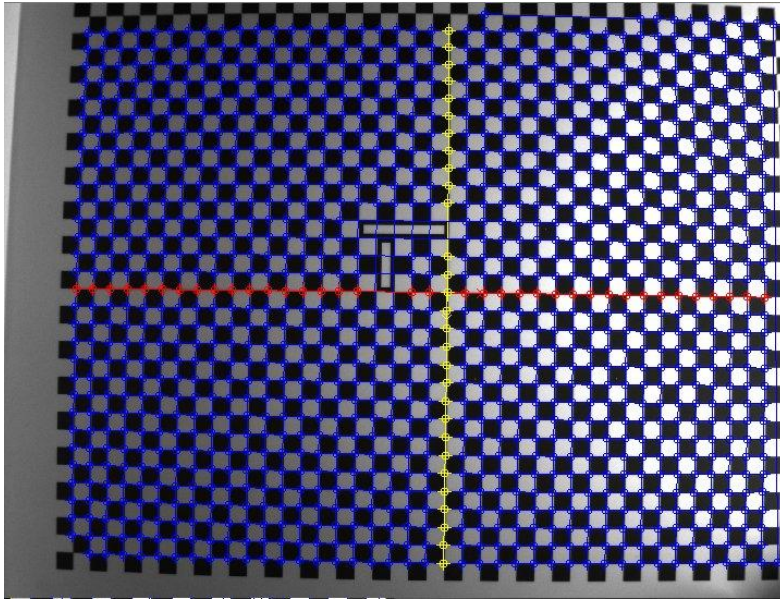
Choosing a Calibration Plate

OmniView supports three different calibration plates consisting of a checkerboard pattern along with a fiducial mark located near the center. You specify which plate you want to use for the calibration process when you use the Setup Wizard to configure your new OmniView application, as described in *OmniView Application Getting Started*. The choice of which calibration plate to use depends on the size of the container you want to verify:

Plate Size	Diameter of Container	Description
Small	1" or less	1/16" grid square size (1/16" = 0.0625 inches = 1.5875 millimeters)
Medium	1" to 3"	1/8" grid square size (1/8" = 0.125 inches = 3.175 millimeters)
Large	3" or greater	1/4" grid square size (1/4" = 0.250 inches = 6.35 millimeters)

The calibration process involves capturing five separate images of the calibration pattern for each camera, with the position of the grid changed from one image to the next. A final image is then captured with the calibration plate facing camera 1 and camera 2 equally. OmniView does not require calibration information from a possible 5th camera you might be using to capture an overhead view. The following figure shows an example calibration

image taken by one of the cameras:



The fiducial mark must be fully visible to the camera undergoing calibration.

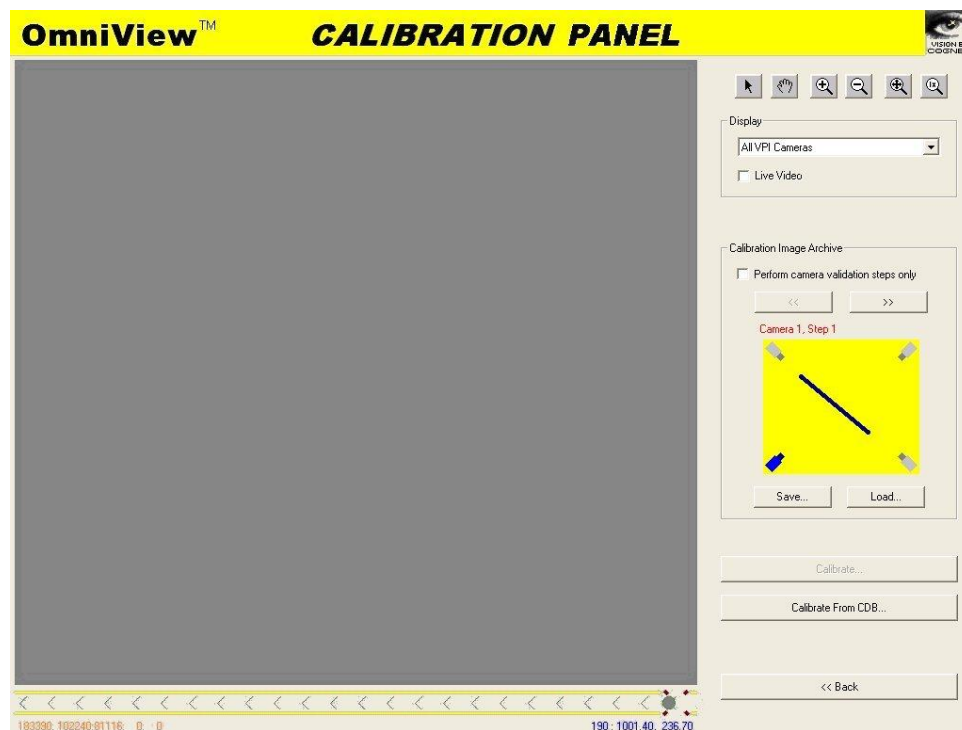
In some production environments, the calibration plate is mounted within a device that allows it to be rotated and tilted as necessary for each calibration image while allowing a single operator to perform the calibration process.

Accessing the Calibration Panel

You must use the Calibration Panel to perform a calibration procedure. To access the Calibration Panel:

1. Click **Supervisor** to access the Supervisor Panel.
2. Click **System** to access the System Panel.
3. Click **Calibrate Camera** to access the Calibration Panel.

The Calibration Panel appears as shown in the following figure:

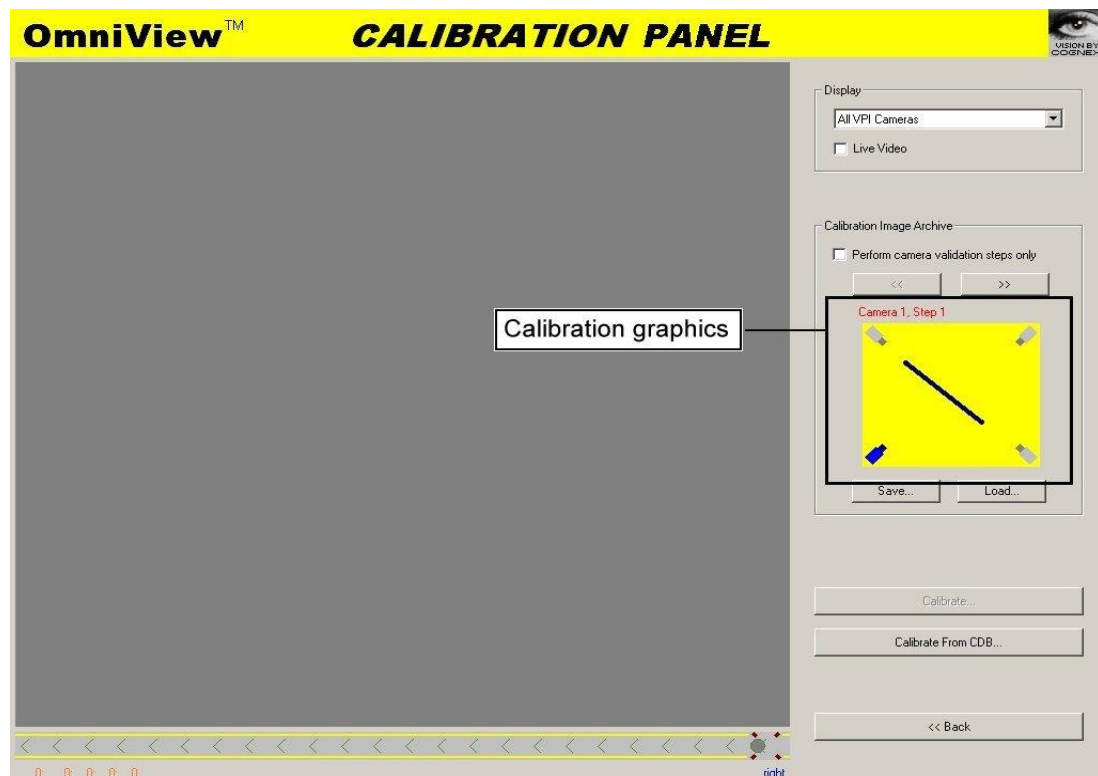


If you have not confirmed that the verification station is able to acquire satisfactory images of your containers, Cognex recommends you do so before beginning the calibration process. Refer to the *OmniView Application Getting Started* for details on acquiring your first set of images.

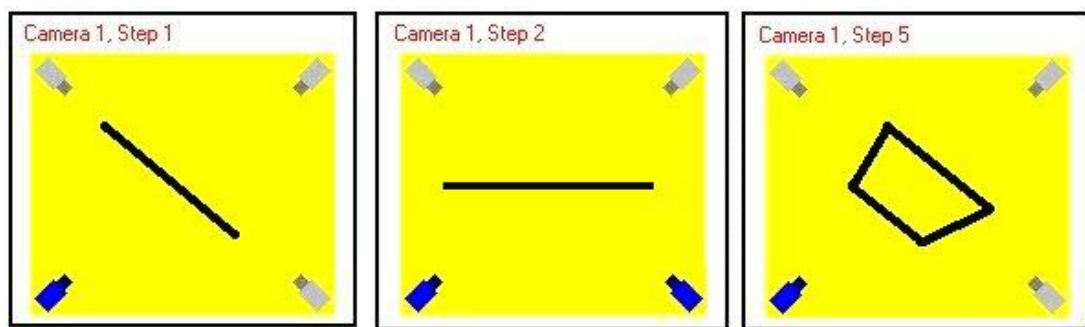
Capturing Camera Calibration Images

Once you have confirmed that your verification station is ready to acquire images, replace the sample container you want OmniView to analyze with the calibration plate you will use. During the calibration process, the Calibration Panel provides graphics to help you position

the calibration plate, as shown in the following figure:



As you perform calibration, the graphics will show you how to present the calibration plate to each camera as from a top view, as shown in the following examples:



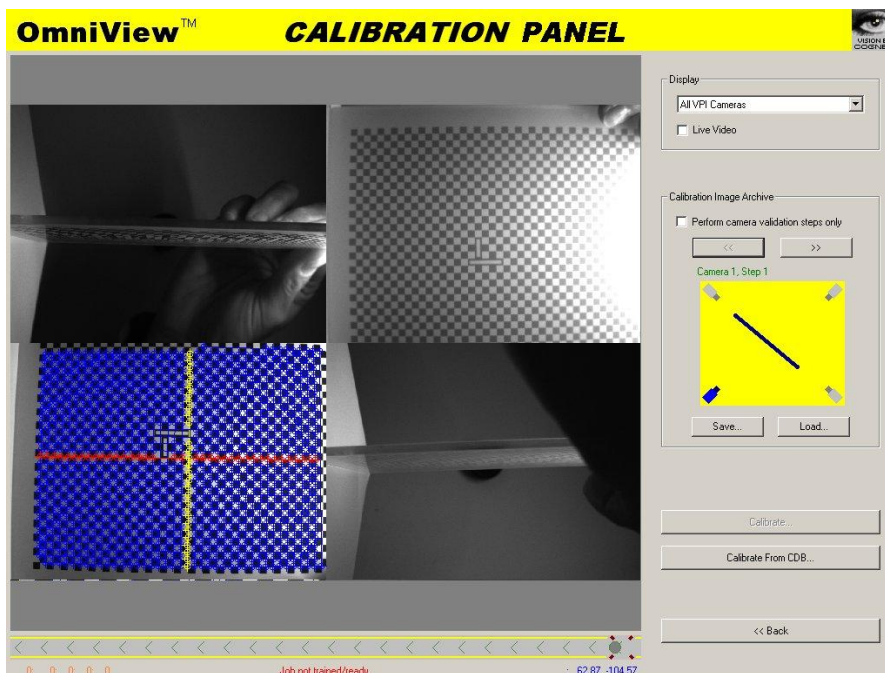
In addition, the label **Camera x, Step x** appears in either red to indicate that the proper calibration image has not been provided, or in green to indicate that the latest acquired image provides sufficient calibration information to OmniView.

Perform the following steps to capture the calibration images OmniView requires:

1. Check the **Live Video** checkbox.
2. Place the calibration plate in the center of the verification station directly in front of Camera 1.

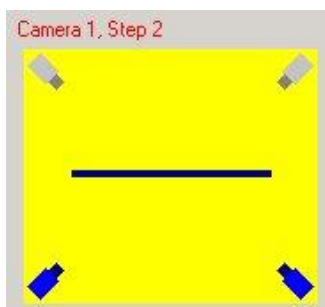
Keep the grid as close to the center of the station as possible and verify that the fiducial mark can be seen clearly.
3. Uncheck the **Live Video** checkbox.

OmniView acquires the image and attempts to locate the fiducial mark and identify the checkerboard pattern. Each step in the calibration process can take several seconds. If successful, the Calibration Panel appears similar to the following figure:



In the previous figure, notice how the label for **Camera 1, Step 1** appears in green to indicate a successful calibration image. If the label still appears in red, the calibration step was not successful and you must take another calibration image.

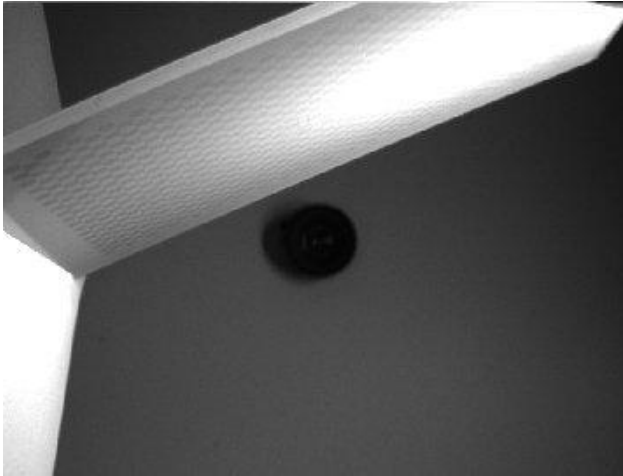
- Click **>>** to continue and acquire the next calibration image for Camera 1, which requires you to present the plate to Camera 1 and Camera 2 along the direction of the conveyor flow, as illustrated:



- Continue acquiring calibration images for Steps 3, 4 and 5 of Camera 1.

For Steps 4 and 5, you must tilt the calibration grid so that the top is closer to the camera and then further away, respectively. For example, this would be the view from Camera 2 when you are performing Step 4 for Camera 1, where the grid is tilted

toward the camera:

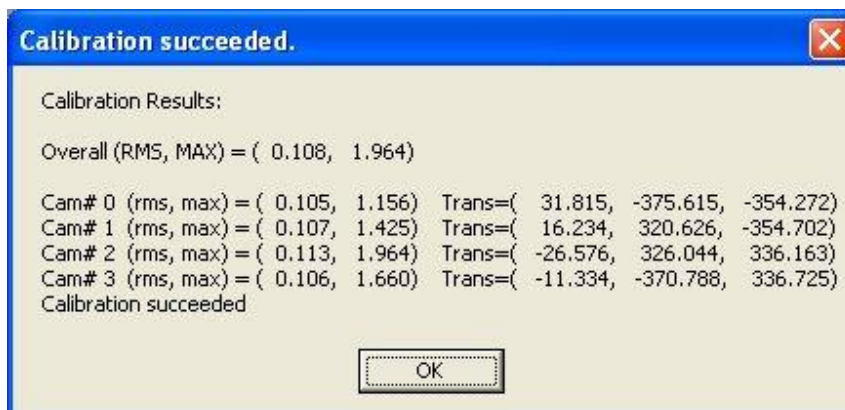


Once you have completed Steps 1 through 5 for Camera 1, you must repeat this procedure for Camera 2, Camera 3 and Camera 4, and then capture a final calibration image as shown:



Click **Calibrate** when the button is enabled.

If OmniView is able to successfully generate the calibration transform based on your setup, a popup appears as shown in the following example:



Once calibrated, you are ready to begin shape training for the containers you want to analyze.

Note: Be aware that memory usage may increase dramatically after calibration. For example, memory usage can go from 20 MB to over 210 MB once calibration has

completed. OmniView reserves this memory in preparation for the incoming images from the 4 (or 5) cameras in your verification station.

Correcting a Failed Calibration Attempt

If OmniView reports a failed calibration attempt when you click **Calibrate**, or if OmniView does not enable the **Calibrate** button, review the images captured along each step of the calibration procedure:

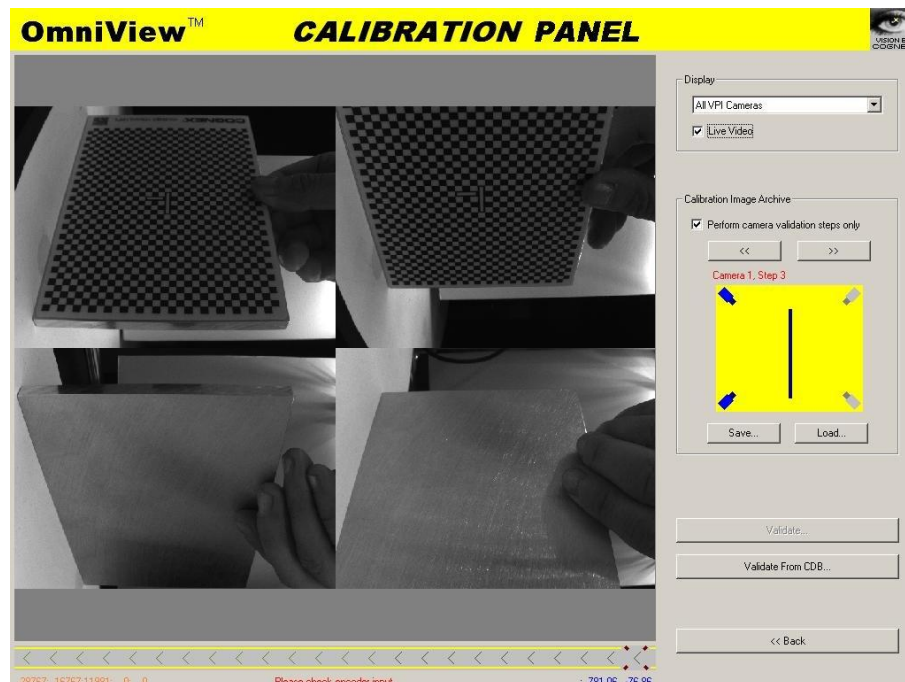
- Verify that the label **Camera x, Step y** appears in green for each calibration image.
- Verify that the images are focused even when the calibration grid is angled towards or away from the camera.
- Increase the lighting within the verification station to improve the contrast between light and dark squares.

Review the images captured along each step of the calibration procedure and recapture images for any calibration steps that were unsuccessful, which are indicated when the label **Camera x, Step y** appears in red.

Performing a Validation on an Existing Calibration

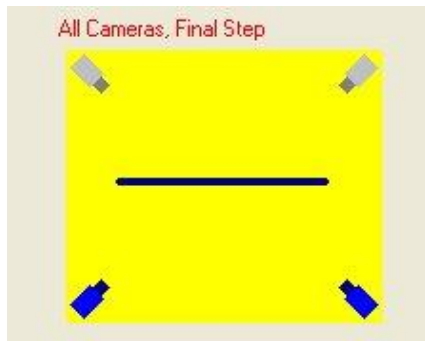
You can validate an existing calibration by checking the **Perform camera validation steps only** checkbox and then capturing images of the calibration plate in the positions indicated in the Calibration Panel. Perform a validation at any time to ensure the calibration information currently available to OmniView is still valid.

The following figure shows a user presenting a calibration plate to the verification station in the first step of the validation procedure:



As you acquire images for the validation, be sure to position the calibration plate in the same orientation (horizontal or vertical) you used for the final step of the calibration process, which consists of presenting the plate between cameras 1 and 2 as shown in the

following figure:



For example, if you positioned the plate in a horizontal orientation during calibration, you must present the plate in a horizontal orientation again during validation. Attempting to validate the current calibration using a vertical orientation can cause OmniView to fail the validation attempt.

Once the validation images have been acquired, click **Validate**. OmniView compares existing calibration data with the validation images and notifies you if a complete recalibration is necessary.

With this option enabled, the button **Calibrate from CDB** becomes **Validate from CDB**.

Using Other Calibration Options

The Calibration Panel offers other options you can choose to use in your efforts to calibrate your application:

Save	You can save the current set of calibration/validation images into an image-database (.idb) or Cognex database (.cdb) image file. Cognex recommends you save all calibration/validation images for future reference.
Load	You can load a set of saved images and use them for calibration/validation.
Calibrate/Validate from CDB	You can load a set of saved images from a Cognex database (.cdb) and allow OmniView to attempt to calibrate/validate the application using this set of images. OmniView reports an error if you choose an incomplete image set or if the image set does not include images of a calibration plate.

Performing a White Balance

If you are using any combination of color cameras in your OmniView enclosure, you have the option of performing a white balance adjustment, which can help to ensure that the colors in your acquired images are accurate for your production environment. Cognex recommends performing a white balance after the calibration process.

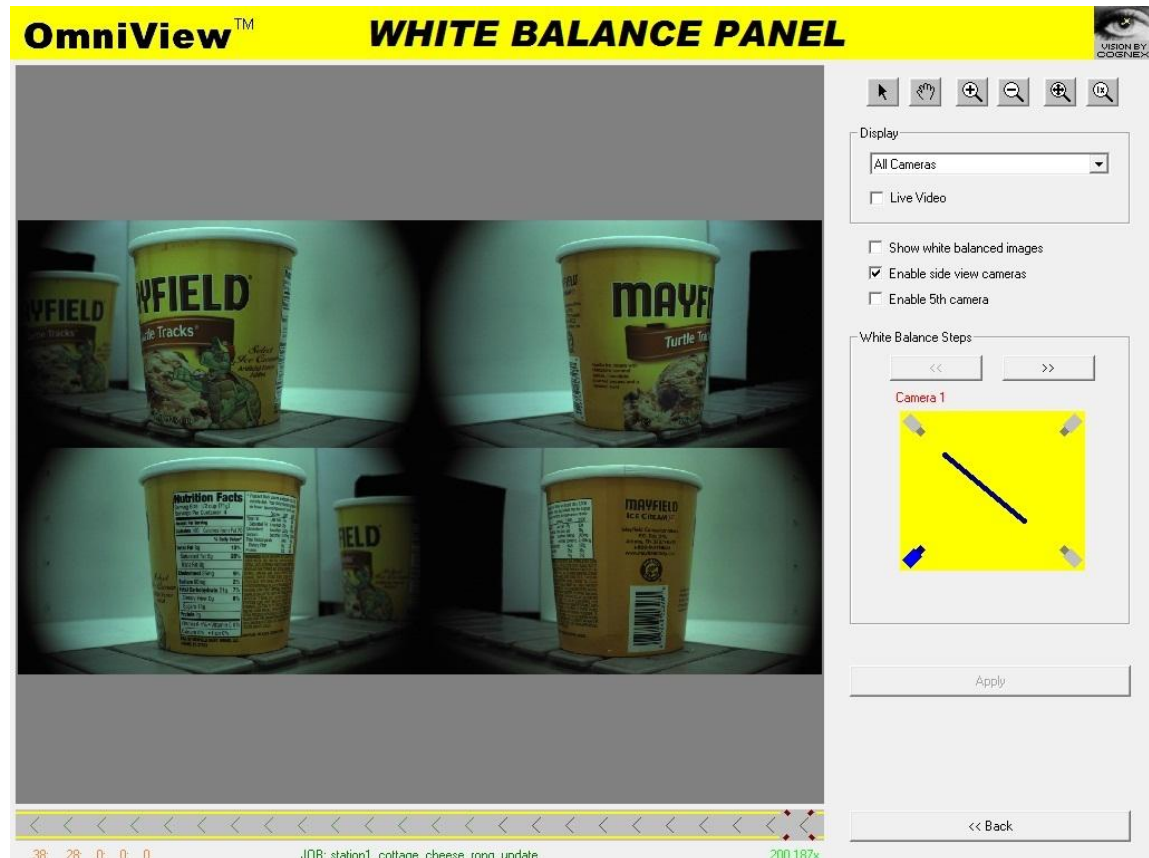
The white balance adjustment is only available if you are using color cameras, and the white balance settings the procedure generates are specific to the Basler color cameras that OmniView Application supports. By default, OmniView does not attempt to perform a white balance on your color cameras.

A white balance can be performed with the calibration plate or a white piece of paper. You cannot change the computed white balance values.

You must use the White Balance Panel to perform a manual white balance. To access the White Balance Panel:

1. Click **Supervisor** to access the Supervisor Panel.
2. Click **System** to access the System Panel.
3. Click **White Balance Cameras** to access the White Balance Panel.

The following figure shows the start of an example white balance:



The graphics in the White Balance Panel indicate which camera is ready to perform a white balance. Perform the following steps:

1. Present the white balance target (calibration plate or white piece of paper) to the camera indicated by the graphic (Camera 1 in the previous figure).
The target must occupy at least 50% of the field of view of the camera and must not be saturated from the available light source.
2. Check **Live Video** and allow OmniView to capture a live image.
3. Uncheck **Live Video** and allow OmniView to generate the white balance values for this camera.

The label for the camera in the camera graphic turns from red to green after OmniView has successfully generated white balance information for this camera.

4. (Optional) Check **Show white balanced images** to display the white balanced image in the White Balance Panel.
5. Click **>>** to move to the next color camera in your enclosure, relocate the white balance target, and use **Live Video** to capture a live image from this camera.
6. If you are using a color camera as your 5th camera, present your white balance target to this camera and use **Live Video** to capture a live image.
7. Click **Apply** once all the color cameras have captured images of the white balance target.
8. Click **<<Back** and return to the System Panel.

Shape Training

OmniView must be trained with information regarding the shape of the container you want to analyze before it can reliably and repeatedly blend four separate images into one unwrapped image.

To perform *shape training*, you must capture images of the container you want to analyze and use a set of interactive graphics to define the boundaries of the container for each of the four cameras in your verification station. Shape training ensures that each camera contributes the correct image information to the final unwrapped image. Shape training does not apply to a 5th camera you might be using to capture an overhead view of the container.

You must perform shape training before you can specify which vision tools you want to use to analyze the unwrapped image. Before you perform shape training, you should calibrate your application. See the section *Calibration* on page 20 for details on how to configure your OmniView application.

OmniView is designed to operate on sufficiently rigid containers. Parts that suffer geometric deformation compared to the trained shape, or parts that have embossing, protrusions, or intrusions on the surface to be verified will show noticeable unwrap irregularities, especially in the horizontal direction.

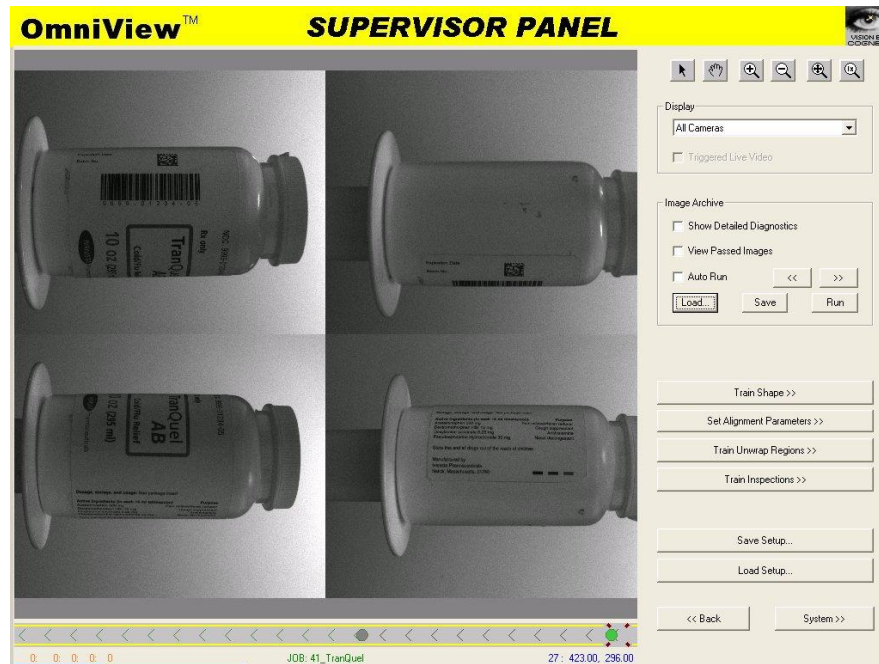
Acquiring a Training Image

To begin shape training you must have an image of a good container from all four cameras. Use either method to acquire this image:

- Place a known good container on the conveyer and allow it to pass through the verification station with the **Triggered Live Video** checkbox enabled in the Supervisor Panel.
- Enable **Live Video** in the System Panel, place the container at the center of your verification station, and uncheck **Live Video**.

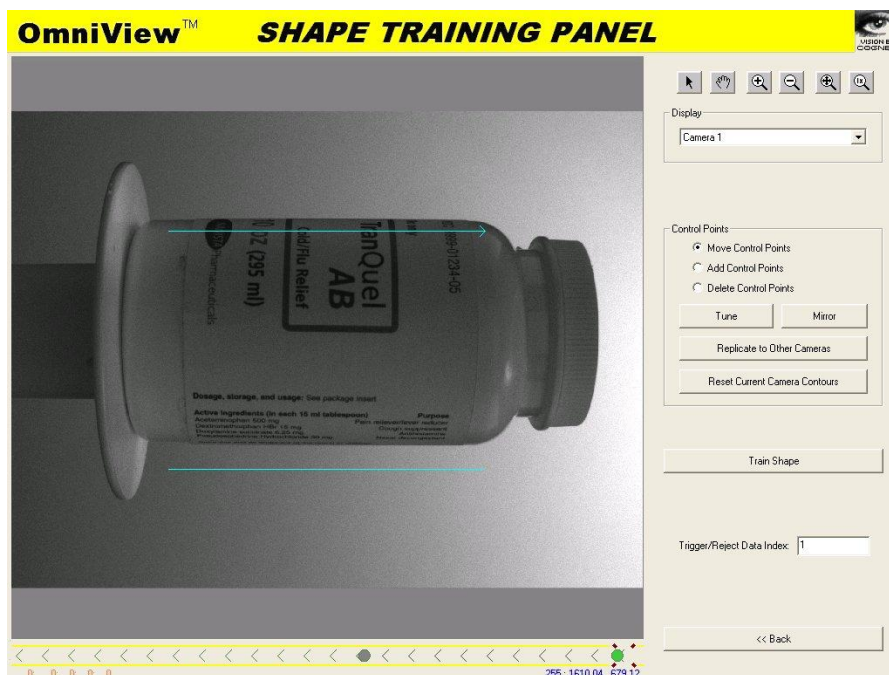
Shape Training

Depending on how your cameras are mounted, the images will contain either horizontal or vertical images of the container you want to analyze, as shown in the following figure:



Starting Shape Training

You must use the Shape Training Panel to perform shape training. Click **Train Shape** from the Supervisor Panel to access the Shape Training Panel, as shown in the following example:

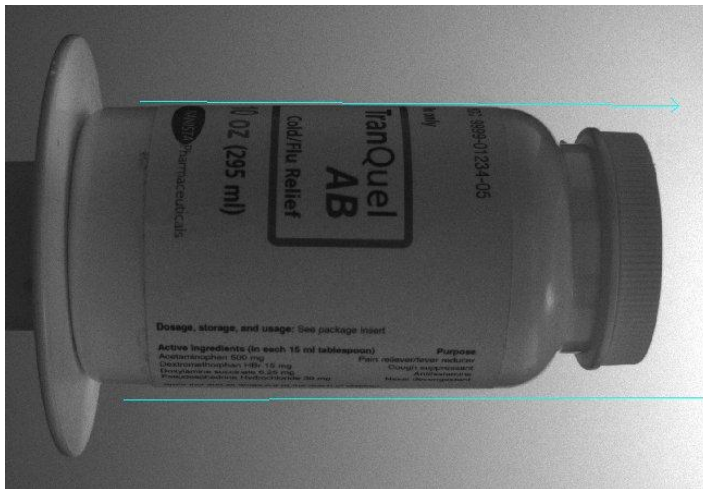


Shape Training

By default, the Shape Training Panel displays the image captured by Camera 1, which you can use to perform the bulk of shape training. Later, you can copy the shape information you provide for this image to the images for the other three cameras.

The Shape Training Panel provides two contour graphics that you can manipulate and shape to the general contour of your container. The graphics can appear either in a horizontal or vertical orientation depending on your camera mounting scheme, but it does not affect the unwrapped image, which corresponds to a top-down scan of the surface. One of the graphics has an arrow to indicate the positive Z-axis direction, which points to the top of the container and appears using orange graphics as the job executes. Use this contour graphic to indicate which direction is up in your verification station, which can be helpful later as OmniView captures images in your production environment.

Click on a graphic to highlight it and display the resizing handles that allow you to move either end of the graphic as well as change its length. To start shape training, adjust the size and position of both contour graphics so that they surround the edges of the label you want to inspect, as shown in the following figure:



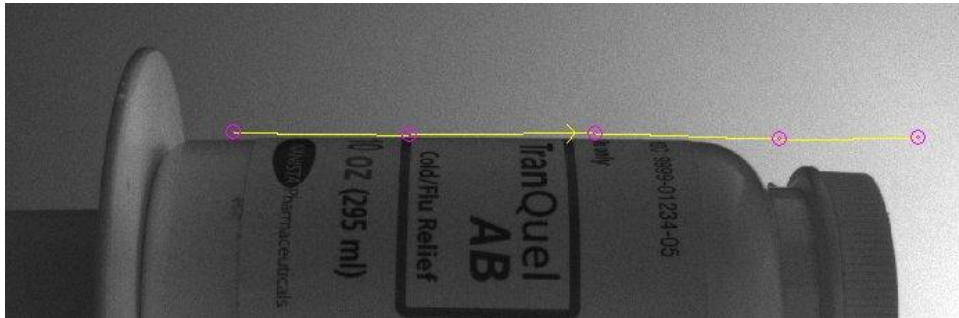
Adding Control Points

To fit the contour graphics to the edges of the container more precisely, you need to use *control points*, which allow you to bend the contour graphics to fit the edges of the container precisely. Check **Add Control Points** to add control points. When enabled, OmniView adds a new control point any place you click on the selected contour graphic.

Check **Move Control Points** to bend and reshape the contour graphic around the shape of the contour in the image.

Shape Training

The following figure highlights some of the control points added to a contour graphic:



Use the following tips as you work with the contour graphics:

- Add and adjust control points as necessary as you work from the top down (or left to right) of the container.

Use the **Move Control Points** option to move existing control points.

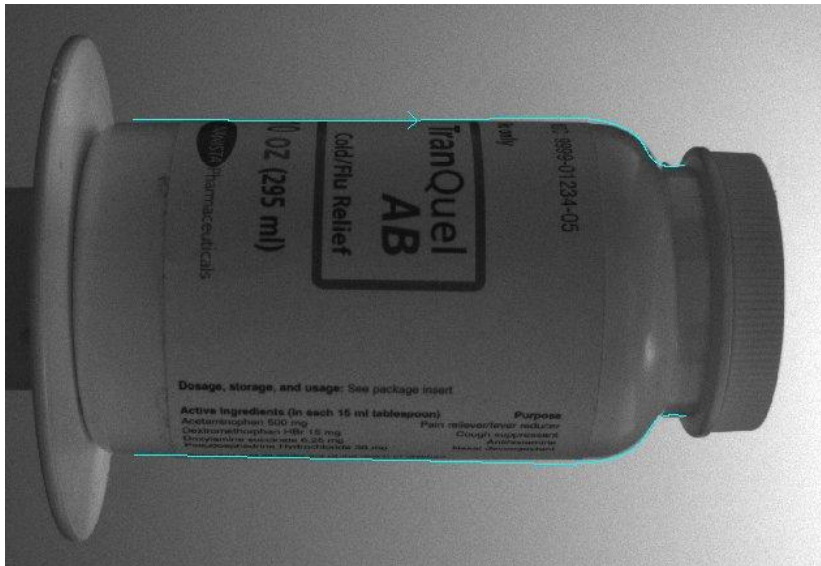
- Add more control points to make it easier to shape the contour graphic around curves of the container.
- Click **Tune** periodically and have OmniView attempt to fit the contour graphic around the edges of the container based on the current shape of the contour graphic.
- Click and drag a control point to change its position along the contour graphic.
- Check **Delete Control Points** and click on control points you do not need.
- Click **Reset Current Camera Contours** at any time to start over.

When you are finished, the contour graphic should closely fit the edges for one side of the container, as shown in the following figure:



Next, click **Mirror** to have OmniView attempt to apply the information from the fitted contour graphic to the unfitted contour graphic, which should result in both graphics

roughly fitting the shape of the container, as shown:



Be aware that the operation may not have any effect if OmniView fails to apply the contour to the unfitted side. In this case, you need to create the contour from scratch as you did with the first one. Select the graphic and position it across your container as necessary.

Copying Shape Information

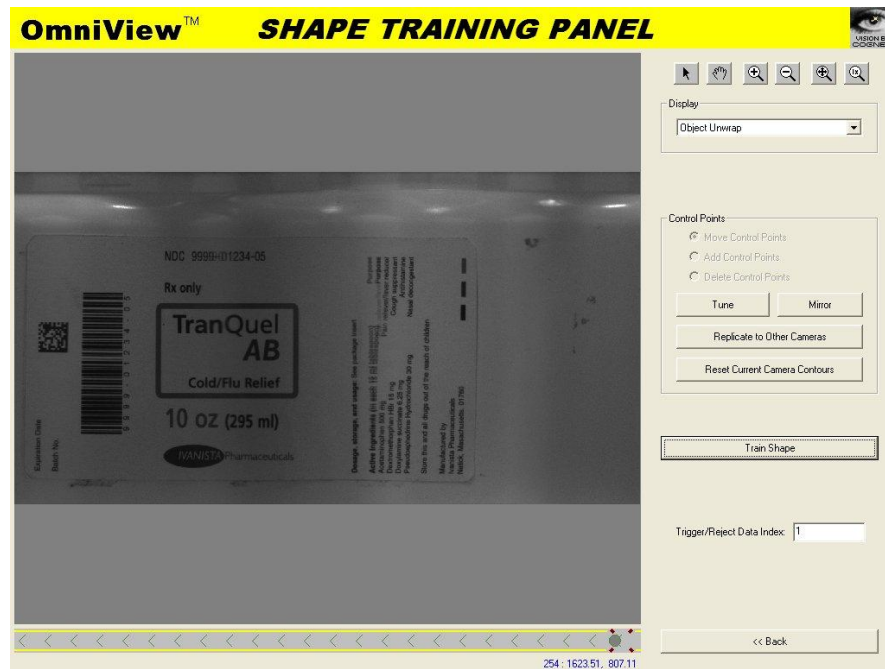
Once you are satisfied with the shape training from Camera 1, click **Replicate to Other Cameras** to copy this image information to the other three cameras in your verification station. OmniView takes the contour information and applies it to the images from the other three cameras.

Cognex recommends you check the display from the other three cameras to ensure that the graphics closely fit the shape of the container. If the contour graphics do not closely fit the shape of the container in all four images, you must select the graphics for those images as necessary and improve the fit.

Training the Shape

Click **Train Shape** to train your application with the shape of the container. OmniView prompts you for the System password if it is set. If successful, OmniView blends the images from all four cameras and displays the new blended image, as shown in the

following figure:



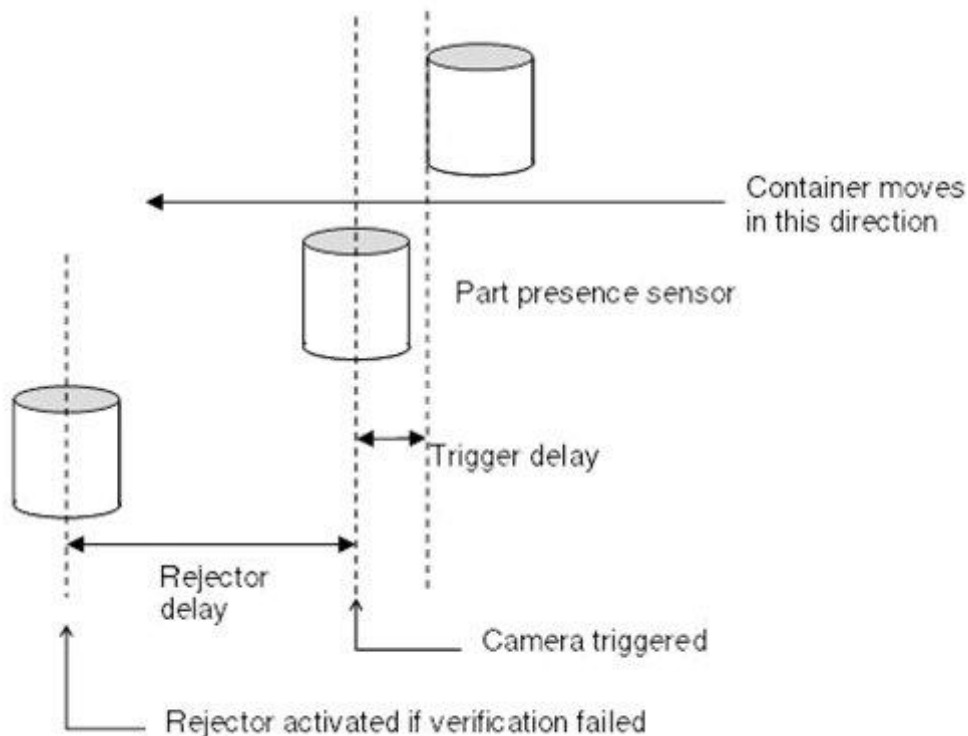
If the unwrapped image shows any distortion in the blended regions, try either of the following:

- Perform a calibration validation as described in the section *Calibration* on page 20.
- Check the shape contour information for each camera and perform shape training per camera as necessary.

Specifying a Trigger/Reject Index

As a part presence sensor detects a new container has entered the verification station, there will likely be some camera delay while the container moves to the center of view of all four (or five) cameras, known as the *trigger delay*. There is also another delay between analyzing the container and actuating a reject mechanism attached to the production line,

also known as the *rejector delay*. See the following diagram:



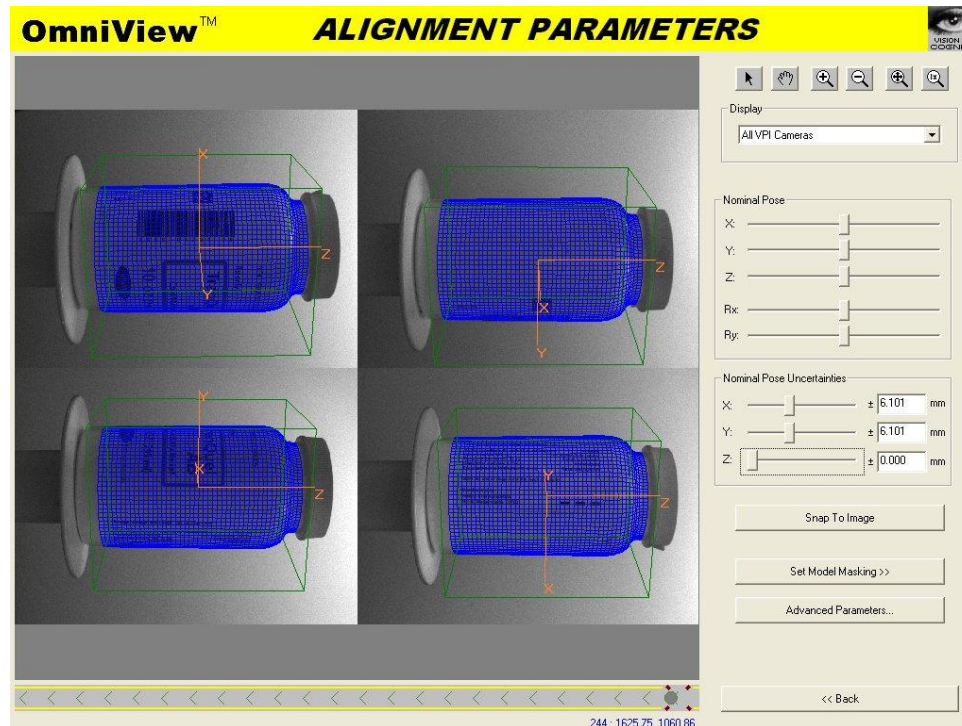
One of the system properties you must configure is the number of encoder pulses that represent the trigger delay and the rejector delay. If you are analyzing containers of different sizes in the same verification station, the values for trigger delay and rejector delay will vary between containers. OmniView allows you to enter pairs of values for trigger delay and rejector delay, with each pair of values matching a specific container size. You can enter up to eight different pairs of values in a single OmniView job.

Use the **Trigger/Reject Data Index** on the Shape Training Panel to specify an index of 1 through 16 to represent which pair of values match the container currently displayed. If all the containers you are analyzing are of the same size, keep the default value of 1.

Adjust Alignment Parameters

After you have trained the shape using the Shape Training Panel, go back to the Supervisor Panel and click **Set Alignment Parameters** to set the *nominal pose*, which is the expected position of each container in a calibrated 3D coordinate system. In some development and production environments, the actual position of each container can be slightly different than the position shown during shape training, and you need to provide a suitable range of uncertainties.

The following figure shows the Alignment Parameters panel where you set the *Nominal Pose* and the *Nominal Pose Uncertainties*:



The panel uses a wire grid to represent the 3-dimensional aspect of the container at the location where OmniView expects it to be in your production environment. The panel also uses a green outline of the 3-dimensional space indicating the area within the verification station that the object might appear from one inspection to the next.

Use the **Nominal Pose** sliders to alter the location and rotation of the wire grid representing the object, relocating the grid as necessary for the expected location of the container within your verification station as OmniView executes:

- Use X, Y, Z to set expected position of container in a three-dimensional physical coordinate space system that is roughly the center of the station, as shown by orange line segments
- Use Rx and Ry to adjust the orientation of the expected position
- Use **Snap to Image** at any time to have OmniView align the wire grid to the image of the container if this viewset is representative of where the containers are typically positioned when they pass through the system.

Use the **Nominal Pose Uncertainties** sliders to define a 3-dimensional space wherein the container can vary from one inspection to the next. You should set the Nominal Pose to be the most representative of the part position, and limit the Nominal Pose Uncertainty as small as practical, since using a large Nominal Pose Uncertainty can add computational and memory overhead. Disable the uncertainty for the Z axis for containers that are perfectly cylindrical and there is no change in contour information at the top or bottom.

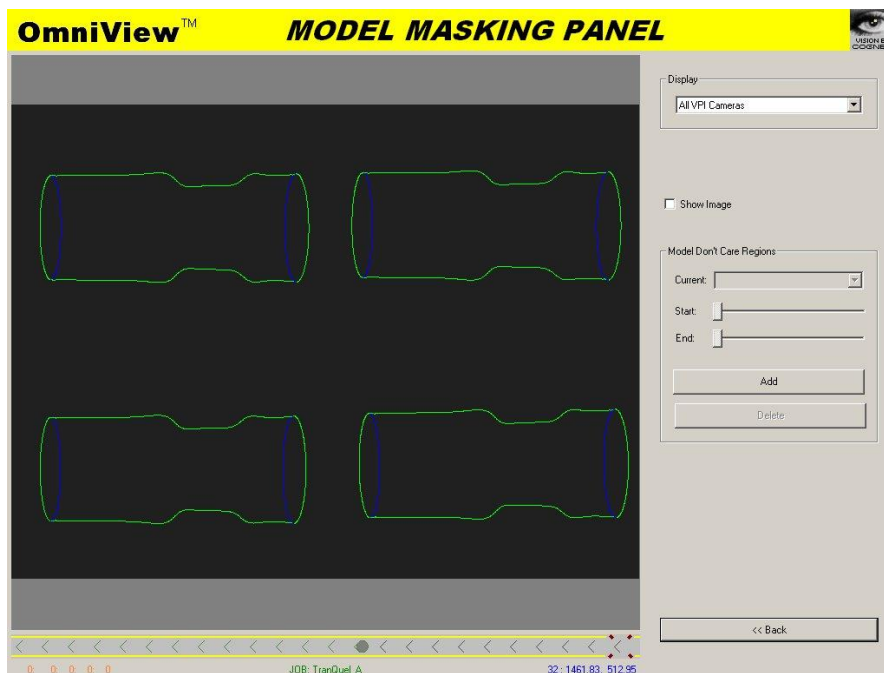
Once the range has been set, the wireframe graphics should match each container that enters the verification station. If you are testing your application and find the graphics do not appear over the container, you need to adjust your **Nominal Pose Uncertainties** and provide a larger range.

Use a Model Mask

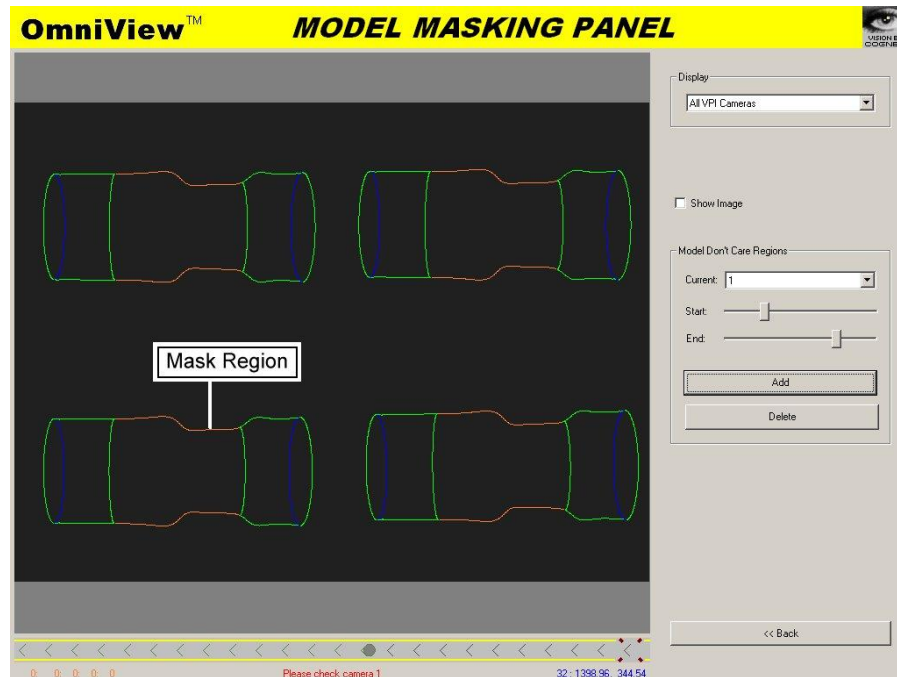
Your container may exhibit a physical trait that can make unwrapping difficult from one inspection to the next. For example, a portion in the middle of the container can have a different diameter than portions above and below it. Since this feature can appear unreliable in the raw 2D images from the side-view cameras, it can be masked out of consideration for each container entering the verification station.

Any mask regions you add will not affect the unwrapped image OmniView will generate for each container. The mask regions affect only the portions of the shape model that OmniView uses to locate each container.

Click **Set Model Masking** from the Alignment Parameters panel to add one or more masks. Clear the **Show Image** checkbox to remove the image of the container and just use the outline, as shown in the following figure:

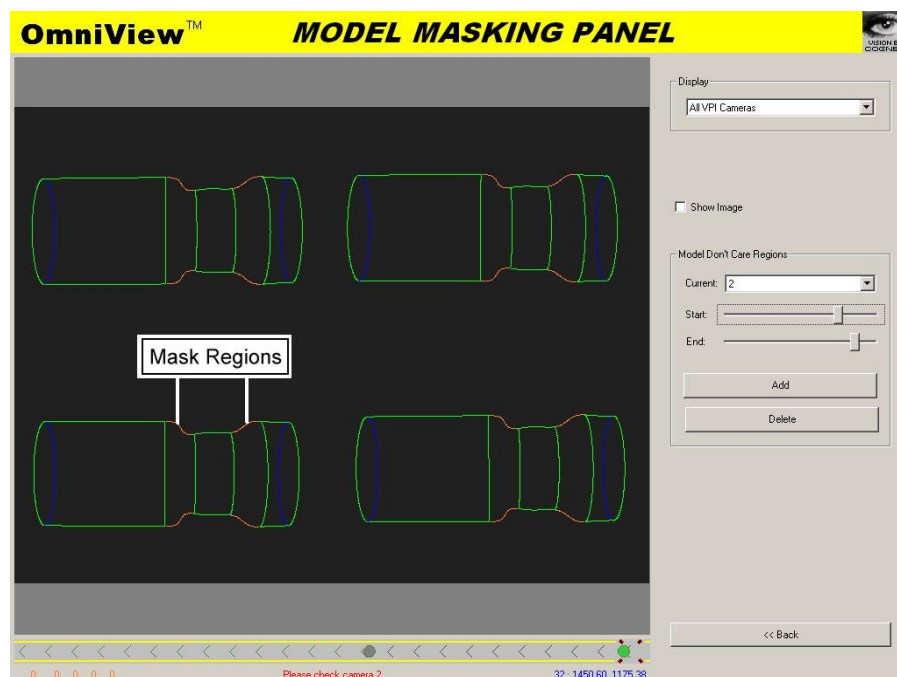


Click **Add** to add a single mask region to the image. OmniView represents a mask region in orange, as shown:



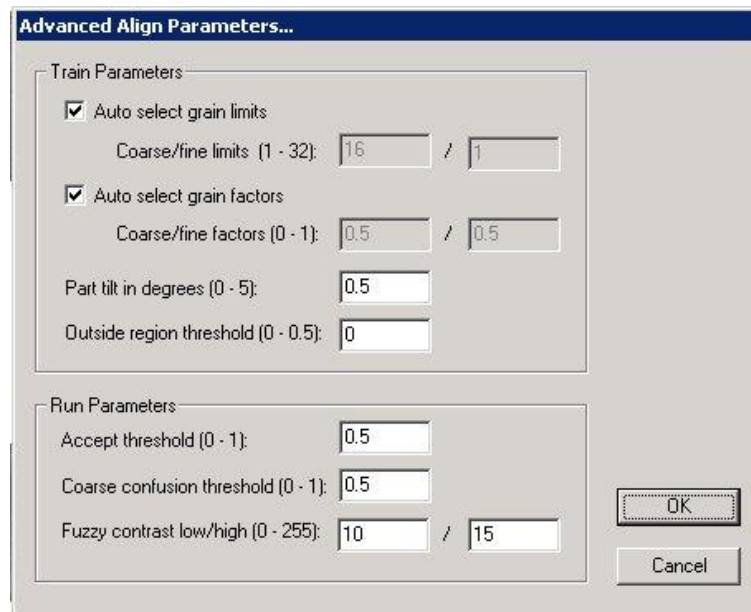
Use the **Start** and **End** sliders to graphically indicate where the mask region begins and ends, respectively. The graphics on the container will update dynamically to reflect the current settings.

You can add more than one mask region as necessary for your container. For example, the following figure shows the same container having two mask regions to make out the sharp curvature of the container:



Using Advanced Parameters

Click **Advanced Parameters** to access a set of advanced alignment parameters:



The image shows a dialog box titled "Advanced Align Parameters...". It is divided into two main sections: "Train Parameters" and "Run Parameters".

Train Parameters:

- ☒ Auto select grain limits
Coarse/fine limits (1 - 32): /
- ☒ Auto select grain factors
Coarse/fine factors (0 - 1): /
- Part tilt in degrees (0 - 5):
- Outside region threshold (0 - 0.5):

Run Parameters:

- Accept threshold (0 - 1):
- Coarse confusion threshold (0 - 1):
- Fuzzy contrast low/high (0 - 255): /

At the bottom right of the dialog are "OK" and "Cancel" buttons.

The default parameters work for most applications, but you may need to alter these parameters based on your application.

Train Parameters

Grain limits specify the range of feature sizes that the tool will use to locate the container, and are specified in model coordinate system units. You can clear the **Auto select** checkboxes and specify your own values for coarse and fine grain limits. In some applications, for containers that are small in comparison with the field of view, you might be able to improve alignment robustness by reducing the coarse grain limit, although this will cause the tool to run more slowly.

Grain factors, meanwhile, control the sampling rate used to encode model features during training. You can clear the **Auto select** checkboxes and reduce this sampling rate to increase performance.

By default, OmniView expects each container to be accurately aligned with the positive Z-axis, but you can specify a larger **Part tilt** as necessary for your verification station.

Finally you can change the value of **Outside region threshold** to specify a fraction of features in each container that can be outside the cameras' fields of view. By default the allowable percentage is 0, but increasing this value allows OmniView to still locate the container even when some percentage of it is not visible.

Run Parameters

You can modify particular run parameters that affect the algorithm OmniView uses to locate the contours of each container:

- **Accept threshold:** Lowering the threshold can allow OmniView to successfully locate the contours in certain applications.

- **Coarse confusion threshold:** Specify a higher confusion threshold to indicate that the scene contains features that resemble the container but are not valid instances of the container, or use a lower value to indicate that only the features in the scene that resemble the container are actually the container itself. You should always set the confusion threshold greater than or equal to the acceptance threshold.
- **Fuzzy contrast low/high:** OmniView uses a weighting function to control the weight assigned to features in images based on their contrast.

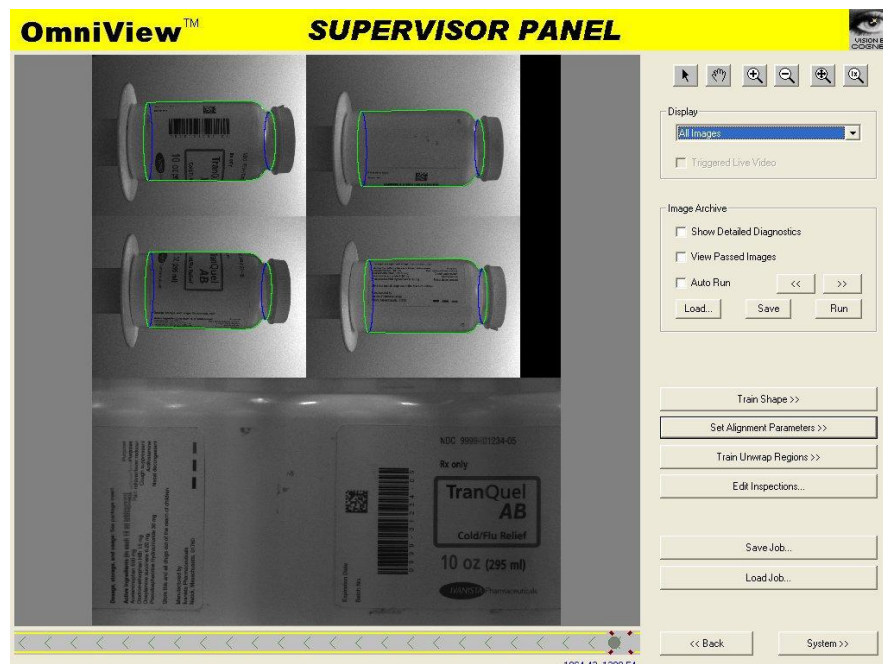
Unwrapping the Image

As you select which vision tools you want to use to analyze each unwrapped image, you must specify the area of interest for each tool. For example, if you want to read a barcode symbol on a label, you must specify where in the unwrapped image the barcode symbol will appear. The smaller the size of the area of interest, the faster any particular vision tool can execute.

Because containers entering the verification station can have a random orientation around the Z-axis, there is no way to predict where the areas of interest are in each unwrapped label. A barcode symbol, for example, might appear anywhere within the unwrapped image, perhaps even split across the beginning and end of the image that OmniView generates. This random orientation can prevent any vision tool from examining the correct area of the image.

To ensure that each vision tool examines the appropriate area of interest in each unwrapped image, you must train your application with the starting point of each unwrapped image. This allows OmniView to generate unwrapped images based on the same starting point, and allows you to know where to locate the vision tools you want to add later.

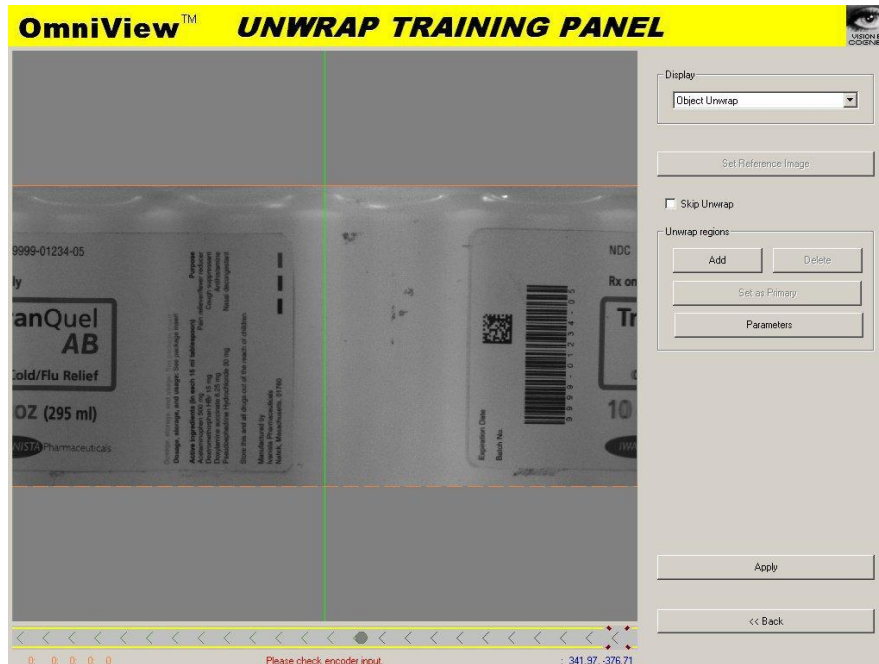
After successfully training the shape of the containers you want to analyze, the Supervisor Panel will display the images of the container from each camera as well as the unwrapped image of the surface of the container with an arbitrary starting point, as shown in the following example:



Perform the following steps to train your OmniView application with the desired unwrap regions and starting location for each unwrapped image:

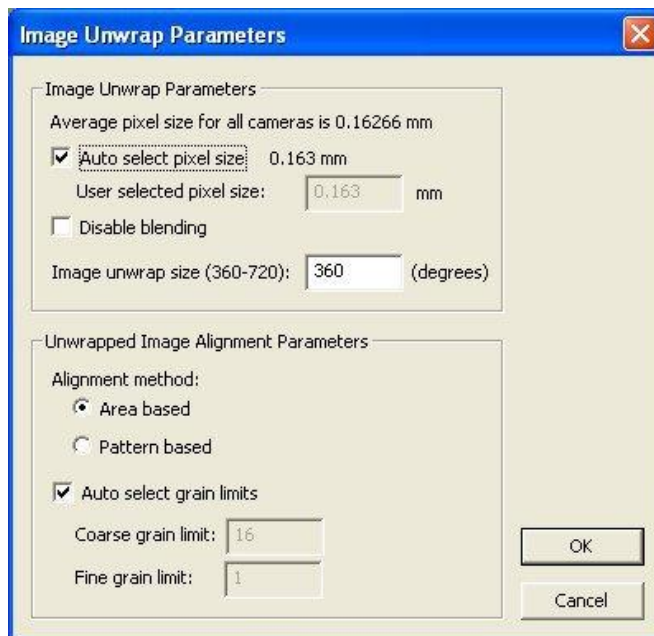
1. Click **Train Unwrap Regions**.

OmniView switches to the Unwrap Training Panel, as shown in the following example:



The horizontal graphics indicate the top and bottom of the unwrapped image. By default, the unwrapped image contains all the image information originally contained within the shape contour graphics you specified during shape training. The panel also contains one vertical graphic that defines the starting point of the unwrapped image.

2. (Optional) Click **Parameters** and access the **Image Unwrap Parameters**:



Cognex recommends using the default options as you test and develop your job. If necessary, use the **Image Unwrap Parameters** to change the area size of the container represented by each pixel. Also, by default OmniView will blend the overlapping pixel data from adjacent cameras to create the output image. Because of the effect of residual calibration error, part variation, and lighting effects, features may not be precisely aligned in the adjacent camera images. The unwrapped image it generates can result in features that appear blurred or doubled. Check **Disable blending** to allow OmniView to concatenate images from adjacent cameras.

By default, OmniView unwraps the image to generate a 360 degree view of the container under inspection. For some applications, a 360 degree view might not generate a view sufficient for your inspection needs. For example, you may need to analyze multiple features in an image where one of the features could appear split between the end of the start of the image, regardless of where you designate the start of the unwrapping. In this type of application, you can specify an unwrap size of greater than 360 degrees to ensure that all the features you need to analyze appear as necessary.

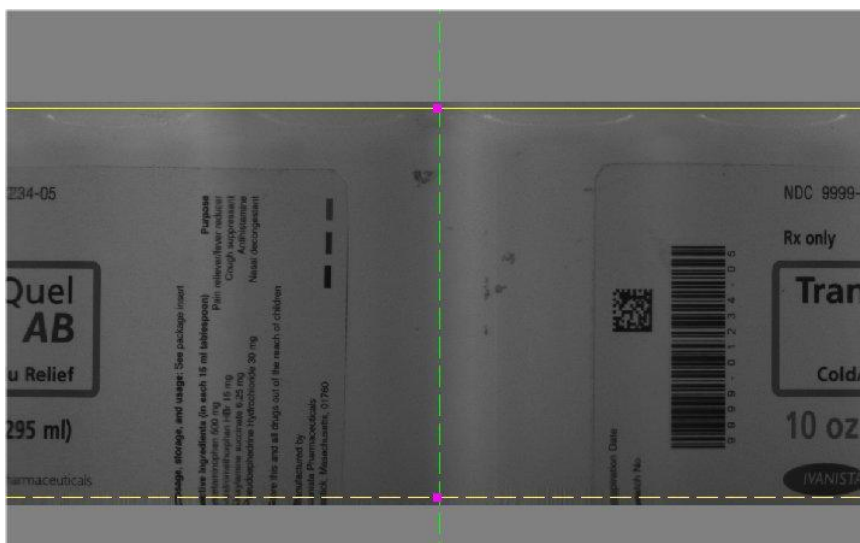
Use the **Image unwrap size** parameter to specify the size, in degrees, of the image you want to generate for your vision tools to analyze. By default the value is **360**, but you can specify any value greater than this up to a maximum of **720**.

Use the **Unwrapped Image Alignment Parameters** to enhance how OmniView performs the unwrapping of each image. The **Area based** option is the most robust option and is suitable for most labels rich with graphics. The **Pattern based** option is suitable for labels with non-repeating patterns.

Alignment of the unwrapped image can be adjusted using the **Auto select grain limits** option. The default setting should be acceptable for most applications, but you can disable the option and provide your own settings for coarse and fine grain limits.

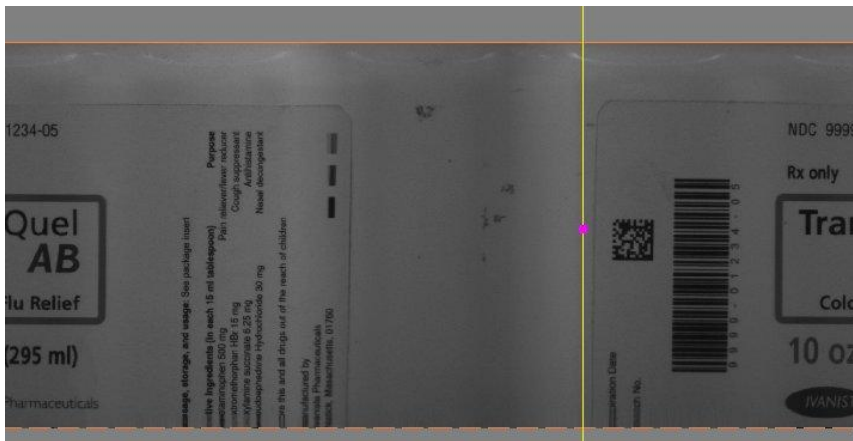
3. (Optional) Select the top or bottom horizontal graphics and relocate them as desired for your vision application.

Moving the graphics closer together generates a slimmer unwrapped image, which can make verification faster. Click on either graphic to highlight them and display their handles, as shown:

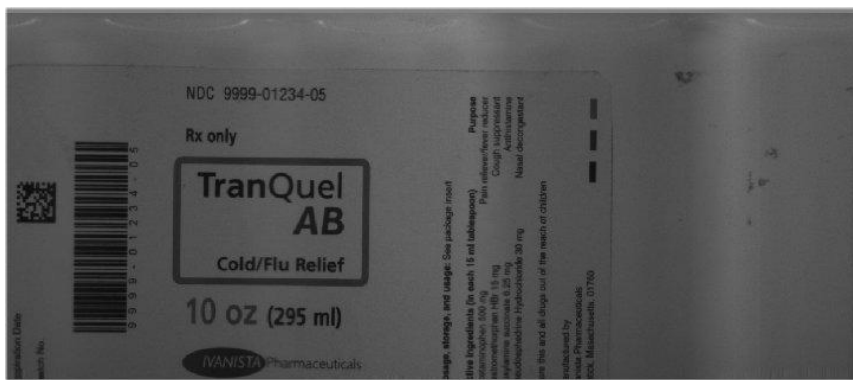


- Click and drag the vertical graphic to the location where you want each unwrapped region to begin before any analysis by the vision tools you add later.

For example, the following figure shows the graphic placed on an unwrapped image:



- Click **Apply** to have OmniView generate a new unwrapped image based on these horizontal borders and the vertical starting **point**, as shown in the following figure:



- (Optional) Add sub-align regions as necessary (see section 3.11 for more information).
- Click **Back** to return to the Supervisor Panel.
- Acquire additional images of the container you want OmniView to analyze to verify that the unwrapped image it generates appears with the features located in the same area each time.

Once you have confirmed that OmniView generates successfully unwrapped images, you can move on to training vision tools.

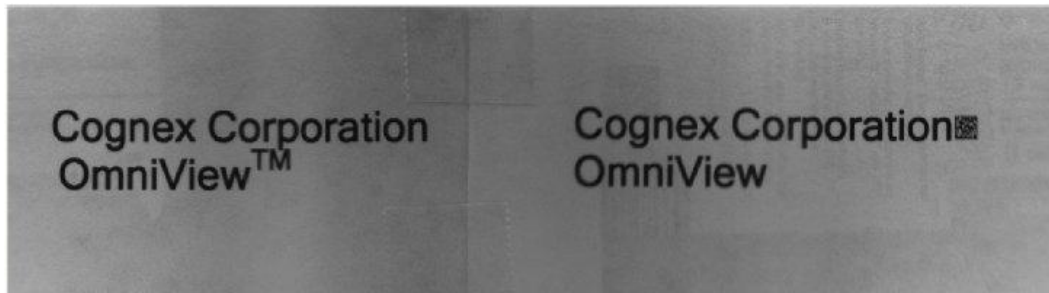
Labels with lumps, texture, or rough surface features will be unwrapped at noticeably lower fidelity than labels with smooth surfaces, even if the container itself is sufficiently rigid. This will adversely affect the ability for vision tools to run on the image.

The consistency of the unwrapping process is typically +/- 5 pixels, depending on label content and surface smoothness. Since there may be a slight horizontal shift from unwrap to unwrap, users are advised to account for this variability by adding a few more pixels to the width of the regions when placing and sizing vision tools on the unwrapped label image.

Adding Sub-align Regions

Many labels can be reliably unwrapped using only the single vertical graphic designating where you want the unwrapping to begin. Some labels, however, have features that repeat or are similar enough to other features on the container that the unwrapping process might not always generate an unwrapped image at the expected starting point. This can cause vision tools to fail later because they are analyzing the wrong area.

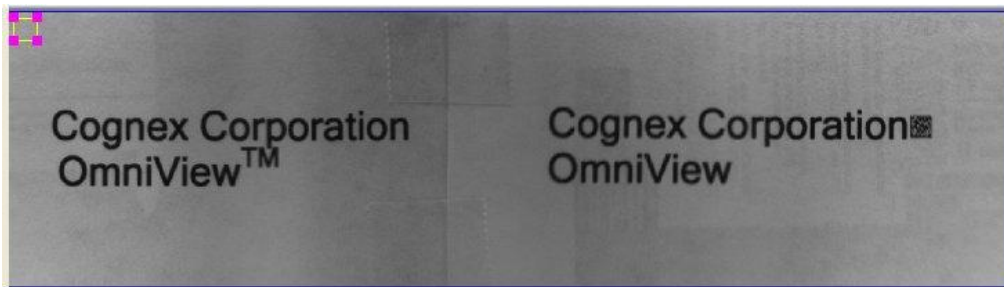
For example, the following figure shows a label with a repeating pattern and few other distinguishing features:



For these types of labels, OmniView allows you to add *sub-align regions*, which can increase the reliability of generating the unwrapped image you expect. To use a sub-align region, perform the following steps:

1. Return to the Unwrap Training Panel.
2. Select **Region Unwrap** from the **Display** pull down menu.
3. Click **Add** under **Sub-align Regions**.

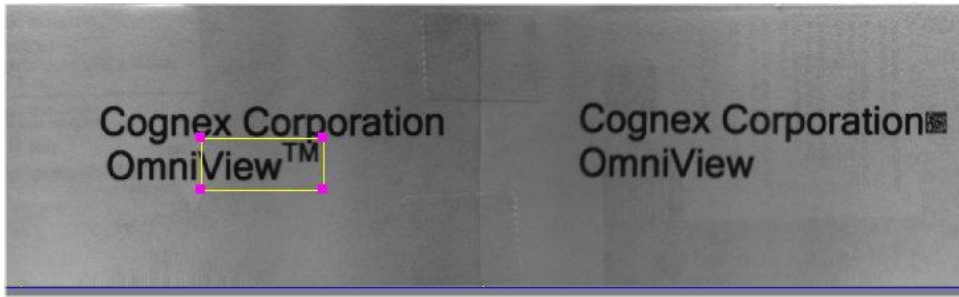
OmniView displays a 360 degree view of the container enclosed in a blue rectangle, with a graphic for the sub-align region in the upper-left corner, as shown:



Any sub-align region graphics must be limited to this blue rectangle.

4. Select the graphic and move it to a unique feature on the label. Larger sub-align regions work more reliably than smaller sub-align regions. Use the resizing handles to

surround a suitable feature, as shown in the following example:



5. Click **Apply**.

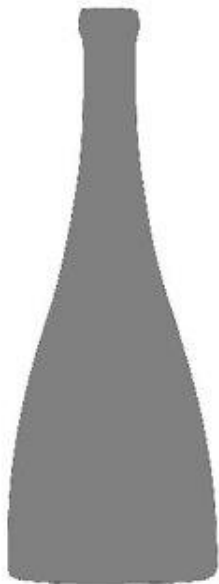
Choose **Region Unwrap** from the **Display** pull down list to see the unwrapped image OmniView now generates, which may appear exactly as before. The sub-align region you added, however, will ensure that the same unwrapped image is generated each time.

Add as many sub-align regions as your application requires.

Using Multiple Unwrap Regions

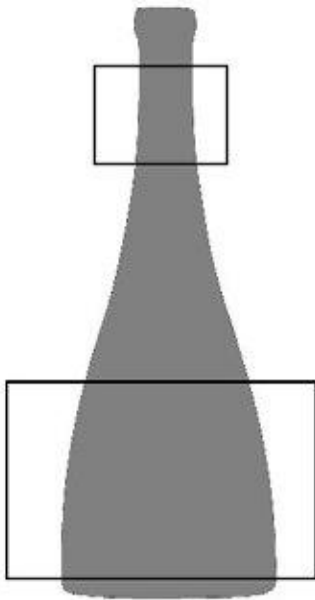
As you perform unwrap training so that each unwrapped image has a consistent look, you have the option of adding additional unwrap regions. Click **Add** from the Unwrap Training Panel to add additional unwrap regions.

Adding additional unwrap regions can be useful for curved containers where OmniView needs to inspect separate areas that have a different diameter and where a single unwrapped image can introduce distortion. For example, the following silhouette represents a bottle where different sections of the bottle have a different diameter:



Generating a single unwrapped image from this type of bottle would mean the features closer to the top would be artificially stretched compared to the features along the bottom. Inspecting this type of container might be handled best by using multiple unwrap regions,

as illustrated in the next figure:



If you use multiple unwrap regions, you should select one and click **Set as Primary**, which sets the selected region as the primary region and is used to calculate the start of the unwrapping algorithm. Any sub-align region you find necessary to add can only be added to the primary region.

Changing the Reference Image

The unwrapped image OmniView generates is the *reference image* you will use to train the vision tools you add to the job. In some production environments, an OmniView verification station will inspect several different containers all having the same shape.

If you have performed shape training and vision tool training on one type of container and need to create another job for a container of the same shape, you do not have to perform shape training again for the new container. You can replace the reference image OmniView uses to perform unwrapping by performing the following steps:

1. Save your current job by clicking **Save Job** on the Supervisor Panel.
2. Capture an image of the new container as described in the section *Image Acquisition* on page 16.
3. Obtain an unwrapped image of the new container by running it through the system. Adjust the Nominal Pose if the representative position of the part has changed.
4. Click **Train Unwrap Regions** from the Supervisor Panel.
5. Click **Set Reference Image** to replace the current unwrapped image with an unwrapped image of the new container.
6. Use the new reference image to perform unwrap training and vision tool training for the new container.
7. Save the job with a different filename.

Vision Tool Training

After calibrating your OmniView application as described in the section *Calibration* on page 20 and performing shape training as described in the section *Shape Training* on page 28, you can begin to specify which vision tools will analyze either the unwrapped image generated by OmniView or the raw images acquired by the cameras in your verification station.

OmniView uses *OmniBuild*, a Cognex VisionPro application, to organize and manage the data flow between the vision tools you use. OmniBuild allows you to take advantage of the following features of VisionPro:

- The wide array of vision tools that can analyze an image for any number of goals, including identification, searching, pattern matching, and measurement
- Custom vision tools you create
- A image-conversion tool for converting color images you acquire to the grayscale images that many vision tools require for analysis.

For information on converting color images to grayscale, see the section *Analyzing Color Images* on page 52.

- Control of image-analysis through scripting, either C# or VB.NET

This guide does not contain specific information about the capabilities and use of the VisionPro vision tools. Instead, you can refer to the VisionPro documentation set installed along with OmniView. Choose **Programs->Cognex->VisionPro-> Documentation->English->VisionPro Online Documentation** to view documentation for all the VisionPro tools.

Accessing the Cognex OmniBuild Window

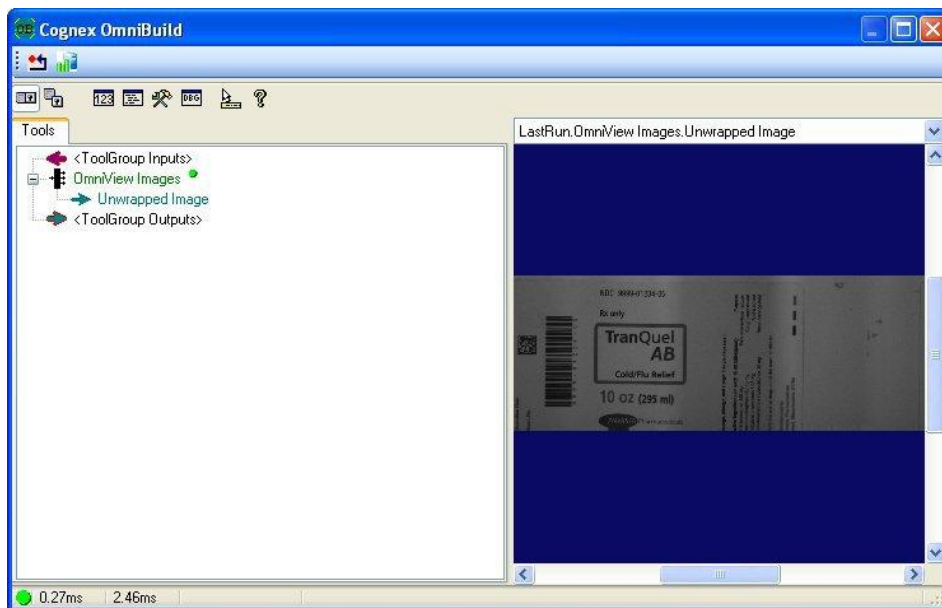
You can add and configure vision tools to your OmniView application once an image of the containers you want to analyze has been successfully trained. The following figure shows a set of images where a container has been unwrapped and the blended image it generates

Vision Tool Training

is ready for vision tool training:



Click **Edit Inspections** on the Supervisor Panel to access *OmniBuild*, the OmniView interface that allows you to add and configure any vision tool from the VisionPro tool set, as shown in the following figure:



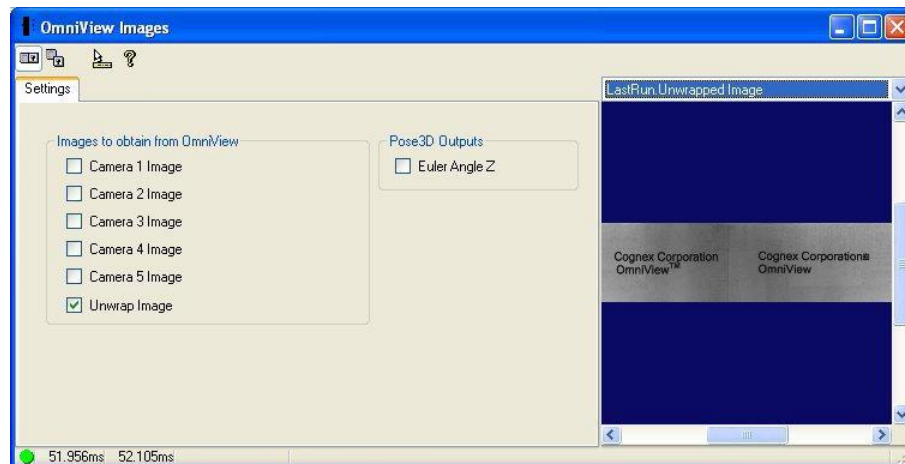
The image window on the right displays the unwrapped image generated from the images acquired by the four cameras. OmniView uses the interface on the left to display the vision tools you have selected to analyze each unwrapped image.

Click the reset button in the top-left of the OmniBuild window at any time to reset the window to its default settings.

Choosing Additional Image Information

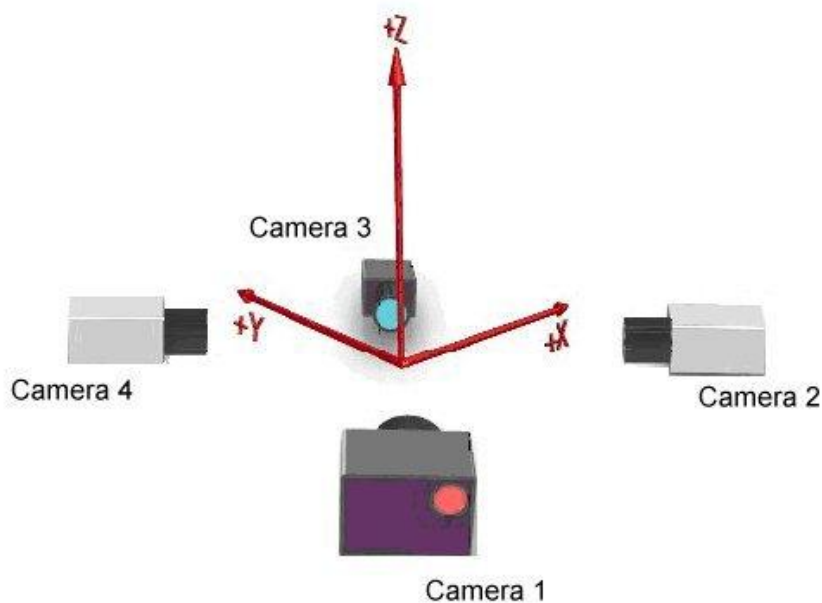
By default, OmniBuild analyzes the unwrapped image with the vision tools you add to the application. You have the option of using the raw images from each side-view camera or the 5th overhead camera. In addition, you can access the change in z-rotation of the container under inspection as compared to the container used during shape training and vision tool training.

Double-click the **OmniView Images** tool to display the following dialog box:



The left side allows you to make any one of the raw images available to the application, while the right side allows you to expose an output terminal for the Euler Angle Z rotation, in degrees.

OmniView represents the pose of the container in a world-coordinate system that is determined by camera calibration. In the world-coordinate system, the x-axis points from Camera 1 to Camera 2 and the z-axis is the average of the cross product of camera positions, as shown in the following figure:

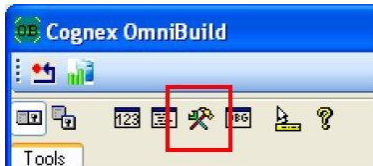


And Euler angles represent the rotation about any axis within the three-dimensional space of the verification station.

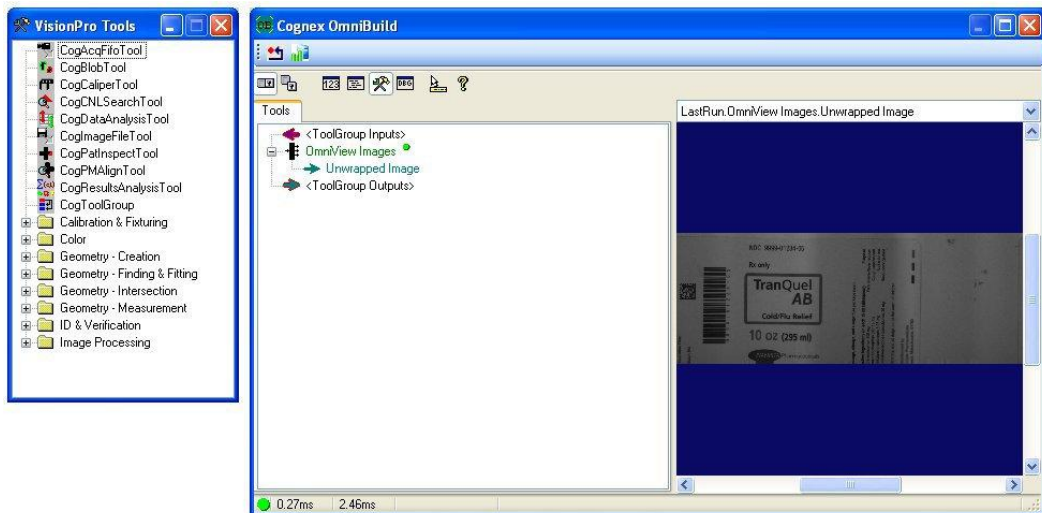
Adding Vision Tools

To add and configure a vision tool, perform the following steps:

1. Click the Show ToolBox button, highlighted in the following figure, to open the list of VisionPro vision tools:



OmniView displays a list of VisionPro tools besides the Cognex OmniBuild window:

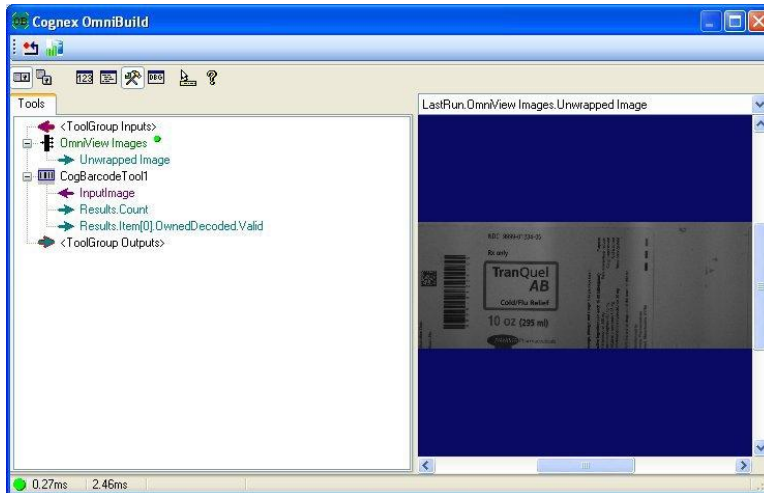


Note: The first VisionPro tool listed is the **CogAcqFifoTool**, but you never need to add one to OmniBuild since it already contains a tool for **OmniView Images**.

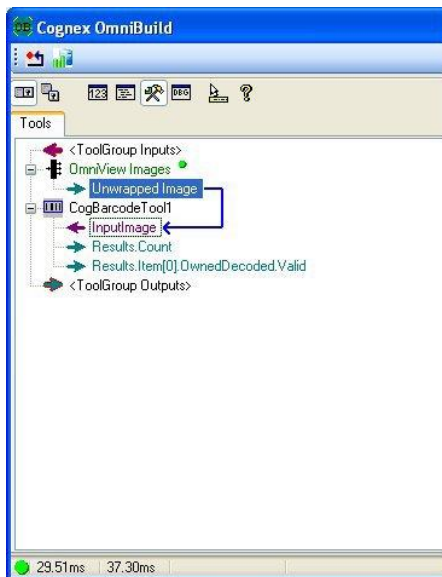
2. Navigate through the tool box and double click on the desired vision tool.

Vision Tool Training

For example, the following figure shows the OmniBuild window when a Barcode tool has been chosen:



3. Select **Unwrapped Image** and link it to the **InputImage** of the vision tool you added, as shown:

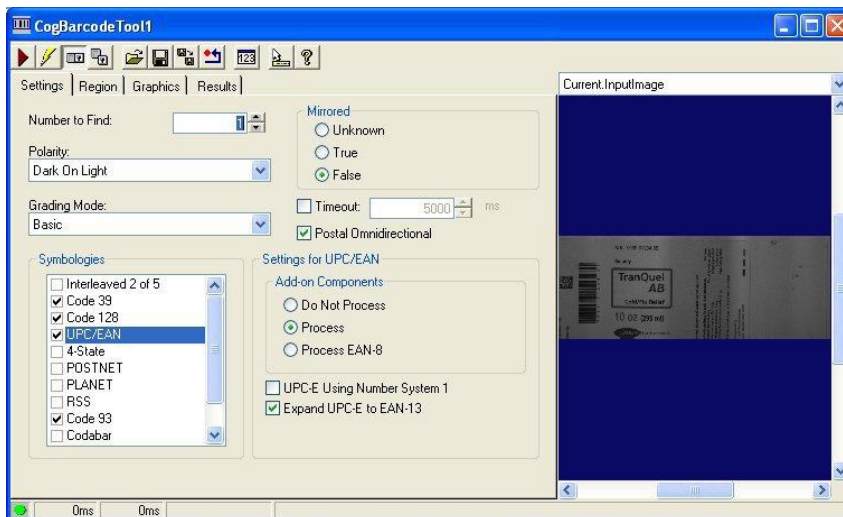


This provides the vision tool you added with the unwrapped image. Now you can train the vision tool with the area of interest you want to analyze and any configuration parameters you need to specify.

4. Double-click on the new vision tool you added to open its edit control.

Vision Tool Training

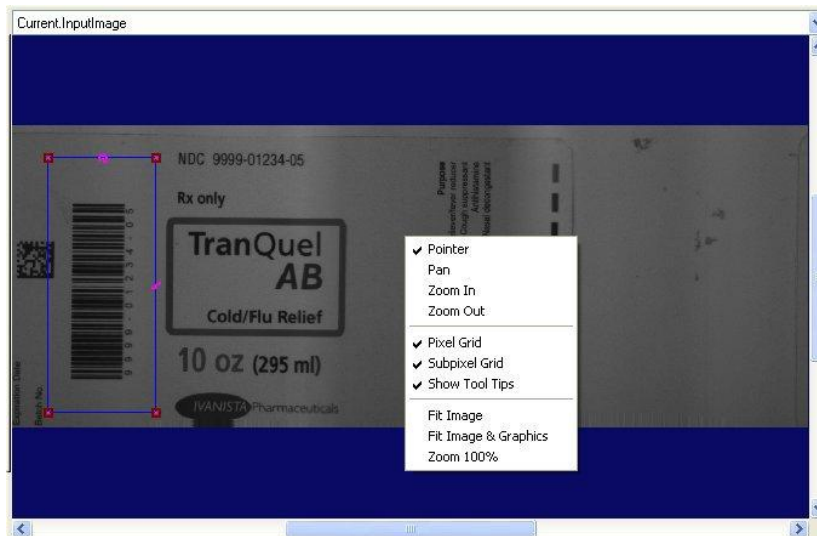
For example, the following figure shows the edit control for the Barcode tool:



For information on any particular feature of the vision tool, click the "?" button along the top. This opens the VisionPro online documentation set to the specific topic on the interface for that vision tool.

5. Configure the vision tool as necessary for your application.

The edit controls can be resized as necessary to view the image information relevant to the vision tool you need to configure. In addition, you can right-click in the image window and select a number of options that allow you to zoom in or out and pan the image as necessary, as shown:



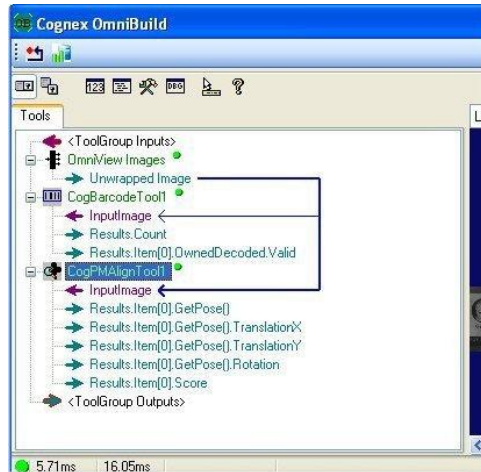
6. Run your vision tool on the current image by clicking **Run** on the Supervisor Panel to ensure it generates the expected results.
7. Test the vision tool on successive images to ensure it analyzes the correct area of the image each time and that it generates the correct results.

Cognex recommends you test the vision tool on known good and bad images (live images or from an image-database file) to know that the vision tool will not generate false accepts or false rejects.

To run the vision tool on the latest acquired image rather than the current reference image acquired earlier, click **Run** in the Supervisor Panel.

8. Add the other vision tools necessary for your vision task.

Your application might require multiple vision tools to analyze each input image. In this case, the OmniBuild window might appear similar to the following example, where both a Barcode tool and a PMAlign tool analyze the same image:



If you need to insert a custom tool to OmniBuild tool group, the corresponding dll needs to be copied to `%OMNIVIEW_ROOT%\bin`. Alternatively, you can build the custom tool assembly as strong named and install it into the Global Assembly Cache (GAC). In addition, OmniView includes sample scripting code for advanced use of VisionPro scripting to change inspection tasks.

Analyzing Color Images

Your application might acquire color images, where each image is represented by three 8-bit planes of RGB (red, green, blue) or HSI (hue, saturation, intensity) values. VisionPro supports a number of color vision tools that can analyze the unwrapped color image your application generates:

- The Color Extractor tool extracts pixels from a color image in order to create a grey-scale image of light and dark pixels based on the reference colors you specify.
- The Color Match tool compares a region of color in a run-time image against a table of reference colors, and generates a set of scores to indicate how closely the area of the run-time image matches each known color.
- The Color Segmenter tool analyzes a color image in order to produce a grey scale image consisting of light pixels against a dark background, where the light pixels correspond to features from the color input that fell into one or more desirable color ranges.
- The Composite Color Match tool examines the color content of an area in an image against a table of reference samples, and generates a set of scores to indicate how closely the area of the run-time image matches each known entry. The higher the comparison score, the greater the similarity.

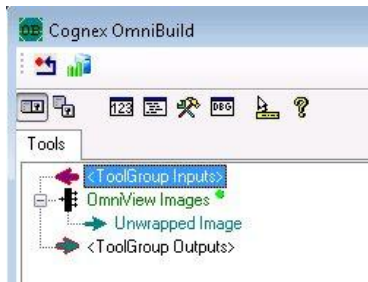
See the VisionPro online documentation for complete details of the color vision tools.

The other VisionPro vision tools require an unwrapped image represented by a single plane of 8-bit grey values, requiring you to convert your color images to grayscale images before they can be analyzed.

Use the Image Convert tool to convert color images to grayscale images. Perform the following steps:

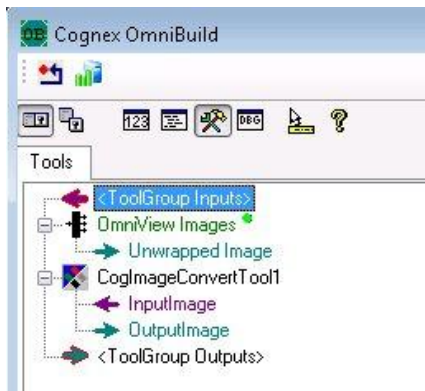
1. Click **Edit Inspections** on the Supervisor Panel to launch OmniBuild.

OmniBuild contains an entry for the unwrapped image:

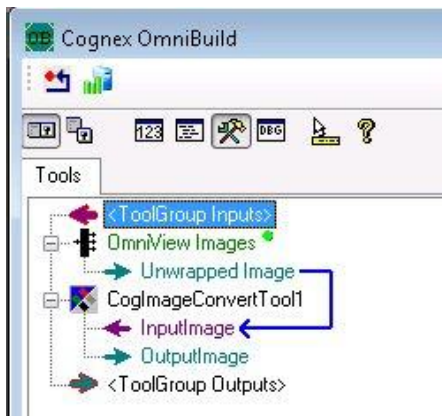


2. Open the Tool Box and select the **CogImageConvertTool** from the **Image Processing** folder.

OmniBuild adds the CogImageConvertTool to the application:



3. Link the **Unwrapped** Image to the **InputImage** of the Image Convert tool:



By default, the Image Convert tool generates an Intensity, or grayscale, output image from any input image. You do not need to change any settings in the Image Convert tool.

4. Link the **OutputImage** of the Image Convert tool to any vision tools you want to analyze the grayscale image.

Enabling Critical Failures

OmniView examines each container for a pass/fail status based on the vision tools you use to examine them. In many production environments, the failure of a vision tool signals that a container should be rejected from the production line.

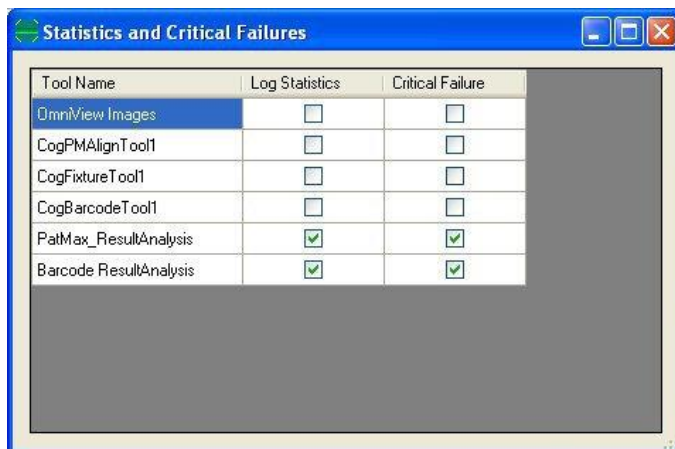
In some instances, the results of a vision tool might indicate that a *critical failure* has occurred on a production line. For example, an application might read the string encoded within a data matrix symbol printed on the label of each container. If the decoded string reveals that the wrong type of container is currently moving through the verification station, this might indicate a need to stop the production line entirely, rather than simply rejecting the containers displaying an unexpected data matrix symbol.

Enabling Critical Failure Conditions

OmniView allows you to configure an application to generate both a failing signal for rejection and a critical failure signal, to be used as necessary for your production environment. To configure your application to generate a critical failure for a particular tool, click the Statistics and Critical Failure icon in the OmniBuild window, highlighted in the following figure:



The following figure displays an example **Statistics and Critical Failures** dialog box:



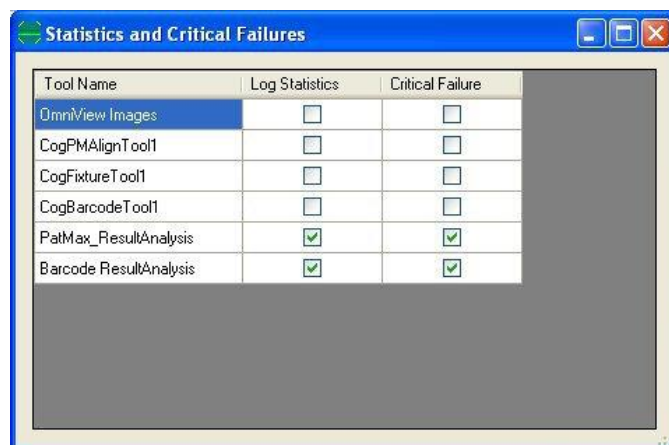
In this figure, two separate vision tools can generate a critical failure signal from the application. The section I/O Configuration, describes how to enable the critical failure output signal for connection to external equipment.

Using a Result Analysis Tool for Critical Failures

Some vision tools do not generate a failed result the same way others do. For example, a Barcode tool does not fail if it cannot locate and decode a barcode symbol in the area of an image you trained it to analyze. Instead, the Barcode tool generates zero results while the tool itself passes.

If you want to generate a critical failure based on the results of certain vision tools, you must use a Results Analysis tool to examine the results the vision tool generates. A Results Analysis tool can examine the results generated by another vision tool and generate a Pass, Warn, or Reject result. For example, you can have a Results Analysis tool examine the results generated by a Barcode tool, and have the Results Analysis tool generate a Reject result if the Barcode tool fails to read any strings or decodes a string that is different from the expected string. Many OmniView applications can benefit from using a Results Analysis tool in determining the result of any particular inspection.

By enabling the Critical Failure option in OmniBuild, OmniView can use the results of a Results Analysis tool to generate a critical failure signal based on the decoded results of ID tool, and not the pass/fail result of the ID tool itself. The following figure shows a **Statistics and Critical Failures** dialog box with the critical failure condition enabled for two Results Analysis tools:



See the VisionPro online documentation for details on how to add and configure a Results Analysis tool in the OmniBuild window.

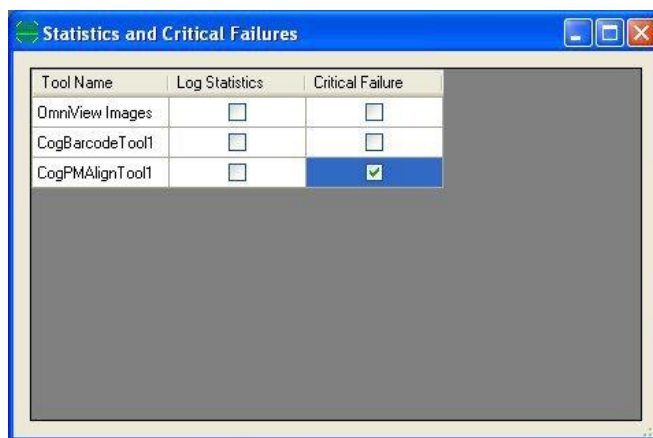
Collecting Statistics on Vision Tools

You can collect and display various statistics regarding the vision tools in your application. Click the Statistics and Critical Failure icon in the OmniBuild window, highlighted in the

following figure:

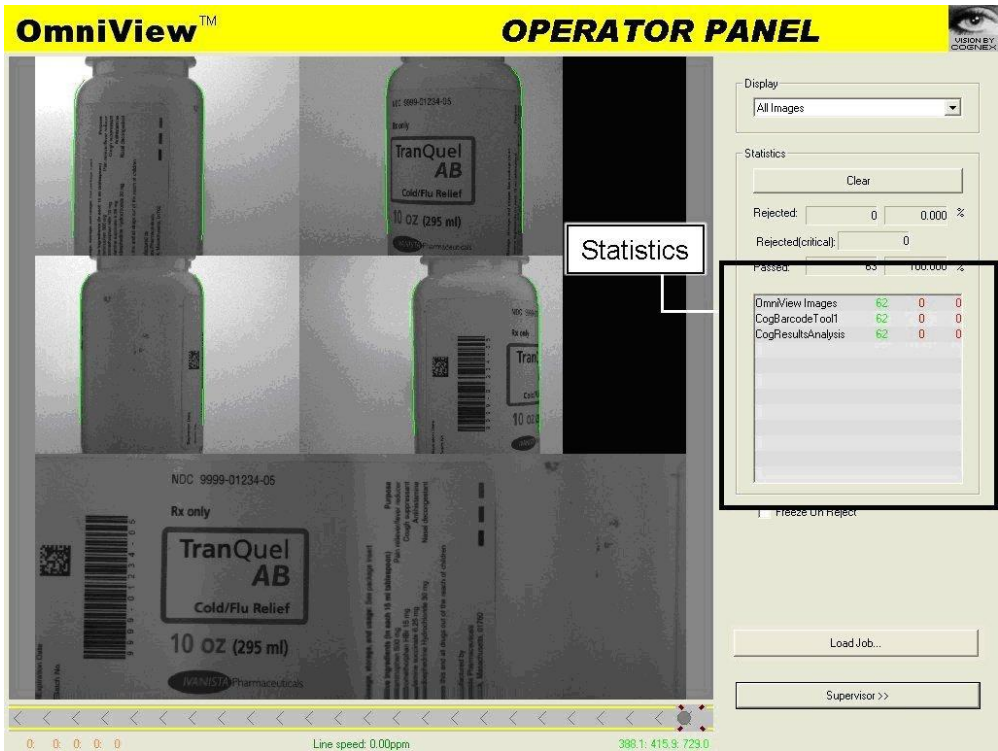


The following figure displays an example **Statistics and Critical Failures** dialog box:



Check the **Log Statistics** checkbox for any tool for which you want to collect statistics. As the application executes, the Operator Panel displays statistics for the number of times

each tool passes or fails, as shown:



Testing the Application

Once you add vision tools to your application, OmniView will display graphics corresponding to the tool results on the corresponding image on which you defined the vision tools. For example, the following figure shows an Operator Panel after the unwrapped image has

been successfully analyzed with a Barcode tool:

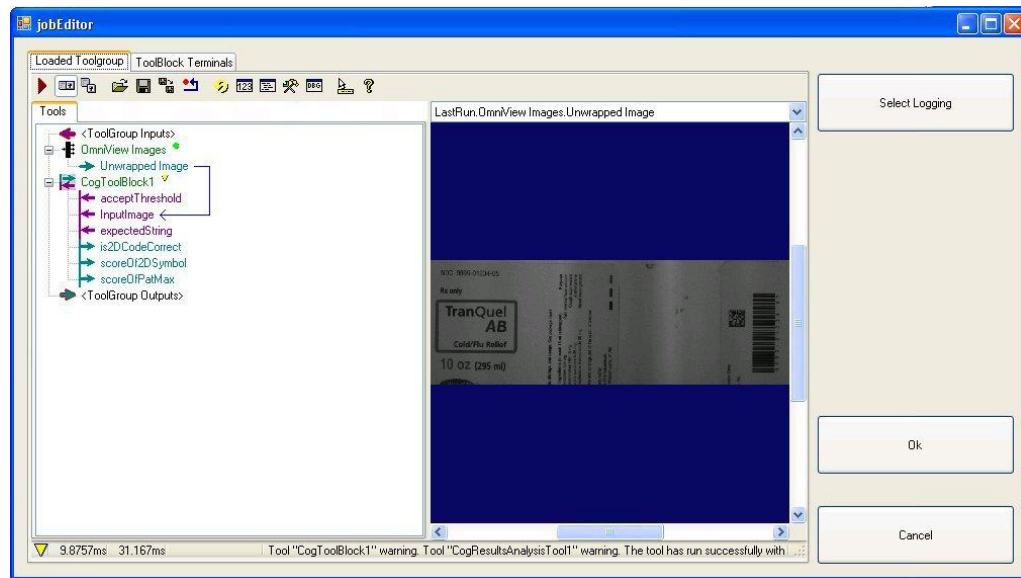


Using a Custom Job Editor

Once your application is complete, you can deploy it to your production environment. In some cases, you may want to limit exposure to the OmniBuild development environment for any changes that have to be made to the configuration of the vision tools by a supervisor or an operator on the factory floor.

OmniView supports a custom job editor you may have developed to act as a simplified interface to the existing vision tools you add and configure using OmniBuild. You design the actual interface the OmniView operator sees when using the custom job editor. The

following figure shows an example job editor interface provided with OmniView:



The custom job editor you create can be configured to expose different levels of the underlying application based on whether the user accesses it through the Operator Panel or the Supervisor Panel. Specifically, OmniView launches the custom job editor with the keyword *Operator* when launched through the Operator Panel and *Supervisor* when launched through the Supervisor Panel. As you define your custom job editor you define what aspects of the job are accessible for each keyword. For example, you may define your custom job editor so that launching it from the Operator Panel prevents the user from modifying the configuration of any vision tools in the job.

Enabling a Custom Job Editor

To use a custom job editor instead of OmniBuild, perform the following steps:

1. Switch to the System Panel.
2. Click **System Properties**.
3. Select the **Misc** tab.
4. Select the option **Use custom job editor** to switch from OmniBuild.

The custom job editor must have the name *jobEditor.exe*. Be aware that if you select **Use custom job editor** but do not select the *jobEditor.exe* you want to use, OmniView will continue to use OmniBuild to add and configure vision tools.

OmniView launches the custom job editor internally using the following command:

```
jobEditor.exe vpp_file_name cdb_file_name mode
```

Where *vpp_file_name* represents the current job, *cdb_file_name* stores the images of the container for which you want to modify the application, and *mode* is either *Operator* or *Supervisor* to reflect the user access to the underlying vision application.

OmniView performs the following tasks anytime the custom job editor is in use:

- Continuously checks to see if the application specified by the variable "custom job editor" is still executing. If so, OmniView ignores new containers entering the verification station.

- Prevents the user from switching to online mode until the application specified by the variable "custom job editor" is closed.

Even when the application uses the custom job editor, OmniView continues to go through OmniBuild for inspection at run time. OmniView continues to pass all information to OmniBuild which runs transparently.

Be aware that the application specified by the custom job editor is required to:

- Take three arguments: the first one is the job being edited; the second one specifies the image cdb to be used, and the third specifies mode (*Operator* or *Supervisor*). The job editor application needs to respect the cdb content as described above. As OmniView interacts with it only via temporary job files, end users can save the job by clicking the **Save Job** button in Supervisor Panel.
- Reference Cognex.OmniView.dll which can be found in %OMNIVIEW_ROOT%/bin. If the application does not reference the dll, it will not be able to load the OmniBuild job file.

A sample Job Editor project is included in the %OMNIVIEW_ROOT%/samples/ASAP/OmniBuildJobEditor/JobEditor directory. For instructions on running the sample code, refer to the file %OMNIVIEW_ROOT%/samples/ASAP/OmniBuildJobEditor/ReadMe.txt.

Launching through the Operator Panel

With a custom job editor enabled and defined, the Operator Panel contains the additional button **Edit Inspections**, which launches the custom job editor with the *Operator* keyword. As you create the custom job editor application, you define which features are accessible from a user accessing the application through the Operator Panel.

Launching through the Supervisor Panel

With a custom job editor enabled and defined, the **Edit Inspections** button will launch the custom job editor rather than OmniBuild. In addition, OmniView launches the custom job editor with the *Supervisor* keyword. As you create the custom job editor application, you define which features are accessible through the Supervisor Panel.

Using Globally Accessible Variables

The globally accessible variables defined in the assembly Cognex.OmniView.DataPool.dll allow you to use thread-safe variables in your OmniView scripts and .NET applications. The Cognex.OmniView.DataPool.dll assembly is located in %OMNIVIEW_ROOT%/bin. The assembly supports a number of classes.

CogToolData

Use the CogToolData() class to collect data about the tools in the current job. The following listing shows the definition of the class in Cognex.OmniView.DataPool.dll:

```
// -----  
// CogToolData  
// -----  
  
public ref class CogToolData{  
public:  
    CogToolData()  
    {  
        mName= nullptr;  
        mNumPassed= 0;  
    }  
}
```

```

        mNumCritical= 0;
        mNumFailed= 0;
    }
    property System::String ^Name
    {
        System::String ^get() {return mName;}
        void set(System::String ^v) {mName= v;}
    }
    property int NumPassed
    {
        int get() {return mNumPassed;}
        void set(int v) {mNumPassed= v;}
    }
    property int NumFailed
    {
        int get() {return mNumFailed;}
        void set(int v) {mNumFailed= v;}
    }
    property int NumCritical
    {
        int get() {return mNumCritical;}
        void set(int v) {mNumCritical= v;}
    }
private:
    System::String ^mName;
    int mNumPassed;
    int mNumFailed;
    int mNumCritical;
};

```

CogJobData

Use the CogJobData() class to collect data about the current job. The following listing defines the interface for CogJobData():

```

// -----
// CogJobData
// -----
public ref class CogJobData{
public:
    void ReadFromDataPool();
    void WriteToDataPool();
    CogJobData()
    {
        mNumPassed= 0;
        mNumFailed= 0;
        mNumCritical= 0;
        mNumTimeout= 0;
        mToolData= nullptr;
    }
    property int NumPassed
    {
        int get() {return mNumPassed;}
        void set(int v) {mNumPassed= v;}
    }
    property int NumFailed

```

```
{
    int get() {return mNumFailed;}
    void set(int v) {mNumFailed= v;}
}
property int NumCritical
{
    int get() {return mNumCritical;}
    void set(int v) {mNumCritical= v;}
}
property int NumTimeout
{
    int get() {return mNumTimeout;}
    void set(int v) {mNumTimeout= v;}
}
property array<CogToolData ^> ^ToolData
{
    array<CogToolData ^> ^get() {return mToolData;}
    void set(array<CogToolData ^> ^v ) {mToolData= v;}
}
private:
int mNumPassed;
int mNumFailed;
int mNumCritical;
int mNumTimeout;
array<CogToolData ^> ^mToolData;
};
```

CogNamedDouble

Use the CogNamedDouble() class to define a thread-safe double value.

```
// -----
// CogNamedDouble
// -----
public ref class CogNamedDouble : public System::IDisposable
{
public:
    CogNamedDouble(System::String ^name, double initValue);
    ~CogNamedDouble()
    {
        this->!CogNamedDouble();
    }
    !CogNamedDouble()
    {
        if (mNamedDouble!=0)
            delete mNamedDouble;
        mNamedDouble= 0;
    }
    double Add(double value);
    property double Value { double get(); void set(double); }

private:
    ccNamedDouble *mNamedDouble;
};
```

CogNamedInt32

Use the CogNamedInt32() class to define a thread-safe integer value.

```
// -----  
// CogNamedInt32  
// -----  
public ref class CogNamedInt32 : public System::IDisposable  
{  
public:  
    CogNamedInt32(System::String ^name, int initValue);  
    ~CogNamedInt32()  
    {  
        this->!CogNamedInt32();  
    }  
    !CogNamedInt32()  
    {  
        if (mNamedInt32!=0)  
            delete mNamedInt32;  
        mNamedInt32= 0;  
    }  
    int Add(int value);  
    property int Value { int get(); void set(int); }  
private:  
    ccNamedInt32 *mNamedInt32;  
};
```

CogNamedString

Use the CogNamedString() class to define a thread-safe string.

```
// -----  
// CogNamedString  
// -----  
public ref class CogNamedString : public System::IDisposable  
{  
public:  
    CogNamedString(System::String ^name, System::String ^initValue);  
    ~CogNamedString()  
    {  
        this->!CogNamedString();  
    }  
    !CogNamedString()  
    {  
        if (mNamedString!=0)  
            delete mNamedString;  
        mNamedString= 0;  
    }  
    property System::String ^Value { System::String ^get(); void  
        set(System::String ^); }  
private:  
    ccNamedString *mNamedString;  
};
```

Global Variable Sample Application

The directory `%OMNIVIEW_ROOT%/samples/ASAP/OmniBuildScripting/jobs` contains a sample Job that uses the `Cognex.OmniView.DataPool.dll` assembly. View the *Readme.txt* file for details on these global variables.

I/O Configuration

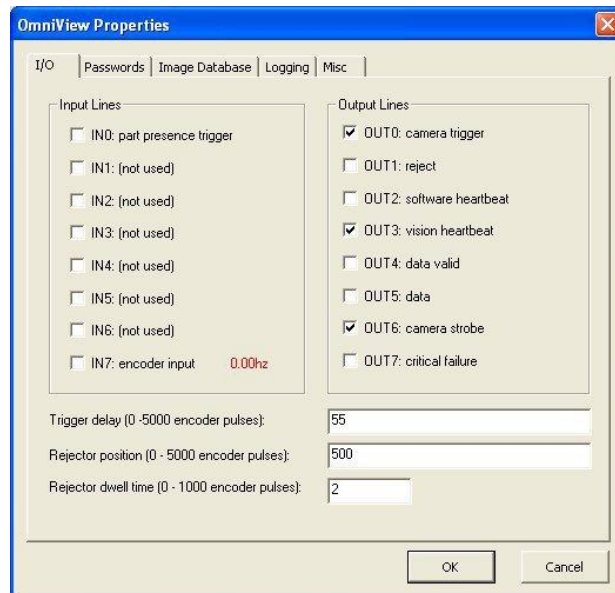
OmniView uses the opto-isolated I/O features of a Cognex MVS-8500 series frame grabber to provide eight input signals and eight output signals for your vision application. Use the I/O signals to configure your part presence trigger, the encoder input signal, and various output signals. The following table summarizes the I/O signals you can configure within OmniView:

Part presence trigger	OmniView accepts this input signal which is high when a container is present in the verification station and is detected by the part presence sensor.
Encoder input	OmniView accepts this input signal which is high for each encoder pulse the application receives.
Camera trigger	OmniView sets this output signal high to trigger all cameras to acquire an image. By default, the falling edge of the signal triggers the cameras, but this behavior can be changed by modifying the configuration file config.xml. See the section <i>The Configuration File</i> on page 71 for details on modifying config.xml.
Reject	OmniView sets this output signal high when one or more vision tools does not generate a successful result.
Software heartbeat	OmniView generates this square wave signal to indicate that the application is running.
Vision heartbeat	OmniView generates this signal to indicate it is online and able to perform verifications.
Data valid	OmniView generates a high signal to indicate that the current value of the Data line is valid.
Data	OmniView generates a high signal to indicate the most recent verification attempt has passed. Similarly, a low signal (combined with a high signal from the Data valid line) indicates that the most recent verification attempt has failed.
Camera strobe	OmniView generates a low-to-high transition to fire any connected strobe lights. This should occur one encoder period after that camera has been triggered.
Critical failure	OmniView can generate a high critical failure signal any time the following conditions occur: • The Results Analysis tool that examines the results generated by a particular vision tool generates a Reject status. • A vision tool that has been enabled to generate a critical failure generates a failing result. • Make sure the conveyor belt is running. In either case (vision tool or Results Analysis tool), the Critical Failure condition in the OmniBuild window must be enabled for that tool. Refer to the section <i>Vision Tool Training</i> on page 46 more information.

	Your verification station may use the Critical failure signal to completely stop a production line as opposed to simply rejecting the object generating this status condition.
Trigger delay	Specify an integer value for the number of encoder pulses OmniView will wait between sensing that a container is present through the part presence sensor and triggering the (encoder pulses) cameras to capture an image. Since the leading edge of any container will trigger the part presence sensor, you want OmniView to wait a number of encoder pulses before acquiring the image. This allows OmniView to capture images of containers when they are at the center of the station. Using a trigger delay of 0 forces OmniView to capture images of the containers when they are not at the center of the station and makes it difficult or impossible for OmniView to capture the proper images necessary to creating a single unwrapped image. You might have to experiment with this setting to ensure that OmniView captures images when the container under inspection is located at the center of your verification station. See for the section Specifying a Trigger Index on page 33 for more details on setting the correct value. When multiple container sizes need to be handled, specify a set of integer values, each corresponds to one container size. In the Shape Training Panel, select the index (1-based) of these trigger delay values corresponding to the trained shape.
Rejector position (encoder pulse)	Specify an integer value for the number of encoder pulses between the center of your verification station and any reject mechanism you have installed in your production environment. The encoder count must exceed the peak verification cycle time of the system. The minimum recommended distance from verification station to reject mechanism is 8 containers, including spacing. The maximum distance from trigger to reject mechanism is 32 containers (the smallest size to be analyzed), including spacing. See the section Specifying a Trigger/Reject Index on page 33 for more details on setting the correct value.
Dwell time (encoder pulse)	The number of encoder pulses you want the reject mechanism to remain active.
Rejector dwell time	The amount of time, measured in encoder pulses, that the rejector mechanism should be enabled.

Configuring I/O Signals

To enable I/O signals for your OmniView application, click **System Properties** from the System Panel and select the **I/O** tab. The I/O tab appears as shown:

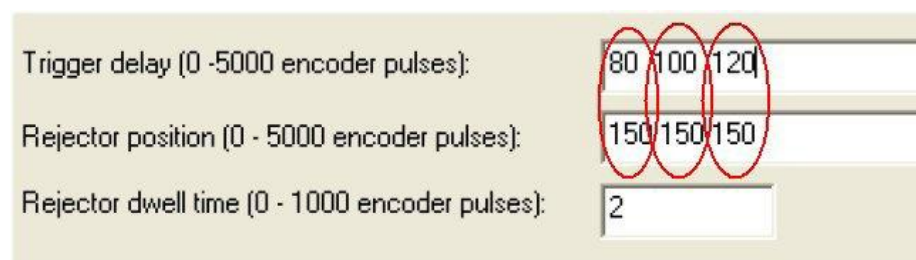


Note: The checkboxes next to the input line signals for **part presence trigger** and **encoder input** may not accurately reflect the incoming signals from connected equipment as OmniView operates. Cognex recommends you use these indicators only to verify that the respective signals are received from your production environment as you connect and test the wiring for your verification station.

If you are using an MVS-8504, some of the output line options appear in different places but the options are the same.

Using Multiple Values for Trigger Delay and Rejector Positions

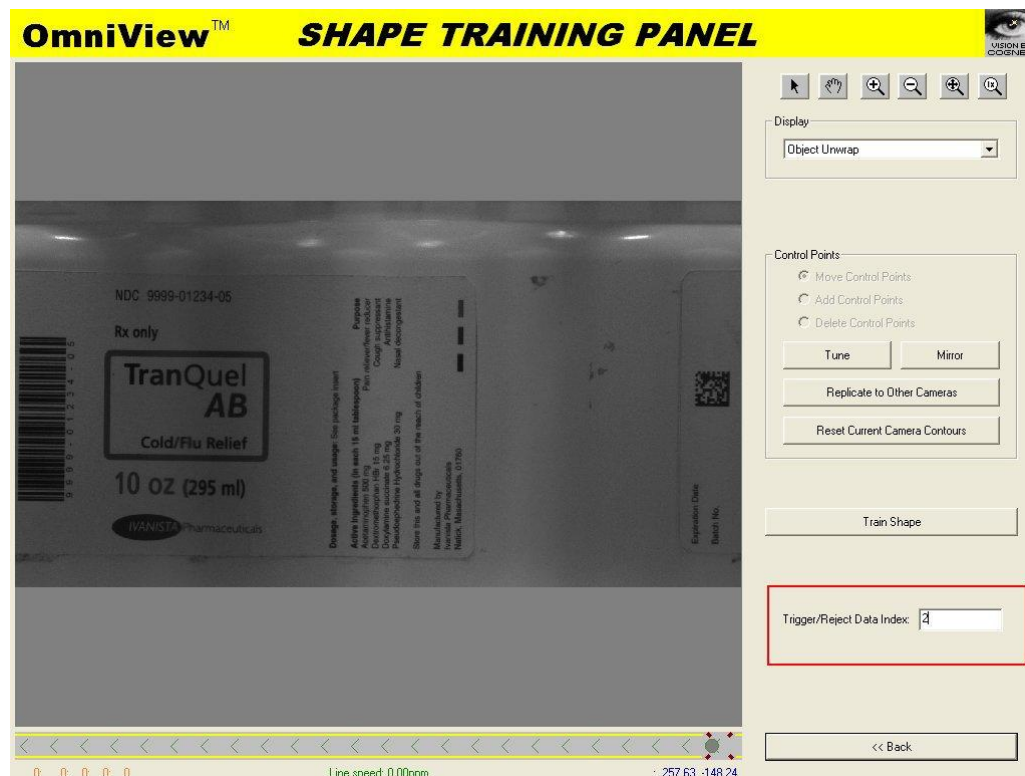
If you are using the same verification station for multiple containers that have different diameters, you can use the **I/O** tab to enter pairs of values for **Trigger Delay** and **Rejector position**, where each pair corresponds to a different container size. For example, the following figure shows the I/O tab configured with three pairs of values:



Specify which pair you want to use in the current application using the **Trigger/Reject Data Index** field of the Shape Training Panel, as highlighted in the following figure which

I/O Configuration

specifies the second pair:



You can specify and select up to 16 different pairs for **Trigger Delay** and **Rejector Position**.

Wiring the I/O Board

For more information on wiring, refer to the I/O wiring diagram, available by choosing **Start->Programs->Cognex->OmniView->Wiring Reference**.

External Control Interface

OmniView allows an external program running on the local machine to control online/offline status, select from (or query) the list of saved jobs on a given system, and query the latest pass/fail statistics.

A C# source file for a .NET API reference to the External Control Interface (ECI) functions and strings can be found at `%OMNIVIEW_ROOT%/samples/asap/eci/ecisample/eci.cs`.

Supported Functions

Refer to the .NET API reference file for the interface describing the functions supported by the External Control Interface:

Function	Purpose
IsReady	Queries the status of the ECI module Your local application must successfully call this function before calling any other ECI functions.
Send	Sends a command string and a unique number identifying the command sequence to OmniView The function has a number of different command strings, described in the next table.
Receive	Receives a response string from OmniView For each OmniView command string, there will be at least one response string returned from OmniView. OmniView will respond with a <InvalidCmd> response string when it cannot recognize a command string.
Query	Gets number of response strings pending in the receiving queue
Flush	Removes all pending response strings from the receiving queue

The .NET API reference file also details the command strings you can send to OmniView through the Send command:

Command	Purpose
dryRunMode	Sets the operational mode for OmniView.
loadJob	Loads a job from hard disk
saveJob	Saves the current job to hard disk
queryJobs	Instructs OmniView to return a list of saved job names to user application
stats	Queries OmniView for the current inspection statistics
dockToTray	Instructs OmniView to hide or show itself
statsTools	Queries OmniView for the current inspection statistics of a single vision tool or all the vision tools used in the

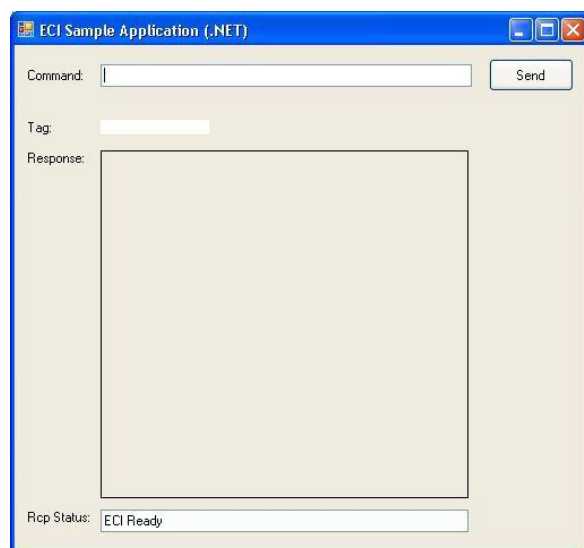
External Control Interface

	inspection.
getToolBlockInputTerminalNames getToolBlockOutputTerminalNames	Queries OmniView for the names and types of all input and output terminals for a particular CogToolBlock within the application.
getToolBlockInputTerminalValue getToolBlockOutputTerminalValue	Queries OmniView for the value of a specific input or output terminal of a CogToolBlock within the application.
setToolBlockInputTerminalValue	Sets the input terminal value for a particular CogToolBlock within the application.

Extending the Support Functionality

You have the option to extend the local external control to full remote control, and thus add the ability to remotely control OmniView from a separate workstation or interface device. The local application serves as an intermediary, which links OmniView and the remote application.

The OmniView installation utility installs the sample code that demonstrates how an application that communicates with OmniView, using ECI functions, can be developed in a .NET development environment. By default, the sample application is located in the directory *C:\Program Files\OmniView\samples\ASAP\ECI\ECISample*. The following figure shows the interface for the sample application:

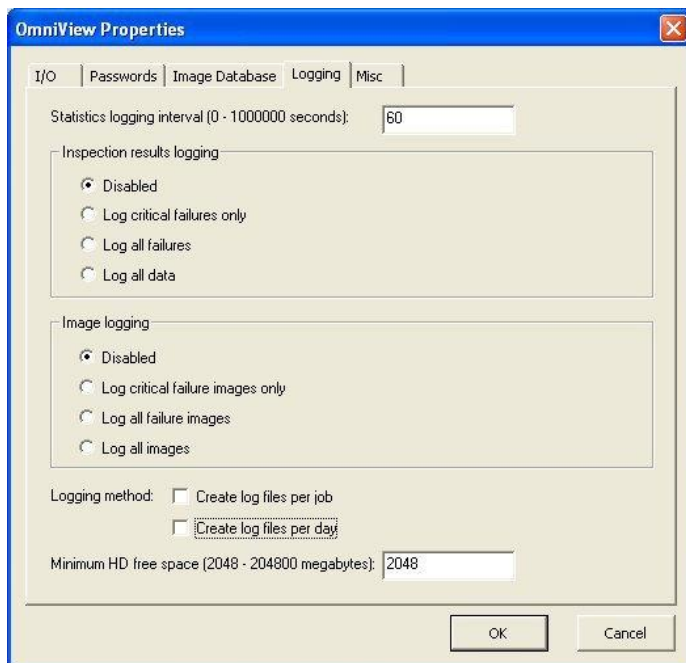


Data Logging

As your OmniView application executes it can record data such as the number of inspections that have passed, failed or generated a critical failure. In addition, OmniView offers a variety of options for storing the unwrapped images it generates. For example, you might want to store the unwrapped images that fail an inspection for later analysis.

Cognex recommends you use data logging only as you test that your application generates the correct results, as using the data logging features of OmniView while your application executes can negatively impact the overall performance of your vision application.

To establish data logging, click **System Properties** from the System Panel and select the **Logging** tab, which appears as shown:



Set the following logging parameters:

- **Interval for statistics logging**
Specify an interval for when OmniView will log the current statistics for passed and failed inspections. The default value of 0 indicates that OmniView will not log these statistics as your application executes.
- **Inspection results logging and image logging**
Choose a condition for logging inspection results to a log file. OmniView records inspection results in a text file with the suffix *.log* and uses the bitmap file format to store images. OmniView stores both text files and images in your *\data\log* directory.
- **Choose a logging method**
OmniView can record statistics against a particular job, log statistics throughout the same day, or both.
If you choose to log information based on the job then OmniView will create a log file with the name *jobname.log* and append all statistics to this log file.

If you choose to log information on a per day basis OmniView will create a log file with the name *yyyymmdd.log* (year, month, and day) and append all statistics to this log file.

- Specify the minimum hard disk space required on the PC to keep logging enabled

By default, OmniView will continue to log statistics so long as the available hard disk space is above 2048 Mbytes. OmniView scans the hard disk periodically for this minimum amount. If the hard disk capacity falls below this minimum, OmniView disables data logging, displays a warning message on the operator interface, and adds an entry to the log file regarding the lack of storage space.

You can change the minimum amount of hard disk space required for logging to remain enabled.

Logging Inspection Results

Choose a condition for logging inspection results to a log file. OmniView records inspection results in a text file with the suffix *.log* and uses the bitmap file format to store images. OmniView stores both text files and images in your *\data\log* directory.

Each line in an inspection results log file can contain inspection results either for a container or overall inspection statistics for the current job, but not both. The first character in the line defines the data type. If the first character is an "S", the line contains overall inspection statistics and uses the following format:

S,mm/dd/yyyy,hh:mm:ss,numPass,numFailure,numCriticalFailure

If the first character is an "R", the line contains inspection results for each container and uses the following format:

R,mm/dd/yyyy,hh:mm:ss,P/F/C,imageFileName.bmp,omniBuildLogMsg1,omniBuildLogMsg2,...

(P = pass, F = fail, C = critical fail)

If you are using image databases to test your application prior to moving it to the production environment, the inspection results for each container replaces the timestamp with the file name and record number:

R,cdb.idb,recNumber,P/F/C,imageFileName.bmp,omniBuildLogMsg1,omniBuildLogMsg2,...

Logging Images

Based on the image logging option you choose, OmniView can save the unwrapped images it generates. OmniView stores image files in the *\log\images* directory with the format *yyyymmdd_hhmmss_x.bmp*, where "x" is an integer starting at 0.

Frequently Asked Questions

Q: What are the keys to successful OmniView lighting?

A: Use a lighting scheme that produces a very bright and very even field of light around the entire container where the background is not saturated.

Q: How do I maximize Depth of Field?

A: More depth of field is required for OmniView as the part diameter increases. To increase depth of field, use as much light as you can generate and set the aperture of the lenses to the highest possible 'f-stop' number that still allows 20 grey values of contrast (or more) between part body and background. Normally this is f6 or above. Make sure the edges of the part are sharply focused without losing focus at the center of the part, even when the part is shifted or tilted by 1/8" or more.

Q: What are the best practices when it comes to shape training in OmniView?

A: Include any available 'curvy' feature on the profile of the container. This will make OmniView's default object location method work best.

Q: I just ran through the calibration and trained a part, but the system won't locate it when I run. What steps should I take?

A: Check to see if there are any diagnostic messages. Make sure the final pose is also correct according to the procedure.

Q: What techniques can be used to avoid triggering problems in noisy industrial environments?

A: Connect the system's electrical service entrance to the output of a battery backup system or Uninterruptible Power Supply (UPS). Use heavier gauge wiring for the DC commons, and ensure that all ground lines are terminated to a single physical location, usually in the control enclosure. We recommend the use of shielded twisted pair cable, where the shield (or 'drain') wire is only connected at the terminal block. The 'drain' wire runs the length of the cable, but is not connected at the far end, such as a trigger input. Use compression-crimped connections in the terminal block to avoid loose or intermittent wiring. If noise is still found to be a problem on camera or light trigger lines, consider using a capacitor in parallel with the trigger output line in order to form a simple low-pass filter.

Q: What should I do if I see red wireframe around a part in Operator Panel?

A: OmniView displays results with high-enough score in green wireframe, results with lower score in red wireframe.

When this happens, you can:

- Confirm that system is still calibrated by using Validation feature in System Panel -> Calibrate Cameras. Recalibrate cameras if you are prompted to.
- Check that part images have the best contrast you can achieve.
- Check the alignment parameters in Supervisor Panel -> Set Alignment Parameters, make sure the Pose

Uncertainties are large enough to cover the part movement, and the Accept Threshold in Advanced Parameters is set reasonably.

The Configuration File

OmniView stores various configuration information in the file *config.xml*, stored by default in %OMNIVIEW_ROOT%\data. Both the OmniView Setup Wizard and the OmniView application will access the variables stored in *config.xml* at various times during initial setup and each time OmniView launches.

Depending on your role in developing an OmniView vision application, you might want to access the variables in *config.xml* directly. In general, operators in a production environment should never need to modify *config.xml*, which is intended for system integrators as they ready a verification station for deployment.

The following table describes variables and their default values that cannot be modified through the use of the Setup Wizard or the OmniView interface:

Variable	Default	Description
IOTriggerPolarity	1	An active-low state triggers the photo sensor to begin an image acquisition.
IORejectPolarity	1	An active-high state trigger the rejector mechanism
IODefaultOutputState	0	A container fails an inspection if it reaches the reject mechanism and no status as to its pass/fail status has been determined by the vision tools inspecting it.
TrgBlockedFilterSize	0	When OmniView detects that part presence sensor stays blocked for this many encoder pulses, the reject line will be set to high until presence sensor is cleared. The default value 0 disables this feature. If the part presence sensor is already blocked when you start OmniView, the reject line will not be set to high. To test that this feature is operating normally: 1. Pass 10 containers through the production line, or use some other method to trigger the part presence sensor roughly 10 times. 2. Block the part presence sensor for an encoder count at least equal to the value of RejectorPosition or TrgBlockedFilterSize. 3. Confirm that the reject line has been set to high until the part presence sensor is unblocked. In addition, ensure that the value of IOTriggerPolarity matches the requirements of the part presence sensor. If blocking the sensor means high, then this variable should be set

The Configuration File

to 1.		
CcfFormat		Change this string value if you are using cameras other than those supplied with OmniView. For the correct string value to use, search the VisionPro Online Documentation for "camera configuration file" to find a list of supported cameras and the strings to describe the video format for each
CameraTriggerPolarity	1	The falling edge (0) of the signal triggers the camera rather than the rising edge (1).
CameraStrobePolarity	1	The falling edge (0) of the signal triggers strobe light rather than the rising edge (1). Refer to your OmniView Release Notes for details on how to update this variable if you are migrating from an earlier release.
ExposureTime	0.000250	The exposure time for each camera, in seconds.
UseOmniBuild	1	Use OmniBuild to add and configure the vision tools that will analyze the container under inspection.
OmniBuildJobEditor		Blank by default, but you must edit this variable and specify a custom application if you plan to use the Job Editor instead of OmniBuild to provide access to the vision tools you use for this job.
UseContrastBrightnessFromJob	0	Use the settings for Contrast and Brightness saved in this configuration file rather than using the settings saved as part of any job that you load. If the value is set to 1, then OmniView will use the settings saved as part of the job. Saving particular settings for Contrast and Brightness for any given job can be necessary when a single verification station can run multiple jobs that inspect containers of various colors and graphics and some containers appear darker than others.
AnimateUnwrap	0	Set this value to 1 to display a 3D animation of the object under inspection (for demo purposes only). Dedicating a second monitor to the OmniView Model Viewer is recommended for best visual effect.
ConveyorDirection	0	Determines which direction the graphic representing the conveyor belt on the Operator Panel will move (default is right to left).

The Configuration File

The file *config.xml* also contains other variables which are modified by the Setup Wizard or the OmniView interface. You should not need to modify these variables directly:

Variable	Default	Description
JobName	The name of this job	
EnableMultiThread	1	Enables multi-core processing
BayerConversionMethod	2	Reduces the size of 5MP color images to three RGB images of 1.25 MP
EnableSideCameraWhiteBalance	1	Enable white balance for side-view cameras
Enable5thCameraWhiteBalance	1	Enable white balance for 5 th camera
WhiteBalanceFactors	2.010000 1.000000 2.370000 1.580000 1.000000 1.800000 1.700000 1.000000 2.090000 1.650000 1.000000 2.010000 1.680000 1.000000 2.160000	
TriggerDelay	The delay between when the part presence sensor detects a container has entered the verification station and when an image acquisition is triggered	
RejectorPosition	The delay between when a container has moved from the center of the verification station and is at the appropriate place for any rejector mechanism	
RejectorPulseWidth	The amount of time, measured in encoder pulses, that the rejector mechanism should be enabled	
NumCameras	5	The number of cameras for this application
NumVPICameras	4	The number of cameras you are using to capture images from the side of the container
Units	1	0 = inches; 1 = millimeters
Pitch	6.350000 6.350000	Size of the squares of the calibration plate
SupervisorPassword	Password to access the Supervisor Panel	
EngineerPassword	Password to access the System Panel	
LogOnlineImageLogLevel	0	Type of images to record when logging (disabled by default)
LogOnlineResultLogLevel	0	Type of logging to perform (disabled by default)
LogOnlinePerDay	0	Create log files per day
LogOnlinePerjob	0	Create log files per job
LogOnlineInterval	0.00000	Logging interval, measured in seconds
MinHDCapacity	2048	Minimum free hard disk space, in megabytes, before logging stops
ImageTestFile	The .scp file you are using to test OmniView using a	

The Configuration File

	set of image-database files
ImageTestSpeed	How often the images in the .scp file are cycled through OmniView
ImageArchiveDir	The directory for storing archive images
ImageArchiveTime	How often to perform image archiving
<DCAMSerialNumbers>	The serial numbers of the connected GigE Vision cameras
<ContrastBrightness>	Eight values that list the current settings for Contrast and Brightness to use for each image acquisition.

Other variables contained within *config.xml* but not listed here are for internal use only and should never be modified.