AdeptSight 2.0 Online Tutorials
March 2007

AdeptSight Tutorials
AdeptSight Tutorials help you learn how to use AdeptSight by walking you through setting up and running some basic, functional, vision/robot applications.

Welcome to AdeptSight Tutorials
We recommend that you start with the Getting Started with AdeptSight tutorial, to familiarize yourself with the AdeptSight environment.

Getting Started with AdeptSight
This tutorial explains how to build a basic vision application. This tutorial requires a PC with AdeptSight, and a camera. You do not need to connect to a controller to complete this tutorial.

AdeptSight Pick-and-Place Tutorial
This tutorial explains how to create a basic pick and place application. It requires an Adept Cobra robot, connected to a CX controller or an i-Series Cobra with AIB. Familiarity with V+/MicroV+ is recommended, but not necessary.

AdeptSight Conveyor Tracking Tutorial
This tutorial requires an CX Controller and conveyor belt. It explains how to create a basic conveyor-tracking application. Knowledge of V+ is recommended.

AdeptSight Upward-Looking Camera Tutorial
This tutorial explains how to create an application for an upwards-facing camera. This tutorial applies to a system where the robot pick objects and brings each object into the field of view of an upward-facing came. Each part is then placed by the robot in a location called the 'Place Location'.

AdeptSight Standalone C# Tutorial
This tutorial explains how to create a vision application in Visual Studio, using the AdeptSight Framework. It also provides an introduction to some often-used vision tools. Familiarity with the Visual Studio environment and with the C# programming language is recommended.
Getting Started with AdeptSight

This tutorial explains the basics of the AdeptSight software and guides you through the creation of a simple vision application.

Tutorial Overview

This tutorial does not require being connected to a controller and robot. You can also follow this tutorial without a camera, by importing images such as those provided in the Tutorials folder that is in the AdeptSight installation folder.

- Installing AdeptSight Hardware
- Installing the Software
- Starting AdeptSight
- Adjusting the Camera
- Calibrating the Camera
- Creating a Vision Sequence
- Adding Tools to the Vision Sequence
- Acquiring Images
- Adding a Locator Tool
- Creating a Model
- Editing the Model
- Configuring Search Parameters
- Running and Testing the Locator
- Adding a Frame-Based Vision Tool
- Configuring Frame-Based Positioning of a Vision Tool
- Completing the Vision Application

Additional Tutorials

- AdeptSight Pick-and-Place Tutorial
- AdeptSight Conveyor Tracking Tutorial
- AdeptSight Standalone C# Tutorial

System Requirements

- PC running Windows 2000 SP4 or Windows XP
- PC with an OHCI-compliant 1394 (FireWire) Bus Controller
- Windows .NET Framework. If it is not present on your computer, it will be installed during the software installation.
The type of PC processor will influence the execution speed of the vision applications.
Installing AdeptSight Hardware

Installing the Robot

This tutorial does not require that you be connected to a controller and a robot. However this is a requirement for the other AdeptSight Tutorials. Refer to installation instructions in the manuals that came with the robot and controller for details on installing the robot.

Figure 2 illustrates a setup for a robotic vision guidance application with AdeptSight.

![Figure 2 Overview of a Robotic Setup with AdeptSight](image)

Installing the Lens

1. Locate the lens that came with the package.
2. Install the lens on the camera. Do not over-tighten.
3. Do not set the lock screws, since you will need to adjust the lens aperture and focus later.

Installing the Camera

1. Mount the camera in the workcell using the camera mount brackets.
2. Locate the IEEE 1394 (FireWire) cable that is included in the shipment box.
3. Connect one end of the cable to the 1394 port on the camera. See Figure 2.
4. Connect the other end of the cable to a 1394 FireWire port on the PC. A hub may be required if the PC (laptop) has a 4-pin port.
5. Typically, you should mount the camera so that it is perpendicular to the workspace.
6. Make sure that the installed camera clears the top of the robot and the entire work envelop.
In Windows 2000 you may get a "Found New Hardware" popup. In such a case, click **Cancel**: the Basler camera driver will be installed with the AdeptSight software.

**Figure 4** Basler Camera connection ports

**Next:** [Installing the Software](#)
Installing the Software

Before Installing

- Install the USB hardware key (dongle) that came with AdeptSight. This dongle is required and must be present at all times to ensure the proper functioning of AdeptSight.
- Uninstall any previous Adept DeskTop versions that are on the PC.
- Uninstall any previous AdeptSight versions.
- Uninstall any existing HexSight versions

Installing the Software

1. Launch the installation from the AdeptSight CD-ROM
2. Follow the instructions of the installation program.
3. The installation program will install the correct Adept DeskTop version that is required for AdeptSight.
4. The installation will install and/or update:
   - The driver for the Safenet Sentinel USB hardware key (dongle)
   - The Basler camera driver (BCAM 1394 Driver)
   - Microsoft .NET Framework 2.0

Next: Starting AdeptSight
Starting AdeptSight

Vision applications are created and managed in the **Vision Project** window in AdeptSight.

This tutorial explains how to open and use AdeptSight within Adept DeskTop. To follow this tutorial you can also open AdeptSight in one of the example programs provided in the support files that were installed with AdeptSight.

**To start AdeptSight from Adept DeskTop:**

1. Open Adept DeskTop.

2. From the Adept DeskTop menu, select **View > AdeptSight**, or click the 'Open AdeptSight’ icon in the Adept DeskTop toolbar.

3. If you have more than one controller license on your system, the **Controller Information** dialog opens. Select the type of controller you will use.

![Figure 5 Select Controller](image)

4. The Vision Project window opens, similar to **Figure 6**.

   Vision applications are built and configured through the Vision Project window, also called the Vision Project **manager**.

![Figure 6 The Vision Manager Window](image)

**Using the Vision Project Window**

A vision project consists of one or more vision sequences, which are managed and run from the **Sequence Manager** section of **Vision Project** interface.

From the **Sequence Manager** you open the Sequence Editor to add and configure vision tools. This is covered later in this tutorial in the section **Creating a Vision Sequence**.
From the **System Devices Manager** you add, manage and configure the camera, controllers, robots and conveyor belts needed for your application. This is explained in other tutorials: [AdeptSight Pick-and-Place Tutorial](#) and [AdeptSight Conveyor Tracking Tutorial](#).

Before creating a new vision application, you will have to adjust the camera, set up the devices used by the application, and calibrate the system.

**Next:** [Adjusting the Camera](#)
Adjusting the Camera

The **Camera** tab in the **System Devices** manager should display the cameras detected on the system. To adjust camera focus and contrast, you will need to open a live display of the images being grabbed by the camera.

**Displaying Live Camera Images**

1. Select the **Cameras** tab in the **System Devices** manager, as shown in [Figure 7](#).
2. In the list, select the Basler Camera (A601F...).
3. Click the 'Live Display' icon:
4. The Live Display window opens. Use this display to adjust the camera position, as well as focus and aperture, as explained below.

**Adjusting Lens Focus and Aperture**

When the Live Display window is open, you can use it to guide you in adjusting the lens aperture and focus.

**To adjust camera focus and aperture:**

1. Place one or more objects in the field of view.
2. If needed, use zoom options by right-clicking in the display window. See [Figure 8](#).
3. Adjust the focus until objects in the display are sharp.
4. Once you have obtained the best possible focus, adjust the lens aperture (f-stop) until you obtain a well-contrasted image. If it is too highly contrasted (saturated) you will lose detail.
You can now optionally adjust camera parameters, although the default camera parameters should be satisfactory for this tutorial.

Next:  Adjusting Camera Properties (optional) or skip to Calibrating the Camera
Adjusting Camera Properties (optional)

If you want to adjust camera parameters, follow the steps below. Otherwise go to the next module.

Opening the Camera Properties Window

1. Select the Basler camera in the list of **Available Cameras** (A601F...).
2. Click on the 'Camera Properties' icon. See **Figure 9**.

![Figure 9 Opening the Camera Properties](image)

Configuring the Camera Properties

In the Camera Properties window:

1. Select the **Stream Format** tab and set the following properties:
   - **Format**: select **Format 0**.
   - **Frame Rate**: select **60 fps**.
   - **Mode**: select **640 x 480, Mono 8**
2. Select the **Video Format** tab and set the following properties by moving the sliders or directly typing in the values:
   - **Shutter**: set to 600.
   - **Gain**: set to 10.
   - **Brightness**: set to 400.
3. Leave other parameters at their default settings, and click **OK** to close the camera properties window.
Getting Started with AdeptSight - Adjusting Camera Properties (optional)

Figure 10 Camera Properties Window

Next: Calibrating the Camera
Calibrating the Camera

Calibrating the camera/vision system increases the accuracy of your results by correcting image errors.

- The camera calibration requires a grid of dots target. For the purpose of this tutorial, you can print out one of the sample calibration targets that is provided in the Tutorials/Calibration folder, installed in the AdeptSight program folder.

- Sample targets are intended for teaching purposes only. Targets printed on paper are not accurate calibration targets. See Why is Vision Calibration Important? for information on creating accurate dot targets and the importance of calibrating the vision system.

- You can skip this step, in which case the camera will be calibrated during the Vision-to-Robot calibration. This is explained in the next module of this tutorial. However, calibrating the camera separately, with a grid of dots target, can provide higher accuracy to your application than the camera calibration that is done through the Vision-to-Robot calibration.

- If you do not calibrate the camera first, and there is a strong lens distortion, this may cause the Vision-to-Robot to fail.

![Example of a Grid of Dots Calibration Target](image)

**Figure 11** Example of a Grid of Dots Calibration Target

**Calibrating the Camera with a Calibration Target**

1. In the Cameras tab of the System Devices manager, select the camera you want to calibrate in the Devices list.

2. Click the 'Calibrate Camera' icon.

3. The Vision Calibration Wizard opens, beginning the vision (camera) calibration process.

4. Follow the instructions in the wizard, then return to this tutorial once the calibration is finished.

5. If you need help during the Calibration process, Click Help in the Calibration Wizard.
Getting Started with AdeptSight - Calibrating the Camera

Figure 12 Starting the Camera Calibration Wizard

'Calibrate Camera icon launches camera 'calibration wizard

Warning symbols indicates non-completed calibration

Next: Creating a Vision Sequence
Creating a Vision Sequence

A sequence is a series of tasks that are executed by vision tools. When you execute a sequence, each tool in the sequence executes in order. You add, remove, and edit the vision tools in the Sequence Editor.

Saving a Sequence

All sequences in the Sequence Manager are saved when you save the vision project.

- Sequences are saved as part of the project, not individually.
- Project files are saved with the extension hsproj.

To save the vision sequence:

1. Click the 'Save Project' icon:
2. Save the project as GettingStarted.hspoj

Opening the Sequence Editor

By default, there is already a first sequence in the application.

1. Select the first sequence in the list.
2. In the Sequence Manager, click the 'Edit Sequence' icon:
3. The Sequence Editor window opens. Continue to the next step of the tutorial for information on the Sequence Editor.

Figure 13 New vision sequence in the Sequence Manager

Next: Adding Tools to the Vision Sequence
Getting Started with AdeptSight - Adding Tools to the Vision Sequence

Adding Tools to the Vision Sequence

A vision sequence is built by adding vision tools to the sequence. These tools are added in the Sequence Editor Window.

In this module you will add an image acquisition tool, called **Acquire Image** and an object finding tool, called **Locator**.

**The Sequence Editor Window**

![The Sequence Editor interface](image)

When you first open the Sequence Editor, it is empty, as illustrated in Figure 14. Tools are added by dragging them from the Toolbox into the Process Manager area labeled "Drop Tools Here".

**Add an Acquire Image Tool**

**Acquire Image** is the first tool to add because it supplies images to other tools in the sequence.

1. In the Toolbox, select **Acquire Image** and drag it to the area marked "Drop Tools Here".
2. The Process Manager should look like the image shown Figure 15.
3. You are now ready to acquire images for the application.

![Acquire Image Tool Added to the Editor](image)

Next: **Acquiring Images**
Acquiring Images

The **Acquire Image** tool provides the images to subsequent tools in the sequence, such as the Locator tool that you will add later.

Displaying Images

Executing the Acquire Image tool displays images. You can also preview images as a continuous live display or as single static images.

1. Select the Basler camera in the list if it is not already selected. If you are not using a camera for this tutorial, see the note below.

2. To grab an image, click the 'Execute tool' icon:

3. The display should now contain an image.

4. Place an object in the field of view of the camera. To assist you in positioning the object, use the Live display mode, by clicking the 'Live Mode' icon:

5. In the Live display mode, the word **Live** appears at top left of the display, as shown in **Figure 16**

6. If you are not satisfied with the image quality, click the 'Camera Properties' icon to access and edit the camera parameters:

7. To exit the Live display mode, click again on the 'Live Mode' icon.

---

**Figure 16** Live Display in the Sequence Editor
If you are following this tutorial without a camera, select **Emulation** in the **Camera** dropdown list. The properties icon opens the **Emulation** dialog from which you can import images, such as those contained in the Tutorials folder of the AdeptSight program folder.

**Next:**  [Adding a Locator Tool](#)
Adding a Locator Tool

The Locator tool searches for the objects you have defined in your application and returns results on the location of the objects it finds.

1. In the **Toolbox**, select **Locator** and drag it into the Process Manager area, below the **Acquire Image** tool.

   ![Locator Tool added to the vision sequence](image)

   **Figure 17** Locator Tool added to the vision sequence

2. A Locator tool should now appear in the Process Manager area, as shown in **Figure 17**.
   
   Beside **Input**, select Acquire Images. **Input** defines the tool that provides images to the Locator tool.

3. Under **Location**, leave the **Entire Image** check box enabled. This ensures that the search process will look for objects in the entire image provided by the camera.

4. Execute the sequence at least once to ensure that the Locator has an image available before continuing to the next step.
   
   To execute the sequence, click the execute sequence icon in the toolbar:

5. You are now ready to create a model for the object that you want to find with this application.

   Next:  [Creating a Model]
Creating a Model

To find a specific part or object, AdeptSight must have an active model of the part. You will now create the model for the part you want to locate with this application.

![Basic Model Edition mode provides quick model-building](image)

**Figure 18** Basic Model Edition mode provides quick model-building

**Creating a New Model**

To create a model:

1. Place the object in the camera field of view and acquire an image by executing the sequence.

   To execute the sequence, click the 'Execute Sequence' icon: ⏯️

2. In the Models section, click the '+' icon to create a new model. The display is now in Model Edition mode as illustrated in Figure 18.

3. The Model's bounding box appears in the image as a green rectangle.

4. Drag and resize the green bounding box to completely enclose the entire object. Green outlines show the contours that have been selected to create the model.

5. Ignore the yellow axes marker for now, it will be covered in the next module of the tutorial.

6. Click Done to complete the creation of the model. This new model now appears as Model0 in the list of Models.

Next: Editing the Model
Editing the Model

Each modeled part has a frame of reference called the Object coordinate system. The origin of an object's coordinate system is the position that is returned in the results when an instance of this object is found.

In this module you will edit the model to reposition the coordinate system.

1. Select **Model0** in the list of models.
2. Click **Edit** to enter the Model Edition mode.
3. Reposition the yellow axes marker that indicates the position and orientation of the object's coordinate system:
   - To rotate the marker, click on the arrow of the X or the Y-axis and drag with the mouse.
   - To move the marker, click the origin of the X and Y axes and drag with the mouse.
   - You can also drag the arrow end of the axes to stretch the marker to help align the marker over long features.
4. Once you have finished positioning the marker, click **Done** to apply the changes made to the Model and close the Model Editor.

The Model you have created should be satisfactory and ready to use. If it needs further editing, see the next step of the tutorial: **Editing the Model in Expert Mode (optional)**.

- Typically the next step is to calibrate the gripper offset for the model, which is required for robot handling of parts. The gripper offset calibration is not required for this tutorial; it is explained in other AdeptSight tutorials.
- Optionally, you can continue to edit the model, as explained in the following module.
Getting Started with AdeptSight - Editing the Model

Next: Editing the Model in Expert Mode (optional) or skip to Configuring Search Parameters
Editing the Model in Expert Mode (optional)

If you want to edit model features, follow the steps below. Otherwise go to the next module: Configuring Search Parameters

In this module you will edit model features with the Expert model-edition mode.

1. If you are not in model edition mode, Select Model0 in the list of models and click Edit to open the Model Edition mode.
2. Click Expert to enter the advanced model-edition mode.
3. Under Show, select Outline to display the outline-level features of the Model. Outline features are coarser than the Detail features. The model contains features in both Outline and Detail level. You can edit features in either of these levels.
4. Once you have finished editing the Model, click Done to exit the Model Editor.

Sections below explain a few basic model-edition tasks. More extensive information on editing models is available in the online User Guide.

To build/rebuild a model

1. Click Build to build a new model. This rebuilds the model using the current 'Expert' mode parameters.
2. Use the Feature Selection slider to modify the amount of features that are added when you Build the model.
3. The model is rebuilt each time you click Build. This will undo any manual changes made to the model such as adding or removing features.
4. Click Apply to save the modifications to the model and Done to exit model edition mode.

To delete a feature:

1. Select a feature by clicking or double-clicking the feature. The selected feature is displayed as a bold red contour.
2. Click the Delete key to remove the feature. The contour now appears in blue.
3. Click Apply to save the modifications to the model.

To add a feature:

1. Double click on a blue contour to select it.
2. Click the Insert key to add the selected contour to model features
3. Click Apply to save the modifications to the model.
Getting Started with AdeptSight - Editing the Model in Expert Mode (optional)

Figure 21  Advanced Model Editing in Expert Mode

Models Contain both Detail and Outline Level Features

Next:  Configuring Search Parameters
Configuring Search Parameters

Search parameters set basic constraints for the search process. This module introduces you to editing of the basic search parameters.

![Basic Search Parameters](image)

**Figure 22** Configuring Search Parameters

1. Under **Search**, locate **Min Model Recognition (%)**.

2. Replace the default value (50) by 75. This instructs the Locator to search only for objects that contain at least 75% of the feature contours that were defined in the Model.

3. Leave other parameters to their default settings. For more information on this subject, consult the online User Guide.

You can experiment later with other Search constraints:

- **Scale**: Select **Range** then select a scale range to find objects of varying scale.

- **Rotation**: Select **Nominal** and then enter rotation value to find only those objects that are positioned in the defined angle of rotation.

- **Instance to Find**: Restricts the number of objects that the Locator can find in an input image.

Next:  [Running and Testing the Locator](#)
Running and Testing the Locator

Now that you have created a Model and configured search parameters, you will verify that the Locator tool finds the object in images similar to the one that was used to create the model.

1. Click the 'Execute Sequence' icon in the toolbar:

2. When an object is found, it is shown with a blue contour. The results for the object appear in the grid below the display. See Figure 23.

3. If results do not appear in the grid, click on the Locator tool interface. The name of the tool ('Locator') will be highlighted in blue as shown in Figure 23.

4. Move the object, or add more objects in the field of view.

5. The results for the found instances are updated every time you press the 'Execute Sequence' button.

Test in Continuous Mode

1. To start a continuous running mode, click the 'Continuous Loop' icon in the toolbar:

2. Click 'Execute Sequence' icon. The application should run in continuous mode.

3. Exit the continuous running mode by clicking 'Stop Sequence':

Figure 23  Display and Results of Found Objects

Next: Adding a Frame-Based Vision Tool
Adding a Frame-Based Vision Tool

In this module, you will add a tool to the vision application. This tool will be configured to use AdeptSight’s frame-based positioning feature.

In this tutorial, you will add an Image Histogram tool that will be positioned relative to the Locator tool frame of reference.

Adding the Image Histogram Tool

1. From the Toolbox, select and drag an Image Histogram tool below the Locator tool.
2. Alternatively, you can right click in the Process Manager and select the tool from the context menu.
3. The Histogram Tool should appear below the Locator tool as illustrated in Figure 24.
4. In the Input box, select: ‘Acquire Image’. This instructs the histogram tool to use input images provided by the Acquire Image tool.

![Image Histogram Tool added to the Vision Sequence](image)

In the following Module you will configure the Image Histogram Tool.

Next: Configuring Frame-Based Positioning of a Vision Tool
Configuring Frame-Based Positioning of a Vision Tool

When a tool is frame-based, it is positioned relative to another tool, which is the "frame provider".

- When the vision sequence executes, the frame-based tool is automatically positioned relative to the results provided by the frame-provider tool.
- In this tutorial, the Locator tool will be the frame provider.

Positioning the Histogram Tool

1. In the Input Frame text-box, select Locator.
2. Enable the All Frames check-box.
3. Click the Location button. This opens the tool’s Location dialog, as shown in Figure 25
   - The Location parameters define the area of interest to which the tool will be applied.
   - This area of interest is shown as a green bounding box in the image display.
4. Enter values in the Location dialog, or manually set the bounding box in the display. The X and Y positions and Rotation are expressed relative to the frame of reference represented in blue in the display area.
5. To modify the bounding box in the display area, click the Selection icon then use your mouse to drag handles, or to rotate the X-axis marker.
6. Position the bounding box to an area in the image. As shown in Figure 25, an Image Histogram tool can be placed to analyze the area just beyond the edges of an object, so that the results can be used to determine if the area is free of obstacles.
7. Once the bounding box is positioned, click OK to apply the parameters.

Figure 25 Frame-Based Positioning of the Image Histogram Tool

Enable 'Selection' icon to resize and rotate the bounding box in the display.

Green box shows region of interest of Histogram tool

Blue axes marker represents a Locator frame
Testing the Image Histogram Tool

For this tutorial, ignore other parameters and settings of the Histogram tool.

To execute tool and view results:

1. Execute the sequence by clicking the 'Execute Sequence' icon in the toolbar:

2. When an object is found by the Locator, the Histogram tool is applied to the area that was defined in the Location window. The Histogram tool is represented by a green rectangle, as shown in Figure 26.

3. Verify the histogram results in the grid below the display.

4. If the Image Histogram results do not appear in the display or grid of results, click on the 'Image Histogram' title. The tool title should be blue; other tool titles will be displayed in black. See Figure 26.

5. Results are updated every time the sequence is executed. See step 1.

6. If you enable the Results Log and define a log file you can save these results for further statistical analysis.

To test in continuous mode:

1. To start a continuous running mode, click the 'Continuous Loop' icon in the toolbar:

2. Click 'Execute Sequence' icon. The application will run in continuous mode.

3. Exit the continuous running mode by clicking the 'Stop Sequence' icon:

![Figure 26 Histogram Tool Results](image)

To complete the tutorial, you will save and run the sequence from the Sequence Manager.

Next: Completing the Vision Application
Completing the Vision Application

Return to the Sequence Manager window by clicking on Sequence Manager window or the Adept DeskTopinterface.

Renaming the Vision Sequence

Unless you previously modified the name of the sequence, it is still named 'New Sequence'.

To rename the sequence:

1. In the Sequence Manager, left click once on 'New Sequence' to select the sequence.
2. Left-click once again, on the sequence to enter the 'edit/renamer' mode.
3. Type in a new name for the sequence, for example 'Histogram Inspection'.

Saving the Vision Project

Saving the vision project saves the vision sequence and the configuration of the vision tools, including all models.

To save the vision project:

1. In the Sequence Editor toolbar, click the 'Save Project' icon:
2. Save the project as GettingStarted.hsproj.

Executing the Vision Application

When you execute the vision project from the Sequence Manager, this executes all sequences that are in the application.

To execute the vision application:

1. Select the sequence in the list.
2. In the toolbar, click the 'Continuous Loop' icon the toolbar to enable continuous running of the application:
3. Click the 'Execute Sequence' icon in the toolbar:
   The flag beside the sequence is now green, indicating that the sequence is running.
4. To stop the application, click the 'Stop Sequence' icon:

You have completed the tutorial!

Continue learning about AdeptSight in the following tutorials:

- AdeptSight Pick-and-Place Tutorial
- AdeptSight Conveyor Tracking Tutorial
AdeptSight Pick-and-Place Tutorial

This tutorial will walk you through the creation of a basic pick-and-place application for a single robot and single camera.

This tutorial assumes you have a basic knowledge of Adept robotic systems, MicroV+ and Adept DeskTop. If you are new to 2.0, we recommend that you start with the Getting Started with AdeptSight tutorial.

This tutorial uses as an example a system with:

- AdeptSight 2.0 running from Adept DeskTop
- A Basler Camera (A610f or A631f)
- A Cobra iSeries robot with AIB controller

Some steps may differ from the tutorial, if your system uses a CX controller, or you are not working in Adept DeskTop.

**Tutorial Overview**

- Overview of the Pick and Place System
- Start an AdeptSight Vision Project
- Verify Camera Calibration and Configuration
- Connect to the Controller
- Assign the Robot to the Camera
- Calibrate the Vision System to the Robot
- Create a Vision Sequence
- Add Tools to the Vision Sequence
- Acquire Image
- Add a Locator Tool
- Create a Model
- Calibrate the Gripper Offset for the Model
- Configure Locator Search Parameters
- Run and Test the Locator
- Integrate AdeptSight with a MicroV+ Program

**System Requirements**

- A PC running AdeptSight and Adept DeskTop software
- The camera provided with AdeptSight, or another DirectShow-compatible IEEE 1394 camera.
• A Cobra iSeries robot with AIB controller or an Adept robot controlled by a CX controller.

Before Starting the Tutorial

You will need a few identical objects that you will pick with the robot. These same objects can be used to calibrate the system, during the Vision-to-Robot calibration wizard.

Before starting this tutorial you should:

1. Install the camera.
2. Install the software.
3. Calibrate the camera.

Please refer to the Getting Started with AdeptSight tutorial if you need help with any of these preliminary steps.

Next: Overview of the Pick and Place System
Overview of the Pick and Place System

In this tutorial you will set up a system that picks parts from a work surface.

- AdeptSight acts as a vision server that provides vision guidance to controller.
- Vision applications are created on the PC.
- The MicroV+ (or V+) program on the controller, through Adept DeskTop, integrates the vision application to the motion control.
- Easy-to-use AdeptSight Calibration Wizards allow you to calibrate the entire system to ensure accurate part finding.

![Diagram of a Robotic Setup with AdeptSight - Data Flow Schema](image)

**Figure 27** Overview of a Robotic Setup with AdeptSight - Data Flow Schema

Next: Connect to the Controller
Start an AdeptSight Vision Project

Vision applications are created and managed in the Vision Project manager window in AdeptSight.

Opening AdeptSight

1. From the Adept DeskTop menu, select View > AdeptSight.
2. AdeptSight opens in the Vision Project window, similar to Figure 28.

![Figure 28 The Vision Project manager window](image)

The Vision Project Interface

A vision project consists of one or more vision sequences, which are managed and run from the Sequence Manager that is part of Vision Project interface.

- From the Sequence Manager you open the Sequence Editor to add and configure vision tools.
- From the System Devices Manager you add, manage and configure the camera, controllers, robots and conveyor belts needed for your application.

Creating and Name the New Sequence

You will now create and name the vision sequence that you will use for this tutorial.

1. By default, the Vision Project list contains a sequence named NewSequence.
   If the list is empty, create a new sequence by clicking the 'Create Project' icon.
2. Select NewSequence in the list then left-click once on the name to edit the name.
3. Name the sequence PickAndPlace. The project now contains one vision sequence named PickAndPlace.
4. Click the 'Save Project' icon to save the vision project now.
5. Save the project as PickAndPlace.hsproj.
Next you will verify the camera that you will use for the application.

**Next:** [Verify Camera Calibration and Configuration](#)
Verify Camera Calibration and Configuration

When the camera is correctly installed and recognized by the system it appears in the System Devices manager, in the Cameras tab, as shown in Figure 30.

![System Devices toolbar](image)

**Figure 30** Verifying Camera State and Calibration Status

**Camera Calibration**

If you have not previously calibrated the camera, a warning symbol appears to the right of the camera State icon, as shown in Figure 30.

**Choose a camera calibration method:**

1. **Calibrate the camera now**, by launching the 2D Vision Calibration Wizard from the toolbar. This requires a "grid of dots" calibration target. Sample calibration targets are provided in the AdeptSight support files, in the AdeptSight installation folder: ...\AdeptSight 2.0\Tutorials\Calibration.

2. **Calibrate the camera later**, through the vision to robot calibration, as explained later in this tutorial. This will provide acceptable accuracy in most cases. However, a separate vision calibration can provide increased accuracy to your application.

Calibrating the camera only through the vision-to-robot calibration will not correct for lens distortion. In some cases, strong lens distortion may cause the vision-to-robot calibration to fail if you do not calibrate the vision first.

For more details on this subject, see Why is Vision Calibration Important?

**Camera Configuration**

If you have not yet verified the quality of the images provided by the camera, you can verify and configure the camera now.

**To verify camera images:**

1. In the Devices list, select the camera.

2. In the System Devices toolbar, click the 'Live Display' icon: 📺

3. Use the Live Display window to set camera focus and lens aperture.
To modify camera properties:

1. If you need to configure other camera settings, click the 'Camera Properties' icon:

2. Refer to the camera documentation for information on setting/changing camera properties and parameters.

You are now ready to add devices to the application.

Next: Connect to the Controller
Connect to the Controller

You will now start to set up the devices that will be used by the vision guidance application.

Adding the Controller from Adept DeskTop

If you are using AdeptSight from within Adept DeskTop, a controller device is present in the Controllers tab, as shown in Figure 31.

You must connect to continue setting up this application.

If you have a multiple-controller license, or are creating the application outside the Adept DeskTop environment you may have to add a controller in the Controllers tab. Consult the online User Guide if you need help adding a controller.

System Devices

<table>
<thead>
<tr>
<th>Device</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIB Controller</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 31** AIB Controller device displayed in the System Devices Manager

Connecting to the Controller

1. From the Adept DeskTop menu, select File > Connect...
2. Connect to the controller. Refer to the Adept DeskTop online help if needed.
3. When the controller is connected, the State icon for the controller becomes green, and a robot is attached to the controller, as shown in Figure 32.

**Figure 32** AdeptSight connected to controller with robot

Next: Assign the Robot to the Camera
Assign the Robot to the Camera

You must now assign to the camera the robot that will be used for the vision application. Later in this tutorial you will calibrate the camera and robot together in the Vision-to-Robot calibration.

1. Select the **Cameras** tab.
2. Select the camera you will use for the vision guidance application.
3. In the **System Devices** toolbar, select the 'Add Robot icon:
4. The **Select a Robot** window opens as shown in **Figure 33**.

![Select a Robot](image)

**Figure 33** Assigning a Robot to the Camera

5. From the list, select the robot that you will use for the vision guidance application and click **OK**.
6. The robot is now assigned to the selected camera in the Devices List, as shown in **Figure 35**.

![Robot Assigned](image)

**Figure 34** Robot Assigned to the Camera

You will now need to calibrate the system using a Vision-to-Robot calibration wizard.

Next: **Calibrate the Vision System to the Robot**
Calibrate the Vision System to the Robot

Vision-to-Robot calibration ensures that the robot will accurately move to parts that are seen by the camera.

The calibration enables AdeptSight to accurately transform coordinates in the camera frame of reference to coordinates in the robot frame of reference.

To start the calibration wizard:

1. In the Sequence Devices Manager, select Cameras tab.
2. In the list of devices, select the robot (Robot1).
3. Click the 'Calibrate Vision to Robot' icon, as shown in Figure 35.
4. The Calibration Interview Wizard begins, beginning the Vision-to-Robot calibration process. Questions in the Interview Wizard determine the type of calibration required for your system.

To carry out the calibration:

1. Follow the instructions in the wizard, then return to this tutorial once the calibration is complete.
2. If you need help during the Calibration process, click the Help button in the Calibration Wizard.

Figure 35 Starting Vision-to-Robot Calibration from the Vision Manager

Next: Create a Vision Sequence
Create a Vision Sequence

A sequence is a series of tasks that are executed by vision tools. When you execute a sequence, each tool in the sequence executes in order. You add, remove, and edit the vision tools in the **Sequence Editor**.

Saving a Sequence

All sequences in the Sequence Manager are saved when you save the vision project.

- Sequences are saved as part of the project, not individually.
- Project files are saved with the extension "hsproj".

Click the **Save** icon to save changes you have made up to now:

Opening the Sequence Editor

**To open the Sequence Editor:**

1. In the Sequence Manager, select the **PickAndPlace** sequence.
2. In the toolbar click the 'Edit Sequence' icon. See Figure 36.

![Figure 36](image)

**Figure 36** PickAndPlace sequence in the Sequence Manager

Next: Add Tools to the Vision Sequence
Add Tools to the Vision Sequence

A vision sequence is built by adding vision tools to the sequence. These tools are added to the Process Manager area of the Sequence Editor interface. See Figure 37

In this module you will add an image acquisition tool, called Acquire Image tool.

The Sequence Editor Window

The Toolbox contains the tools available for building sequences.

Add tools by dragging them from the Toolbox into the Process Manager area, labeled "Drop Tools Here".

Adding an Acquire Image Tool

Acquire Image is the first tool to add because it supplies images to other tools in the sequence.

To add the Acquire Image tool:

1. In the Toolbox, select Acquire Image and drag it into the Process Manager area, that reads 'Drop Tools Here'.
2. The Process Manager (blue area) now contains the Acquire Image tool. See Figure 38.
3. You are now ready to acquire images.

Next: Acquire Image
Acquire Image

The **Acquire Image** tool will provide the images taken by the camera to the Locator tool.

Displaying Images

Images acquired by the tool appear in the display, as illustrated in Figure 39.

![Figure 39](image)

**Figure 39** Live Display of camera images in the Sequence Editor

**To display acquired images:**

1. In the toolbar, click the 'Execute Sequence' icon:  
   Alternatively, you can execute only the Acquire Image tool by clicking the 'Execute Tool' icon:  

2. If the Acquire Image tool is unable to get images from the camera, the 'Status Failed' icon appears in the tool title bar:
   In such a case, return to the Vision project window and verify that the camera is active and is detected by the system.

3. If objects are not correctly positioned in the field of view, or if you need to adjust the camera position, focus or aperture, open the Live display mode by clicking the 'Live Mode' icon:

4. To exit the Live display, click the 'Live Mode' icon.

5. To preview single images, click the 'Image Preview' icon:

You will now add the Locator tool to your application.

Next: **Add a Locator Tool**
Add a Locator Tool

The Locator tool will search for the objects you have defined in your application and returns results on the location of the objects it finds.

To add the Locator tool:

1. In the Toolbox, select Locator and drag it into the Process Manager frame, below the Acquire Image tool, as shown in Figure 40.

2. Under Location, leave the Entire Image check box enabled. This ensures that the search process will look for objects in the entire image provided by the camera.

You are now ready to create a model for the object that will be handled by the application.

Next: Create a Model
Create a Model

To find a specific part or object, AdeptSight must have an active model of the part. You will now create the model for the part you want to locate with this application.

Figure 41 Basic Model-Edition Mode Provides Quick Model-Building

Creating a New Model

To create a model:

1. Place the object in the camera field of view and acquire an image by executing the sequence.
2. In the Models section, click the ‘+’ icon to create a new model. The display is now in Model Edition mode as illustrated in Figure 41.
3. Drag and resize the green bounding box to completely enclose the entire object. Green outlines show the features that have been selected to add to the model.
4. Drag and rotate the yellow axes marker to position the coordinate system of the model.
5. If you need to edit the model, for example add or remove features, click Expert to enter the Expert Model Edition mode. Refer to the online User Guide or the 'Getting Started' tutorial if you need help for this feature.
6. Click Done to complete the creation of the model. This new model now appears as Model0 in the list of Models.

Next: Calibrate the Gripper Offset for the Model
Calibrate the Gripper Offset for the Model

For each object model, you must carry out a gripper offset calibration to ensure that parts of this type will be correctly gripped.

- The Gripper Offset Calibration teaches AdeptSight the position on an object to which the robot will move to manipulate the object. If you do not carry out the Gripper Offset Calibration, the robot may not be able to manipulate the found object.

- This calibration needs to be carried out at least once for each model.

- You must recalibrate the gripper offset for a model if the model coordinate system is modified, or if the camera-robot setup is modified.

Launched the Gripper Offset Calibration

A Gripper Offset indicator appears to the right of models in the list of models as shown in Figure 43. The Gripper Offset Calibration wizard walks you through the process of calibrating the Gripper Offset for the model.

To start the calibration wizard:

1. Select the model from the list of models.

2. Select the 'Model Options' icon:

3. From the menu, select Gripper Offset > Manager as shown in Figure 42. This opens the Gripper Offset Manager shown in Figure 43.

4. In the Gripper Offset Manager, select the wizard icon, as shown in Figure 43.
Carrying out the Gripper Offset Calibration

The Gripper Offset Calibration is presented as a Wizard that walks through the steps required for assigning Gripper offsets to a Model.

1. Follow the instructions in the Wizard to complete the process. Click 'Help' for assistance during the calibration.

2. Once the calibration is complete, the Gripper Offset is added to the Gripper Offset Manager.

3. Click Close to return to the Locator tool interface. A check mark icon indicates that the calibration has been completed for the model:

4. Repeat the Gripper Offset Calibration for each model you create.

Before starting the Gripper Offset Wizard: Make sure you have an object that is identical to the object used to create the model.

Next: Configure Locator Search Parameters
Configure Locator Search Parameters

Search parameters set basic constraints for the search process. This module introduces you to editing of the basic search parameters.

![Locator Search Parameters](image)

**Figure 44** Configuring Search Constraints

You can leave **Search** parameters to their default value and continue the Tutorial.

However you may need to make some changes to the basic search parameters. Below is a brief description of these parameters. Refer to the online User Guide for more details on Search parameters.

- **Instances to Find** determines the maximum number of instances that can be searched for, regardless of Model type. To optimize search time you should set this value to no more than the expected number of instances.

- **Scale**: If you want to find parts that vary in scale, select **Range** (instead of **Nominal**) then select the scale range of the objects to find.

- **Rotation**: If you want to find only objects positioned at a specific orientation, select **Nominal** (instead of **Nominal**) and set the required value. The Locator will search only for parts positioned in the defined angle of rotation.

- **Min Model Recognition**: Select the percentage of object contours that are required to locate a valid object instance.

  Lowering this parameter can increase recognition of occluded instances but can also lead to false recognitions. A higher value can help eliminate instances in which objects overlap.

**Next**: [Run and Test the Locator](#)
Run and Test the Locator

Now that you have created a Model and configured search parameters, you will verify that the Locator tool finds the object in images similar to the one that was used to create the model.

1. Click the 'Execute Sequence' icon at the top of the window:

2. When an object is found, it is shown with a blue contour. The results for the object appear in the grid below the display. See Figure 45.

3. Verify in the grid of results that the instance was located correctly.

4. Move the object or add more objects in the field of view.

5. The results for the found instances are updated every time you press the 'Execute Sequence' button.

Test in Continuous Mode

1. To start a continuous running mode, click the 'Continuous Loop' icon in the toolbar:

2. Click 'Execute Sequence' icon. The application should run in continuous mode.

3. Exit the continuous running mode by clicking 'Stop Sequence':

Next: Integrate AdeptSight with a MicroV+ Program
Integrate AdeptSight with a MicroV+ Program

To enable the robot to handle the objects found by the vision application, you now need to create or add a MicroV+ program.

For this tutorial we have provided a sample application that instructs the robot to pick up a model-defined object at whatever position and whatever angle it has been found in the field of view.

If your robot has a pointer-type tool instead of a gripper, the robot will point to the located objects.

Figure 46 Adding and assigning the Micro V+ program

1. In the Adept DeskTop window, Open the Code Library tab. If it is not visible, open the Code Library from the menu: select View > Code Library.

2. In the list of code examples, also called clips, select Vision-guided pick-and-place in the AdeptSight Examples folder. See Figure 46.


4. In the New Program window, click Create.

5. The program is added to the Program Manager.

You must now assign the program to a task in the program execution tool.

1. Select tutorial in the Program Manager list. See Figure 46.

2. Drag it to Task 0 in the Task Manager list.

3. Execute the program.

You have completed the tutorial!

Continue learning about AdeptSight in the following tutorials:

- AdeptSight Pick-and-Place Tutorial
• AdeptSight Standalone C# Tutorial
AdeptSight Conveyor Tracking Tutorial

This tutorial will walk you through the creation of a basic vision application that uses conveyor tracking. This tutorial assumes you have a working knowledge of Adept robotic systems, V+, and Adept DeskTop. If you are new to AdeptSight, we recommend that you start with the Getting Started with AdeptSight tutorial.

System Requirements for this Tutorial

- A PC running AdeptSight and Adept DeskTop software.
- A conveyor tracking license.
- The camera provided with AdeptSight, or another DirectShow compatible camera.
- An Adept robot controlled by a CX controller
- A conveyor belt with an encoder

This tutorial illustrates a system with a Basler camera, a Cobra s-Series robot, and AdeptSight running in Adept DeskTop. Steps may differ if you are using another type of camera, or running Adept DeskTop from a standalone application.

Tutorial Overview

- Overview of the Conveyor-Tracking System
- Start an AdeptSight Vision Project
- Verify Camera Calibration and Configuration
- Connect to the Controller
- Add the Conveyor Belt
- Configure V+ to Define the Latching Signal
- Assign the Conveyor Belt to the Camera
- Assign the Robot to the Camera
- Calibrate the Vision System to the Robot and Belt
- Create a Vision Sequence
- Add the Acquire Image Tool
- Configure Latching Parameters
- Add a Locator Tool
- Create a Model
- Calibrate the Gripper Offset for the Model
- Add an Overlap Tool
- Add a Communication Tool
Before Starting the Tutorial

You will need a few identical objects that you will pick with the robot. These same objects can be used to calibrate the system, during the Belt Calibration Wizard.

Before starting this tutorial you should:

1. Install the camera. Make the required connections between the camera and controller to enable belt encoder latching. Consult the camera documentation for more information.
   For the Basler A601f and A631f connections see Table 1.

2. Install the software.

3. Calibrate the camera.

Table 1 Basler Camera Connections to the CX Controller

<table>
<thead>
<tr>
<th>Connect:</th>
<th>To:</th>
</tr>
</thead>
<tbody>
<tr>
<td>CX Controller XDIO Connector</td>
<td>Basler Camera Pinout</td>
</tr>
<tr>
<td>24 V Output</td>
<td>Pin 7</td>
</tr>
<tr>
<td>Pin 41, 42, 43, or 44</td>
<td>(Out VCC Comm)</td>
</tr>
<tr>
<td>CX Controller XDIO Connector</td>
<td>Basler Camera Pinout</td>
</tr>
<tr>
<td>Input Signal 1001,1002,1003, or</td>
<td>Pin 4 - Output 0</td>
</tr>
<tr>
<td>1004</td>
<td>(integrate enable output)</td>
</tr>
<tr>
<td>Pin 1, 3, 5, or 7</td>
<td></td>
</tr>
<tr>
<td>CX Controller XDIO Connector</td>
<td>CX Controller XDIO Connector</td>
</tr>
<tr>
<td>Connect Return Signal 1001,1002</td>
<td>Ground on CX Controller</td>
</tr>
<tr>
<td>1003, or 1004</td>
<td>Pin 47, 48, 49, or 50</td>
</tr>
<tr>
<td>Pin 2, 4, 6 or 8</td>
<td></td>
</tr>
</tbody>
</table>

Please refer to the Getting Started with AdeptSight tutorial if you need help with any of these preliminary steps.
Overview of the Conveyor-Tracking System

In this tutorial you will set up a system that picks parts from a moving conveyor-belt.

- AdeptSight acts as a vision server that provides vision guidance to controller.
- Vision applications are created on the PC.
- V+ programs on the controller, via Adept DeskTop, integrate the vision application to the motion control.
- AdeptSight Motion tools ensure correct part handling and communication between the vision results and the motion control system.
- Easy-to-use AdeptSight Calibration Wizards allow you to calibrate the entire system, including the conveyor belt, to ensure accurate part finding.

![Overview of a Conveyor Tracking Setup with AdeptSight - Data Flow Schema](image)

**Figure 48** Overview of a Conveyor Tracking Setup with AdeptSight - Data Flow Schema

Next:  [Start an AdeptSight Vision Project](#)
Start an AdeptSight Vision Project

Vision applications are created and managed in the Vision Manager window in AdeptSight.

Opening AdeptSight

1. From the Adept DeskTop menu, select View > AdeptSight.
2. AdeptSight opens in the Vision Project window, similar to Figure 49.

![Figure 49 The Vision Project manager window](image)

The Vision Project Interface

A vision project consists of one or more vision sequences, which are managed and run from the Sequence Manager that is part of Vision Project interface.

- From the Sequence Manager you open the Sequence Editor to add and configure vision tools.
- From the System Devices Manager you add, manage and configure the camera, controllers, robots and conveyor belts needed for your application.

Create and Name the New Sequence

You will now create and name the vision sequence that you will use for this tutorial.

1. By default, there is a blank vision sequence named 'NewSequence' in the Vision project.

   If the list is empty, create a new sequence by clicking the 'Create Project' icon:

2. Left-click once on 'NewSequence' to edit the name.
3. Name the sequence ConveyorTracking. The project now contains one vision sequence named ConveyorTracking.
4. Click the 'Save Project' icon to save the vision project now:
5. Save the project as ConveyorTracking.hsproj.
Next you will verify the camera that you will use for the application.

**Next:**  Verify Camera Calibration and Configuration
Verify Camera Calibration and Configuration

When the camera is correctly installed and recognized by the system it appears in the **System Devices** manager, in the **Cameras** tab, as shown in **Figure 51**.

![System Devices](image)

**Figure 51** Verifying Camera State and Calibration Status

### Camera Calibration

If you have not previously calibrated the camera, a warning symbol appears to the right of the camera **State** icon, as shown in **Figure 51**.

**Choose a camera calibration method:**

1. **Calibrate the camera now**, by launching the 2D Vision Calibration Wizard from the toolbar. This requires a "grid of dots" calibration target.
   Sample calibration targets are provided in the AdeptSight support files, in the AdeptSight installation folder: `...\AdeptSight 2.0\Tutorials\Calibration`.

2. **Calibrate the camera later**, through the vision to robot calibration, as explained later in this tutorial. This will provide acceptable accuracy in most cases. However, a separate vision calibration can provide increased accuracy to your application.

Calibrating the camera only through the vision-to-robot calibration will not correct for lens distortion. In some cases, strong lens distortion may cause the vision-to-robot calibration to fail if you do not calibrate the vision first.

For more details on this subject, see [Why is Vision Calibration Important?](#)

### Camera Configuration

If you have not yet verified the quality of the images provided by the camera, you can verify and configure the camera now.

**To verify camera images**

1. In the Devices list, select the camera.

2. In the System Devices toolbar, click the 'Live Display' icon: 📸

3. Use the Live Display window to set camera focus and lens aperture.
To modify camera properties

1. If you need to configure other camera settings, click the 'Camera Properties' icon:

2. Refer to the camera documentation for information on setting/changing camera properties and parameters.

You are now ready to add devices to the application.

Next: Connect to the Controller
Connect to the Controller

You will now start to set up the devices that will be used by the vision guidance application.

Adding the Controller from Adept DeskTop

If you are using AdeptSight from within Adept DeskTop, a controller device is present in the Controllers tab, as shown in Figure 52.

You must connect to the controller to continue setting up this application.

If you have a multiple-controller license, or are creating the application outside the Adept DeskTop environment you may have to add a controller in the Controllers tab. Consult the Adept DeskTop online help for assistance in adding a controller.

Connecting to the Controller

1. From the Adept DeskTop menu, select File > Connect...

2. Connect to the controller. Refer to the Adept DeskTop online help if needed.

3. When the controller is connected, the State icon for the controller becomes green, and a robot is attached to the controller, as shown in Figure 53.

Next: Add the Conveyor Belt
Add the Conveyor Belt

In this step you will add the conveyor belt, and assign a controller to the belt.

**To add the belt that will be used for the vision application:**

1. In the **System Devices** manager, select the **Belts** tab.
2. In the **System Devices** toolbar, click the 'Add Belt' icon.
3. The belt is added to the list, as shown in Figure 54.

![System Devices](image)

**Figure 54** Conveyor Belt added to the System Devices list

You must now assign a controller to the belt and set the encoder signal.

**To assign a controller and encoder to the belt:**

1. Select the newly-created **Belt**, in the **Device** list.
2. In the toolbar, click the 'Add Controller' icon. This opens the **Select a controller** window.
3. Select the controller that will be used for the application then click **OK**.
4. The controller now appears in the list as shown in Figure 55.

![Controller](image)

**Figure 55** Controller assigned to a conveyor belt

**To set the encoder signal:**

1. In the **Device** list, select the controller that is assigned to the belt.
2. Double-click in the **Encoder** column to edit the **Encoder** value. See Figure 56.
3. This tutorial uses 1 as the value of belt encoder index.

![Encoder](image)

**Figure 56** Selecting a belt encoder value
The Encoder value depends on the configuration of the connection between the controller and the belt and the encoder value defined in the **config_c utility**. Please refer to the CX Controller User Guide for more information.

**Next:** Configure V+ to Define the Latching Signal
Configure V+ to Define the Latching Signal

You now need to configure V+ to define which signal will latch the encoder, using the `config_c utility`. To instruct V+ which signal will latch the encoder, you need to access V+ Configuration Utility: `config_c utility`.

To open the `CONFIG_C Utility`:

1. In Adept DeskTop menu, select View > Debug Tools > Monitor Terminal.
2. In the Monitor Terminal window, execute CONFIG_C by typing the following:
   ```
   cd util
   load config_c
   execute 1 a.config_c
   ```
3. The CONFIG_C utility opens in the Monitor terminal window.

To configure the `CONFIG_C belt statement`:

1. In the *** Adept System Configuration Program *** page, select:
   ```
   '2' (2 => V+ Configuration System)
   ```
2. In the *** Controller Configuration Editor *** page, select:
   ```
   '2' (2 => Edit System Configuration)
   ```
3. In the *** V+ System Configuration Editor *** page, select:
   ```
   '9' (9 => Change ROBOT configuration)
   ```
4. The current V+ statements of the ROBOT section are then displayed, as shown in Figure 57.
5. Add a Belt definition if no belt is configured, or edit the statement if required. Refer to the Config_C documentation if you need help for this step.

   - For POS_LATCH [1], select the signal that will latch the encoder.
   - For POS_LATCH [2], select '1' (1 => None).
   - For LATCH_BUFFER enter '1'.

Next: Assign the Conveyor Belt to the Camera
Figure 57   Editing the Belt statement in the CONFIG_C Utility

Next:  Assign the Conveyor Belt to the Camera
Assign the Conveyor Belt to the Camera

In this step you will assign the conveyor belt to the camera.

The belt must be assigned to the camera before assigning the robot or other devices. If a robot has already been assigned to the camera you must remove the assigned robot. Once the belt is added you can assign the robot, as explained later in this tutorial.

1. In the System Devices manager, Select the Cameras tab.
2. Select the camera you will use for the vision guidance application.
3. In the System Devices toolbar, click the 'Add Belt' icon.
4. The Select a Belt dialog opens. In this dialog, select the belt and click OK.
5. The added belt now appears, assigned (attached) to the camera in the Device list as shown in Figure 58.

![Figure 58](image)

Next: Assign the Robot to the Camera
Assign the Robot to the Camera

You must now assign to the camera the robot that will be used for the vision application. Later in this tutorial you will calibrate the camera and robot together in the Vision-to-Robot calibration.

1. Select the **Cameras** tab.
2. Select the camera you will use for the vision guidance application.
3. In the **System Devices** toolbar, select the 'Add Robot' icon.
4. The **Select a Robot** window opens as shown in **Figure 59**.
5. From the list, select the robot that you will use for the vision guidance application and click **OK**.
6. The robot is now assigned to the selected camera in the Devices List, as shown in **Figure 60**.

You will now need to calibrate the system using a Vision-to-Robot calibration wizard.

**Next:** [Calibrate the Vision System to the Robot and Belt](#)
Calibrate the Vision System to the Robot and Belt

In this Module you will calibrate the system: the camera, the robot and the belt, with the appropriate Vision-to-Robot calibration wizard.

To calibrate this application you will use the Belt Calibration Wizard.

**To start the calibration wizard:**

1. In the System Devices Manager, select the **Cameras** tab.
2. In the list of devices, select the robot that is assigned to the camera (Robot1.)
3. Click the 'Calibrate Vision to Robot' icon:
4. The Calibration Interview Wizard begins, beginning the Vision-to-Robot calibration process. Questions in the Interview Wizard determine the type of calibration required for your system.

**To carry out the calibration:**

1. Follow the instructions in the wizard, then return to this tutorial once the calibration is complete.
2. In the Interview wizard, make sure you select options that specify that you are using a conveyor belt, as shown in Figure 61. This will select the Belt Calibration Wizard.
3. If you need help during the Calibration process, click the **Help** button in the Calibration Wizard.

![Calibration Interview Wizard](image)

**Figure 61** Selecting Belt-Conveyor option in the Calibration Interview Wizard

**Next:** Create a Vision Sequence
Create a Vision Sequence

A sequence is a series of tasks that are executed by vision tools. When you execute a sequence, each tool in the sequence executes in order. You add, remove, and edit the vision tools in the Sequence Editor.

Saving a Sequence

All sequences in the Sequence Manager are saved when you save the vision project.

- Sequences are saved as part of the project, not individually.
- Project files are saved with the extension "hsproj".

Click the 'Save Project' icon to save changes you have made up to now:

Opening the Sequence Editor

To open the Sequence Editor:

1. In the Sequence Manager, select the ConveyorTracking sequence.
2. In the toolbar click the 'Edit Sequence' icon:
3. The Sequence Editor opens, similar to Figure 62.

Figure 62  New vision sequence in the Sequence Manager

In this tutorial you will add the following tools to the sequence:
- Acquire Image
- Locator
- Overlap Tool
- Communication Tool
AdeptSight Conveyor Tracking Tutorial - Create a Vision Sequence

Next:  Add the Acquire Image Tool
Add the Acquire Image Tool

The **Acquire Image** tool is the first tool to add because it supplies images to other tools in the sequence.

The **Acquire Image** tool will provide the images taken by the camera to the Locator tool and define the latching parameters for the conveyor belt.

**To add the Acquire Image tool:**

1. In the **Toolbox**, select **Acquire Image** and drag it into the Process Manager area that reads 'Drop Tools Here'.

2. The Process Manager (blue area) now contains the **Acquire Image** tool. See Figure 63.

3. You are now ready to acquire images.

**To display acquired images:**

1. In the toolbar, click the 'Execute Sequence' icon.

2. Alternatively, you can execute the only the Acquire Image tool by clicking the 'Execute Tool' icon.

**To preview live images grabbed by the camera:**

1. Click 'Live Mode' icon:

   To exit the Live display, click the 'Live Mode' icon again.

2. To preview single images grabbed by the camera, click the 'Image Preview' icon:

   Live Mode an Image Preview modes do not execute the Acquire Image tool, they only display input from the camera.

![Figure 63 Live Display of camera images in the Sequence Editor](image)

You will now configure latching parameters.

**Next**: Configure Latching Parameters
Configure Latching Parameters

In a previous step, you configured V+ to define which signal will latch the encoder, using the config_c utility.

You must now set the corresponding latching parameter in the Acquire Image Tool.

To set the latching parameters:

1. Under **Latching Parameters** there are two sections **Robot** and **Belt**.
2. Under **Belt** select **Read Latched Value option** (add check mark).
3. Under **Robot**, do not select either option (leave check marks blank)

![Figure 64 Latching Parameters of the Acquire Image tool](image)

You will now add the Locator tool to your application.

**Next:** Add a Locator Tool
Add a Locator Tool

The Locator tool will search for the objects you have defined in your application and returns results on the location of the objects it finds.

To add the Locator tool:

1. In the Toolbox, select Locator and drag it into the Process Manager frame, below the Acquire Image tool, as shown in Figure 65.

2. Under Location, leave the Entire Image check box enabled. This ensures that the search process will look for objects in the entire image provided by the camera.

You are now ready to create a model for the object that will be handled by the application.

Next: Create a Model
Create a Model

To find a specific part or object, AdeptSight must have an active model of the part. You will now create the model for the part you want to locate with this application.

To create a model:

1. Place the object in the camera field of view and acquire an image by executing the Acquire Image tool.
2. In the Models section, click the '+' icon to create a new model. The display is now in Model Edition mode as illustrated in Figure 66.
3. Drag and resize the green bounding box to completely enclose the entire object. Green outlines show the features that have been selected to add to the model.
4. Drag and rotate the yellow axes marker to position the coordinate system of the model.
5. If you need to edit the model, for example add or remove features, click Expert to enter the Expert Model Edition mode. Refer to the online User Guide or the 'Getting Started' tutorial if you need help for this feature.
6. Click Done to complete the creation of the model. This new model now appears as Model0 in the list of Models.

Next: Calibrate the Gripper Offset for the Model
Calibrate the Gripper Offset for the Model

For each object model, you must carry out a gripper offset calibration that will enable AdeptSight to correctly pick up or move to the part.

- The Gripper Offset Calibration teaches AdeptSight the position on an object to which the robot will move to manipulate the object. If you do not carry out the Gripper Offset Calibration, the robot may not be able to manipulate the found object.

- This calibration needs to be carried out at least once for each model.

- You must recalibrate the gripper offset for a model if the model frame of reference is modified or the camera-robot setup is modified.

A Gripper Offset indicator appears to the right of models in the list of models as shown in Figure 68. The Gripper Offset Calibration wizard walks you through the process of calibrating the Gripper Offset for the model.

To start the calibration wizard:

1. Select the model from the list of models.

2. Select the 'Model Options' icon:

3. From the menu, select **Gripper Offset** > **Manager as shown in Figure 67**. This opens the Gripper Offset Manager shown in Figure 68.

4. In the Gripper Offset Manager, select the wizard icon, as shown in Figure 68.
Carrying out the Gripper Offset Calibration

The Gripper Offset Calibration is presented as a Wizard that walks through the steps required for assigning Gripper offsets to a Model.

1. Follow the instructions in the Wizard to complete the process. Click 'Help' for assistance during the calibration.

2. Once the calibration is complete, the Gripper Offset is added to the Gripper Offset Manager.

3. Click Close to return to the Locator tool interface. A check mark icon indicates that the calibration has been completed for the model.

4. Repeat the Gripper Offset Calibration for each model you create.

Before starting the Gripper Offset Wizard: Make sure you have an object that is identical to the object used to create the model.

Next: Configure Locator Search Parameters
Configure Locator Search Parameters

Search parameters set basic constraints for the search process. This module introduces you to editing of the basic search parameters.

![Locator Configuration](image)

**Figure 69** Configuring Search Constraints

You can leave **Search** parameters to their default value and continue the Tutorial. However you may need to make some changes to the basic search parameters. Below is a brief description of these parameters. Refer to the online User Guide for more details on Search parameters.

- **Instances to Find** determines the maximum number of instances that can be searched for, regardless of Model type. To optimize search time you should set this value to no more than the expected number of instances.

- **Scale**: If you want to find parts that vary in scale, select **Range** (instead of **Nominal**), then select the scale range of the objects to find.

- **Rotation**: If you want to find only objects positioned at a specific orientation, select **Nominal** (instead of **Range**) and set the required value. The Locator will search only for parts positioned in the defined angle of rotation.

- **Min Model Recognition**: Select the percentage of object contours that are required to locate a valid object instance. Lowering this parameter can increase recognition of occluded instances but can also lead to false recognitions. A higher value can help eliminate instances in which objects overlap.

Next: Run and Test the Locator
Run and Test the Locator

Now that you have created a Model and configured search parameters, you will verify that the Locator tool finds the object in images similar to the one that was used to create the model.

1. Click the 'Execute Sequence' icon at the top of the window:

2. When an object is found, it is shown with a blue contour. The results for the object appear in the grid below the display. See Figure 70.

3. Verify in the grid of results that the instance was located correctly.

4. Move the object or add more objects in the field of view.

5. The results for the found instances are updated every time you press the 'Execute Sequence' button.

Test in Continuous Mode

1. To start a continuous running mode, click the 'Continuous Loop' icon in the toolbar:

2. Click 'Execute Sequence' icon. The application should run in continuous mode:

3. Exit the continuous running mode by clicking the 'Stop Sequence' icon:

Next: Add an Overlap Tool
Add an Overlap Tool

You will now add an Overlap tool to the sequence. The purpose of the Overlap Tool is to make sure that parts moving on the belt are recognized only once.

- Because a part found by the Locator may be present in many images acquired by the camera, the Overlap tool ensures that the robot is not instructed to pick up the same part more than once.
- The Overlap tool requires input from the Locator tool.

![Figure 71 - Adding an Overlap Tool to the sequence](image)

1. In the Toolbox, under Motion Tools, select Overlap Tool and drag it into the Process Manager frame, below the Locator tool, as shown in Figure 71.
2. Under Input, leave Locator as the input provider.
3. Under Advanced Parameters, leave Encoder Ticks selected.
4. Test the Overlap Tool by executing the sequence:
   - When an instance is found by the Locator, *for the first time*, the object is highlighted in blue in the image.
   - In the following executions, the instances recognized by the Overlap Tool are highlighted in red.

Next you will add the Communication Tool.

Next: Add a Communication Tool
Add a Communication Tool

You will now add a Communication Tool to the vision sequence. The Communication Tool provides instructions to the controller for handling the vision results provided by the Overlap tool.

1. In the Toolbox, under Motion Tools, select Communication Tool and drag it into the Process Manager frame, below the Overlap tool, as shown in Figure 72.

2. In the Input text box, leave Overlap Tool as the input provider.

3. In the Robot text box, select the robot used by this application, as shown in Figure 72.

Figure 72  Adding a Communication Tool to the sequence

Select the robot that is used by the application

Acquire Image tool provides the input for the Locator tool
Locator provides the input for the Overlap Tool
Overlap Tool provides the input for the Communication Tool

Next: Integrate AdeptSight with a V+ Program
Integrate AdeptSight with a V+ Program

To enable the robot to handle the objects found by the vision application, you now need to create or add a V+ program.

For this tutorial we have provided a sample application that instructs the robot to pick up a model-defined object on a conveyor belt.

![Adding and assigning the Micro V+ program](image)

**Figure 73** Adding and assigning the Micro V+ program

1. In the Adept DeskTop window, Open the **Code Library** tab. If it is not visible, open the Code Library from the menu: select **View > Code Library**.

2. In the list of code examples, also called clips, select **Vision-guided conveyor tracking** from the **AdeptSight Examples** folder. See Figure 73.

3. Right-click on **Vision-guided belt tracking** and select **New Program**.

4. In the **New Program** window, click **Create**.

5. This adds the belt-demo and other dependent programs to the **Program Manager**.

6. Important: Before continuing, in the new program, configure the value of **$myip** to the IP address of the vision server (the PC on which AdeptSight is running.)

You must now assign the program to a task in the program execution tool.

1. In the **Program Manager** list, select the belt Demo program. See Figure 73.

2. Drag **belt_demo** to Task 0 in the **Task Manager** list.

3. Execute the program.

**You have completed the tutorial!**

Continue learning about AdeptSight in the following tutorials:

- AdeptSight Pick-and-Place Tutorial
- AdeptSight Standalone C# Tutorial
AdeptSight Upward-Looking Camera Tutorial

This tutorial will walk you through the creation of a basic vision application on a system in which the camera faces upwards, and in which the robot picks up objects and moves them to the camera. In this example, once an object is recognized by the vision applicator, the robot places the object in a pallet-type array.

This tutorial assumes you have a working knowledge of Adept robotic systems, V+, and Adept DeskTop. If you are new to AdeptSight, we recommend that you start with the Getting Started with AdeptSight tutorial.

System Requirements for this Tutorial

- A PC running AdeptSight and Adept DeskTop software.
- The camera provided with AdeptSight, or another DirectShow-compatible IEEE 1394 camera.
- The camera must face upwards.
- An Adept robot controlled by a CX controller or a Cobra i-Series robot with AIB controller.
- An end-effector that can pick up and move objects into the camera field of view.

Tutorial Overview

- Start an AdeptSight Vision Project
- Calibrating the Camera
- Connect to the Controller
- Assign the Robot to the Camera
- Calibrate the Vision System to the Robot
- Create a Vision Sequence
- Add the Acquire Image Tool
- Configure Latching Parameters
- Add a Locator Tool
- Create a Model
- Calibrate the Place Location for the Model
- Integrate AdeptSight with a V+ Program

Before Starting the Tutorial

You will need an object that you will pick with the robot. This object will be also be used to calibrate the system, during the Object-Attached-To-Robot Calibration Wizard.
Before starting this tutorial you should:

1. Install the camera. Make the required connections between the camera and controller to enable belt encoder latching. Consult the camera documentation for more information.

2. Install the software.

Please refer to the *Getting Started with AdeptSight* tutorial or the AdeptSight Online Help for assistance any of these preliminary steps.
Start an AdeptSight Vision Project

You will now create the vision project for this tutorial in the AdeptSight Vision Project manager window.

Opening AdeptSight

1. From the Adept DeskTop menu, select View > AdeptSight.
2. AdeptSight opens in the Vision Project window, similar to Figure 74.

![Figure 74 The Vision Project manager window](image)

Create and Name the New Sequence

You will now create and name the vision sequence that you will use for this tutorial.

1. By default, there is a blank vision sequence named NewSequence in the Vision project.
   
   If the list is empty, create a new sequence by clicking the 'Create Project" icon:

2. Left-click once on 'NewSequence' to edit the name.
3. Name the sequence UpwardFacing. The project now contains one vision sequence named UpwardFacing.

4. Click the 'Save Project" icon to save the vision project now:
5. Save the project as UpwardFacing.hsproj.

![Figure 75 Renaming a new vision sequence](image)

Next you will verify the camera that you will use for the application.
AdeptSight Upward-Looking Camera Tutorial - Start an AdeptSight Vision Project

Next: Calibrating the Camera
Calibrating the Camera

When the camera is correctly installed and recognized by the system it appears in the **System Devices** manager, in the **Cameras** tab, as shown in Figure 76.

If you have not previously calibrated the camera, a warning symbol appears to the right of the camera **State** icon, as shown in Figure 76.

**Choose a camera calibration method:**

1. **Calibrate the camera now**, by launching the 2D Vision Calibration Wizard from the toolbar. This requires a "grid of dots" calibration target. Sample calibration targets are provided in the AdeptSight support files, in the AdeptSight installation folder: `...\AdeptSight 2.0\Tutorials\Calibration`.

2. **Calibrate the camera later**, through the vision to robot calibration, as explained later in this tutorial. This will provide acceptable accuracy in most cases. However, a separate vision calibration can provide increased accuracy to your application.

  Calibrating the camera only through the vision-to-robot calibration will not correct for lens distortion. In some cases, strong lens distortion may cause the vision-to-robot calibration to fail if you do not calibrate the vision first.

  For more details on this subject, see [Why is Vision Calibration Important?](#)

**Next:** [Calibrating the Camera With the 2D Vision Calibration](#) or skip and continue to [Connect to the Controller](#)
Calibrating the Camera With the 2D Vision Calibration

To calibrate an upward-looking camera the calibration target must be placed at the end of the robot end-effector, parallel to the camera.

**To correctly position a calibration target for the camera calibration:**

1. Fix the target at the end of the end effector so that it is stable and parallel to the camera field of view, with the calibration pattern facing towards the camera.

2. Make sure the *camera-to-target* height is equal to the *camera-to-object* height. The *camera-to-object* height is the distance between the camera and an object at the moment that the camera takes an image of the object.

**Before starting the calibration make sure:**

1. You have available one or more objects that will be used as calibration objects.

2. You know the *camera-to-object* height that will be used during the real-time application for which you are calibrating the system.
   
   The *camera-to-object* height that you will set during the calibration must be the same as the distance that will be used during the real-time application.

![Camera-to-Target Distance for Calibrating an Upwards-Facing Camera](image)

**To start the camera calibration:**

1. In the **Cameras** tab of the System Devices manager, select the camera you want to calibrate in the Devices list.

2. Click the 'Calibrate Camera' icon:

3. The 2D Vision Calibration Wizard opens, beginning the vision (camera) calibration process.
4. Follow the instructions in the wizard, then return to this tutorial once the calibration is finished.
5. If you need help during the Calibration process, Click Help in the Calibration Wizard.

Next:  Connect to the Controller
Connect to the Controller

You will now start to set up the devices that will be used by the vision guidance application.

**Adding the Controller from Adept DeskTop**

If you are using AdeptSight from within Adept DeskTop, a controller device is present in the **Controllers** tab, as shown in Figure 78. You must connect to the controller to continue setting up this application.

If you have a multiple-controller license, or are creating the application outside the Adept DeskTop environment you may have to add a controller in the **Controllers** tab. Consult the Adept DeskTop online help for assistance in adding a controller.

**Connecting to the Controller**

1. From the Adept DeskTop menu, select **File > Connect...**
2. Connect to the controller. Refer to the Adept DeskTop online help if needed.
3. When the controller is connected, the **State** icon for the controller becomes green, and a robot is attached to the controller, as shown in Figure 79.

**Next:** Assign the Robot to the Camera
Assign the Robot to the Camera

You must now assign to the camera the robot that will be used for the vision application. Later in this tutorial you will calibrate the camera and robot together in the Vision-to-Robot calibration.

1. Select the **Cameras** tab.
2. Select the camera you will use for the vision guidance application.
3. In the **System Devices** toolbar, select the 'Add Robot' icon.
4. The **Select a Robot** window opens as shown in Figure 80.

   ![Figure 80](image)

   **Figure 80** Assigning a Robot to the Camera

5. From the list, select the robot that you will use for the vision guidance application and click **OK**.
6. The robot is now assigned to the selected camera in the Devices List, as shown in Figure 81.

   ![Figure 81](image)

   **Figure 81** Robot Assigned to the Camera

You will now need to calibrate the system using a Vision-to-Robot calibration wizard.

**Next:** [Calibrate the Vision System to the Robot](#)
Calibrate the Vision System to the Robot

In this Module you will calibrate the system. To calibrate this application you will use the Object-Attached-to Robot Calibration Wizard.

To start the calibration wizard:

1. In the System Devices manager, select the Cameras tab.
2. In the list of devices, select the robot that is assigned to the camera (Robot1.)
3. Click the 'Calibrate Vision to Robot' icon:
   
4. The Calibration Interview Wizard begins, beginning the Vision-to-Robot calibration process.
   
   Questions in the Interview Wizard determine the type of calibration required for your system

   In the Calibration Interview Wizard, make sure you the options that specifies that the calibration object
   will be attached to the robot tool (end-effector).

   Figure 82  Selecting Object-Attached-To-Robot options in the Calibration Interview Wizard

To select the Object-Attached-to Robot Calibration Wizard:

1. In the Calibration Interview Wizard, at the Choose Interview Mode step, select: I wish to select the correct calibration options from a list.
2. Under section 1, select: The camera is field-mounted relative to the robot base.
3. Under section 2, select: The workspace is not a conveyor belt.
4. Under section 3, select: The calibration object will be attached to the robot tool.
5. Under section 4, select: The robot is equipped with a tool that can pick up and move an object.
6. Under section 5, select the appropriate option.
   
   • If the robot is able to move freely in the workspace, an Automated calibration will be selected.
   
   In this case the robot will automatically carry out part of the calibration.
• Otherwise, a Manual calibration will be selected. In this case you need to manually move the robot to various parts of the camera field of view to gather calibration data.

7. Under section 6, select the appropriate option.

• If the end-effector is centered on the tool flange, the calibration process will be quicker because the calibration wizard will not have to calculate offset between the end-effector and the robot tool flange.

• If the end effector is not centered on the tool flange, the calibration process will have to correct for the offset between the end-effector gripper and the robot tool flange. If the calibration is a Manual calibration (see step 6.), you will have to manually move the robot to gather the several calibration points required for the offset correction.

Using the Object-Attached-to Robot Calibration Wizard

1. Follow the instructions in the wizard, then return to this tutorial once the calibration is complete.

2. If you need help during the Calibration process, click the Help button in the Calibration Wizard.

Next: Create a Vision Sequence
Create a Vision Sequence

A sequence is a series of tasks that are executed by vision tools. When you execute a sequence, each tool in the sequence executes in order. You add, remove, and edit the vision tools in the **Sequence Editor**.

**Saving a Sequence**

All sequences in the Sequence Manager are saved when you save the vision project.

- Sequences are saved as part of the project, not individually.
- Project files are saved with the extension "hsproj".

Click the 'Save Project' icon to save changes you have made up to now.

**Opening the Sequence Editor**

**To open the Sequence Editor:**

1. In the Sequence Manager, select the *UpwardFacing* sequence.
2. In the toolbar click the 'Edit Sequence' icon:
3. The Sequence Editor opens, similar to *Figure 83*.

![Sequence Editor - Conveyor Tracking](image)

*Figure 83* New vision sequence in the Sequence Manager

In this tutorial you will add the following tools to the sequence:

- **Acquire Image**: This tool acquire images from the camera and outputs the Acquired images for use by other tools in the sequence.
- **Locator**: The Locator will find an object in input images, based on the object model that you will create for the object.

Next:  *Add the Acquire Image Tool*
Add the Acquire Image Tool

The **Acquire Image** tool is the first tool to add because it supplies images to other tools in the sequence.

The **Acquire Image** tool will provide the images taken by the camera to the Locator tool and define the latching parameters for the conveyor belt.

**To add the Acquire Image tool:**

1. In the **Toolbox**, select **Acquire Image** and drag it into the Process Manager area that reads 'Drop Tools Here'.
2. The Process Manager (blue area) now contains the **Acquire Image** tool. See Figure 84.
3. You are now ready to acquire images.

**To display acquired images:**

1. In the toolbar, click the 'Execute Sequence' icon. Alternatively, you can execute the only the Acquire Image tool by clicking the 'Execute Tool' icon:

![Figure 84](image)

**Figure 84** Live Display of camera images in the Sequence Editor

You will now configure latching parameters.

**Next:** [Configure Latching Parameters](#)
Configure Latching Parameters

You must now set a robot-latching parameter in the Acquire Image Tool. This latching parameter enables the V+ to latch the location of the robot when an image is grabbed.

**To set the latching parameters:**

1. Under **Latching Parameters** there are two sections **Robot** and **Belt**.

2. Under **Robot** enable **Read Value** if:
   - The robot comes to a full stop when the camera is taking an image, and/or when the robot movement is very slow and a slight error in the location is not critical to the application.
   - This latching mode *does not* require a cable for latching the signal.

3. Under **Robot** enable **Read Latched Value** if:
   - This robot in continuous movement when it passes the inspected object through the camera field of view (on-the-fly inspection).
   - This latching mode *requires* a cable for latching the signal.

![Latching Parameters of the Acquire Image tool](image)

**Figure 85** Latching Parameters of the Acquire Image tool

You will now add the Locator tool to your application.

Next: Add a Locator Tool
Add a Locator Tool

The Locator tool will search for the objects you have defined in your application and returns results on the location of the objects it finds.

![Locator Tool](Figure 86 Locator Tool added to the vision sequence)

**To add the Locator tool:**

1. In the Toolbox, select **Locator** and drag it into the Process Manager frame, below the **Acquire Image** tool, as shown in Figure 86.

2. Under **Location**, leave the **Entire Image** check box enabled. This ensures that the search process will look for objects in the entire image provided by the camera.

You are now ready to create a model for the object that will be handled by the application.

Next: Create a Model
Create a Model

To find a specific part or object, AdeptSight must have an active model of the part. You will now create the model for the part you want to locate with this application.

![Figure 87: Basic Model-Edition Mode Provides Quick Model-Building](image)

To create a model:

1. With the robot, grip the object that you will handle with this application.
2. Move the robot so that the object is in the field of view of the camera.
3. Execute the sequence to acquire an image, by clicking 'Execute Sequence' icon.
4. In the **Models** section, click the ‘+’ icon to create a new model. The display is now in Model Edition mode as illustrated in Figure 87.
5. Drag and resize the green bounding box to completely enclose the entire object. Green outlines show the features that have been selected to add to the model.
6. Drag and rotate the yellow axes marker to position the coordinate system of the model.
7. If you need to edit the model, for example to add or remove features, click **Expert** to enter the Expert Model Edition mode. Refer to the online User Guide or the 'Getting Started' tutorial if you need help for this feature.
8. Click **Done** to complete the creation of the model. This new model now appears as **Model0** in the list of **Models**.

**Next:** Calibrate the Place Location for the Model
Calibrate the Place Location for the Model

For each object model, you must carry out a place location calibration that will enable the application to correctly place the part in its final place location after it has been found. This is done through the Place Location Wizard.

- The Place Location Wizard teaches AdeptSight the location in the workspace where the robot will place the object after it has been correctly found and identified in the camera image.
- If you do not carry out the Place Location calibration, the robot may not be able to precisely and accurately place the part in its correct location. You will need to manage and define the place location with a V+/MicroV+ program.
- This calibration must be carried out at least once for each model.
- You must recalibrate the place location for a model if the model frame of reference is modified or the camera-robot setup is modified.

To start the Place Location Wizard:

1. Select the model from the list of models.

2. Select the 'Model Options' icon: 

3. From the menu, select **Gripper Offset > Manager** as shown in Figure 88. This opens the Gripper Offset Manager shown in Figure 89.

4. In the Gripper Offset Manager, select the wizard icon, as shown in Figure 89.

5. This will automatically choose the Place Location Wizard because AdeptSight will detect that the system was calibrated for an upwards-facing camera.
Launch Gripper Offset Wizard from here

Figure 89  Gripper Offset Manager
Carrying out the Place Location Calibration

The Place Location Wizard walks through the steps required for assigning a place location to a Model.

1. Follow the instructions in the Wizard to complete the process. Click 'Help' for assistance during the calibration.

2. Once the calibration is complete, the Place Location is added to the Gripper Offset Manager.

3. Click Close to return to the Locator tool interface. A check mark icon indicates that the calibration has been completed for the model.

4. Repeat the Place Location Wizard for each model you create.

Before starting the Gripper Offset Wizard: Make sure you have an object that is identical to the object used to create the model.

Next: Integrate AdeptSight with a V+ Program
Integrate AdeptSight with a V+ Program

To enable the robot to handle the objects found by the vision application and test the application you will need, you now need to create or add a V+ program that will run the application.

Refer to the AdeptSight Pick-and-Place Tutorial for information on integrating this application with V+. The program used for the Pick-And-Place tutorial can be adapted to the upwards facing application by switching the OPEN and CLOSE commands.

You have completed the tutorial!

Continue learning about AdeptSight in the following tutorials:

- AdeptSight Pick-and-Place Tutorial
- AdeptSight Standalone C# Tutorial
AdeptSight Standalone C# Tutorial

Welcome to the AdeptSight Standalone C# Tutorial.

This tutorial will guide you through the development of a standalone AdeptSight application in Visual C#. As you follow the steps for the tutorial you will build an object location application to which you will add and configure a full range of inspection tools.

This tutorial uses as an example a system with Microsoft Visual Studio 2005 and AdeptSight 2.0.

Tutorial Overview

- Create the AdeptSight Vision Project
- Build the Program Interface
- Add Basic Code to the Application Form
- Create a Vision Inspection Sequence
- Create a Model with the Locator Tool
- Add Code for the Locator
- Test the Application
- Add and Configure a Display Interface
- Add a Caliper Tool
- Add Code for the Caliper
- Add a Blob Analyzer Tool
- Add Code for the Blob Analyzer
- Add a Pattern Locator Tool
- Add Code for the Pattern Locator
- Add Two Edge Locator Tools
- Add Code for the Edge Locators

System Requirements

- PC running Windows 2000 SP4 or Windows XP SP2
- Microsoft Visual Studio 2005

The type of PC processor will influence the execution speed of the vision applications. This tutorial presumes you have a basic beginner’s knowledge of Microsoft Visual C#.
Create the AdeptSight Vision Project

This first step shows you how to build a basic AdeptSight application that will locate a model-defined object, at whatever angle and displacement the object appears.

Creating the Project

In this section, you will create the Project of the application, add basic lines of code to interact with the interface, and add an AdeptSight VisionProjectControl, which you will edit to build the application.

Constructing the Project

You will now build the project that will allow you to specify which type of program you want.

2. Create a new Visual C# Windows Application (exe) project.
3. Name it HookInspection and specify a working folder in the location field.
4. Click OK.

You can already compile this basic project, but you must add some lines of code and create the interface for your application to make this a useful project.

Next: Build the Program Interface
Build the Program Interface

In this step you will build the interface that will allow you to interact with the application and visualize results.

![Visual C# Hook Inspection Application Form](image)

**Figure 90** Visual C# Hook Inspection Application Form

1. Click on the Solution Explorer tab, select file named Form1.cs and rename it to HookInspection.cs. Then double-click on the file to edit the form template.

2. Resize the form template to approximately 810 X 570.

3. On the form template, remove all existing dialog items and add the appropriate controls to make the form template look like the one in Figure 90.

4. The top left control is the main AdeptSight control. It is called the VisionProjectControl. It can be added to the form by following these steps:
   a. From the Toolbox context menu, select the Add/Remove Items ... command.
   b. From the .NET Framework Components tab, select VisionProjectControl component.
   c. Click OK to accept selection.
   d. Select VisionProjectControl from the Toolbox and paste one instance on the form.
   e. From Properties, rename the newly created control to mVisionProjectControl.

5. The top right control is also an AdeptSight control. It is called the Display. It can be added to the form by following these steps:
   a. From the Toolbox context menu, select Add/Remove Items ... command.
   b. From the .NET Framework Components tab, select the Display component.
   c. Click OK to accept the selection.
d. Select **Display** from the Toolbox and paste one instance on the form.

e. From **Properties**, rename the newly created control to **mDisplay**.

6. The other components on the form are standard **GroupBox, Label, TextBox, CheckBox** and **Button** components, pasted from the Toolbox. Below is the list of controls you need to add, and the name to give to each control.

<table>
<thead>
<tr>
<th>Control to add</th>
<th>Name of the control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>mType</td>
</tr>
<tr>
<td>Scale</td>
<td>mScale</td>
</tr>
<tr>
<td>Rotation</td>
<td>mRotation</td>
</tr>
<tr>
<td>Translation X</td>
<td>mTranslationX</td>
</tr>
<tr>
<td>Translation Y</td>
<td>mTranslationY</td>
</tr>
<tr>
<td>Width</td>
<td>mWidth</td>
</tr>
<tr>
<td>Height</td>
<td>mHeight</td>
</tr>
<tr>
<td>Diameter</td>
<td>mDiameter</td>
</tr>
<tr>
<td>Offset</td>
<td>mOffset</td>
</tr>
<tr>
<td>Part Label</td>
<td>mPartLabel</td>
</tr>
<tr>
<td>Part Width</td>
<td>mPartWidth</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>mTime</td>
</tr>
<tr>
<td>Continuous Mode</td>
<td>mCheckContinuous</td>
</tr>
<tr>
<td>Execute Inspection</td>
<td>mExecuteButton</td>
</tr>
</tbody>
</table>

7. The interface is now complete. Save your work.
Add Basic Code to the Application Form

You need to add basic code to be able to interact with your application. As you will add new features throughout the other tutorial sections, you will also add lines to this code.

2. From Properties, rename form to HookInspectionForm.
3. Add a Closing event handler by double clicking to the right of event from Events view.
4. Modify created event handler code to look like this:

   ```csharp
   // <summary>
   /// Form Closing event handler.
   /// </summary>
   private void HookInspectionForm_Closing(
   object sender, System.ComponentModel.CancelEventArgs eventArguments
   )
   {
       WaitForCompletion();
   }
   ```

5. Add the private method, shown below, that will be called at appropriate places in the code to ensure continuous mode deactivation, in the case of a critical operation, such as an application closing event.

   ```csharp
   // <summary>
   /// Utility method to ensure continuous mode stopping before any critical operation.
   /// </summary>
   private void WaitForCompletion()
   {
       if ( mCheckContinuous.Checked )
       {
           mCheckContinuous.Checked = false;
       }
   }
   ```

7. Select the 'Execute Inspection' button.
8. Add a Click event handler by typing ExecuteButton_Click in the edit field to the right of event from Events view.
9. Modify the created event handler code to look like this:

   ```csharp
   /// <summary>
   /// Execute button event handler.
   /// </summary>
   private void ExecuteButton_Click(object sender, System.EventArgs e)
   {
       try
   ```
{ 
    mExecuteButton.Enabled = false;
    do
    {
        mVisionProjectControl.VisionProject.Sequences[0].Loop = false;
        mVisionProjectControl.VisionProject.
        Sequences[0].Execute();
        Application.DoEvents();
    }
    while ( mCheckContinuous.Checked );
    mExecuteButton.Enabled = true;
} 

Next: Create a Vision Inspection Sequence
Create a Vision Inspection Sequence

To create and name the vision sequence that you will use for this tutorial, do the following:

1. Run the application.
2. From **VisionProjectControl** interface, rename first sequence to **Hook Inspection**.
3. The project now contains one vision sequence.
4. Click the 'Save Project' icon to save the vision project now. Save the project as **HookInspection.hsproj**.
5. Click the 'Project Properties' icon to change AdeptSight environment settings.
6. From the **Startup** tab, enable **Auto Load Project** and select the previously saved project.
7. Click **Close** to accept changes and quit the application.

You have now created the vision project file for this tutorial. The application has been configured to automatically load the vision project when the application starts.

In this step you will start adding the necessary tools to perform a vision inspection.

1. Restart the application
2. In the **Sequence Manager**, select the Hook Inspection sequence.
3. In the toolbar, click the 'Edit Sequence' icon to start the Sequence Editor.
4. The Toolbox contains the tools available for building sequences.
5. Select **Acquire Image** and drag it into the frame that reads 'Drop Tools Here'.
6. Select **Emulation** and click the 'Camera Properties' icon to show the **Emulation Properties** dialog.
7. Load the **Hook.hdb** emulation database from the Tutorial/Images folder, in the distributed files from the AdeptSight installation.
8. The Sequence Editor should now look like **Figure 91**.
9. Click **Done** to accept changes and close the Sequence Editor.
10. Click the 'Save Project' icon to save the vision project now.
You have now begun building the vision inspection sequence. The first step was to add a tool for image acquisition.

The next step is to create a model from an acquired image and then use this model to locate all instances of a specific object in any acquired image.

Next: Create a Model with the Locator Tool
Create a Model with the Locator Tool

In this step you will add a Locator tool to the sequence and build a model of an object.

**Add the Locator**

1. Double-click on the **Hook Inspection** sequence to start the Sequence Editor.
2. Click the 'Execute Sequence' icon to execute the sequence once and acquire a first image.
3. The Toolbox contains the tools available for building sequences.
4. From the Toolbox, select **Locator** and drag just below the **Acquire Image** tool.
5. Under **Location**, leave the **Entire Image** check box enabled to ensure that the Locator will search the entire image.

You are now ready to create a model for the object that you want to find with this application.

**Create a Model**

1. In the Models section, click the '+' icon to create a new model.
2. The display is now in Model Edition mode.
3. The Model's bounding box appears in the image as a green rectangle.
4. Drag and resize the green bounding box to completely enclose the entire object. Green outlines show the contours that currently represent the model.
5. Drag and rotate the yellow axes marker to position the coordinate system. The Sequence Editor should now look like [Figure 92](#).
6. Click **Done** to accept new model changes.
7. Rename the newly added model from **Model0** to **Hook**.
8. Close the Sequence Editor.
9. Click the 'Save Project' icon to save the vision project.

![Figure 92 Creating a Model in the Model Edition Mode](image-url)
You have now completed creation of the tools required for this first step of the tutorial. Before continuing to add code, run the application once and execute it on all images from emulation database to verify that Locator tool appropriately locates all instances in all images.

To see the Locator results in the Sequence Editor display and the results grid, simply select the tool as the active one by clicking in its title bar.

Next:  Add Code for the Locator
Add Code for the Locator

In this section, you will add code to output the properties of the instance found by the Locator in your application interface. But before adding any lines of code, appropriate reference must be added.

1. From the Solution Explorer, select the Add Reference … command from context menu.
2. From the displayed dialog, click on Browse … button.
3. Move to [Common Files]\Adept Technology\AdeptSight\PlugIns\Tool.
4. Select LocatorPlugIn.dll file and click OK twice to accept adding new reference.
5. Select the newly added reference and from Properties, change Copy Local to false.

Now that appropriate references have been added, code referencing the Locator can be added.

1. Select HookInspection.cs window.
2. At the top, add appropriate 'using' directives, as follows:

   using Adept.iSight.Tools;
   using Adept.iSight.Forms;

3. Locate the ExecuteButton_Click method and insert the lines of code shown in bold:

   ```csharp
   try
   {
   Locator lLocator = null;
   mExecuteButton.Enabled = false;
   The above code simply defines a new variable ready to reference a Locator tool.
   }
   ```

4. In the existing do loop, add the lines of code shown in bold:

   ```csharp
   do
   {
   mVisionProjectControl.VisionProject.Sequences[0].Loop = false;
   mVisionProjectControl.VisionProject.Sequences[0].Execute();
   // Retrieving / Showing Locator results
   lLocator = mVisionProjectControl.VisionProject.Sequences[0][1] as Locator;
   if ( lLocator.GetInstanceCount( 0 ) > 0 )
   {
   // An instance of the object is found
   // Output the properties of the located instance
   mType.Text = lLocator.GetInstanceModelName(0,0);
   mScale.Text = lLocator.GetInstanceScaleFactor(0,0).ToString( "0.00" );
   mRotation.Text = lLocator.GetInstanceRotation(0,0).ToString( "0.00" );
   mTranslationX.Text = lLocator.GetInstanceTranslationX(0,0).ToString( "0.00" );
   mTranslationY.Text = lLocator.GetInstanceTranslationY(0,0).ToString( "0.00" );
   } 
   else
   {
   // No instance of the object found
   } 
   ```
mType.Text = "";
mScale.Text = "";
mRotation.Text = "";
mTranslationX.Text = "";
mTranslationY.Text = "";
}

// Retrieving / Showing sequence execution timing
mTime.Text = mVisionProjectControl.VisionProject.Sequences[0].ElapsedTime.ToString("0.00 ms");
Application.DoEvents();
}
while ( mCheckContinuous.Checked );

This code first creates a reference to the Locator tool to enable specific programmatic access. From this reference, results are retrieved and displayed in corresponding controls. Finally, execution time is retrieved to show current execution time before next execution occurs.

5. Coding is now completed. Save your work and move on to the next step.

Next: Test the Application
Test the Application

You are ready to test your application.

1. To start the running mode, press the F5 key. Click **Execute Inspection** a few times. The properties of the found instance should be updated after each inspection.

2. Enable the **Continuous Mode** check box and click **Execute Inspection**. The application should run in continuous mode. The application should now look like Figure 93.

![Figure 93 Application Interface after first execution of the vision sequence](image)

This concludes this tutorial step. After debugging, save your work and move on to the next tutorial step where you will add a display to your application.

**Next:** [Add and Configure a Display Interface](#)
Add and Configure a Display Interface

This second step shows you how to interface with the Display in your application interface. A Display allows you to view Images and Scenes processed by an AdeptSight Vision Sequence. In the previous step, you have already added a Display object to your application interface.

Adding Code for the Display

Code must now be added to update the Display output every time the inspection loop execute.

1. Locate the **ExecuteButton_Click** method and insert the lines of code shown in bold:

```csharp
try
{
    Locator lLocator = null;
    mDisplay.Markers.Clear();
    mExecuteButton.Enabled = false;
}
```

This code simply removes all overlay graphic markers that could have been added to the Display from a previous execution.

2. In the existing **do** loop, add the lines of code shown in bold:

```csharp
do
{
    mVisionProjectControl.VisionProject.Sequences[0].Loop = false;
    mVisionProjectControl.VisionProject.Sequences[0].Execute();

    // Showing input Image in Display
    mDisplay.Images[0].SetImageDatabase(
        mVisionProjectControl.VisionProject.Sequences[0].Database.Handle);
    mDisplay.Images[0].SetImageViewName(
        mVisionProjectControl.VisionProject.Sequences[0][0].Name);
    mDisplay.Images[0].SetImageName( "Image" );

    // Retrieving / Showing Locator results
    lLocator = mVisionProjectControl.VisionProject.Sequences[0][1] as Locator;
    if ( lLocator.GetInstanceCount( 0 ) > 0 )
    {
        // An instance of the object is found
        // Output the properties of the located instance
        mType.Text = lLocator.GetInstanceModelName(0,0);
        mScale.Text = lLocator.GetInstanceScaleFactor(0,0).ToString( "0.00" );
        mRotation.Text = lLocator.GetInstanceRotation(0,0).ToString( "0.00" );
        mTranslationX.Text = lLocator.GetInstanceTranslationX(0,0).ToString( "0.00" );
        mTranslationY.Text = lLocator.GetInstanceTranslationY(0,0).ToString( "0.00" );

        // Showing instance scene in Display
        mDisplay.Scenes[0].SetSceneDatabase(
```
This code first shows how to setup the Display to show image provided by the Acquire Image tool. Then appropriate Display settings are modified to show the **Output Instance Scene** provided by the Locator tool. Finally, a **RefreshDisplay** call is issued to show current selections before next execution occurs. Details about Display capabilities can be found in the documentation.

3. Coding is now completed. If you execute the application, you should now see a image in which the current instance is highlighted. Save your work and move on to the next step.

Next: Add a Caliper Tool
Add a Caliper Tool

In this tutorial step, you will learn how to set up, configure and use a Caliper tool to precisely measure the distance between parallel edges on an object.

![Figure 94 Positioning the Caliper Tool](image)

Positioning the Caliper

In the previous step, you created the vision project file for this tutorial. The application has been configured to automatically load the vision project when the application starts.

In this step you will modify the vision sequence to include a Caliper tool.

1. Restart the application.
2. In the Sequence Manager, select the Hook Inspection sequence.
3. In the toolbar, click the 'Edit Sequence' icon to start the Sequence Editor.
4. Execute once by clicking the 'Execute' icon.
5. From the context menu (in the frame where tools are created), select Add > Caliper.
6. Double-click on the tool title and rename tool from 'Caliper' to 'Width Caliper.'
7. Acquire Image is automatically selected as the tool that will provide the Input image.
8. Under Location, select Locator as Frame Input and click on the Location button.
9. In the Location dialog, set the frame-based position of the tool as shown in Figure 94. You can either enter values manually or edit the location bounding box in the display, with the mouse.

The Caliper detects edges that are parallel to its Y-axis: adjust the rotation to best match the inclination of the edges you want to measure.

10. Click OK to apply the Location parameters and execute the tool once by clicking the 'Execute Tool' icon.
Configuring the Caliper

You are now ready to set properties for the pair of edge you want to identify.

1. Under Pairs, rename default created pair from Pair0 to Width Measurement. Double click on 'Pair0' to enable renaming.

2. Click on Edit to change pair properties.

3. Set First Edge Polarity to Light to Dark and Second Edge Polarity to Dark to Light

4. Enable Position Constraint for both edges.

5. Set edge position constraints with the bottom controls, as shown in Figure 95.

6. Click OK to apply pair properties and return to the Sequence Editor

7. Leave default values for all other Caliper parameters.

8. Close the Sequence Editor.

9. Click the 'Save Project' icon to save changes made to the tutorial vision project.

---

The Caliper must always be as perpendicular as possible to the edges to be measured. For the current example, the tool had to be rotated to 90 degrees and the skew was left at its default value of 0 degrees.

---

**Figure 95** Setting Edge Pair properties for the Caliper tool
Now that the Caliper is properly configured to measure the rectangular hole feature, you will want to test and observe the Caliper on other instances of the Object. Code must now be added to your application in order to highlight Caliper results.

Next: Add Code for the Caliper
Add Code for the Caliper

You will complete this tool up by adding code to your Visual C# application that will display the measurement of the Caliper in the interface. You will also add a line marker on the display to show the measurement.

1. From the **Solution Explorer** context menu, select the **Add Reference ...** command.
2. In the displayed dialog, click the **Browse ...** button.
3. Move to `[Common Files]\Adept Technology\AdeptSight\PlugIns\Tool`. 
4. Select `CaliperPlugIn.dll` file and click **OK** twice to accept adding new reference.
5. Select newly added reference and from **Properties**, change **Copy Local** to **false**.
6. Now that the appropriate reference has been added, code referencing the Caliper can be added.
7. Select **HookInspection.cs** window.
8. Locate the **ExecuteButton_Click** method and insert the lines of code shown in bold:

```csharp
try {
    Locator lLocator = null;
    Caliper lWidthCaliper = null;
    mExecuteButton.Enabled = false;
    This code simply defines a new variable ready to reference a Caliper tool.

    ... // Retrieving / Showing Calipers results
    lWidthCaliper = mVisionProjectControl.VisionProject.Sequences[0][2] as Caliper;
    if ( lLocator.GetInstanceCount( 0 ) > 0 && lWidthCaliper.GetPairScore( 0, 0 ) > 0 ) {
        // Output the caliper results
        mWidth.Text = lWidthCaliper.GetPairSize( 0, 0 ).ToString( "0.00" );
        // Showing width marker in Display
        MarkerLine lMarker = new MarkerLine(
            "Width",
            lWidthCaliper.GetEdge1PositionX( 0, 0 ),
            lWidthCaliper.GetEdge1PositionY( 0, 0 ),
            lWidthCaliper.GetEdge2PositionX( 0, 0 ),
            lWidthCaliper.GetEdge2PositionY( 0, 0 ),
            true );
        mDisplay.Markers.Add( lMarker );
        lMarker.AnchorStyle = MarkerAnchorStyle.CROSS;
    }
}
```

```csharp
lMarker.Constraints = LineMarkerConstraints.LineNoEdit;
lMarker.PenWidth = PenWidth.Thin;
lMarker.Color = HSColor.Red;
}
else
{
    mWidth.Text = "";
}

// Retrieving / Showing sequence execution timing
mTime.Text =

mVisionProjectControl.VisionProject.Sequences[0].ElapsedTime.ToString(
"0.00 ms" );
mDisplay.RefreshDisplay();
Application.DoEvents();
} while ( mCheckContinuous.Checked );
```

This code first creates a reference to the Caliper tool to enable specific programmatic access. From this reference, results are retrieved and displayed in corresponding controls. If the first pair of the Caliper (Index = 0) is found (Score > 0.250), its measurement is displayed. Using the World coordinate system, the calibrated X-Y position of the first and second edge of the pair is used to draw a non-editable line marker on the Display.

10. Coding is now completed. Save your work and test the application. You should see the measurement in the Inspection Width text box of your application. You should also see the line marker you added on the Display highlighting the measurement.

**Adding a Second Caliper**

Using the previous steps, try adding a second Caliper that will measure the height of the rectangular hole in the hook.

1. Add a second caliper tool and rename it *Height Caliper*.
2. In the Location dialog, place the caliper to measure the height of the rectangular hole in the hook.
3. Rename Pair0 of the second caliper to *Height Measurement*.
4. Edit the edge pair settings similarly to first caliper tool (Width Measurement pair).
5. Add the appropriate code to display the result in the appropriate text box on your application form. Also add the code to draw another line marker on the Display.

**Testing the Application**

You are ready to test your application.

1. To start the running mode, press the F5 key. Click **Execute Inspection** a few times.
2. The measurements on the found instance should be updated after each inspection.
3. Enable the **Continuous Mode** check box and click **Execute Inspection**.

4. The application should run in continuous mode.

5. The application should now look like **Figure 96**.

![Application Interface showing Width and Height Measurements](image)

**Figure 96** Application Interface showing Width and Height Measurements
Add a Blob Analyzer Tool

In this tutorial step, you will learn how to set up, configure and use a Blob Analyzer tool to find and analyze irregular shaped features on the part.

Positioning the Blob Analyzer

In the previous steps, you created and modified the vision project file for this tutorial. The application has been configured to automatically load the vision project when the application starts.

In this step you will modify the vision sequence to include a Blob Analyzer tool.

1. Restart the application.
2. In the Sequence Manager, select the **Hook Inspection** sequence.
3. In the toolbar, click the 'Edit Sequence' icon to start the **Sequence Editor**.
4. Execute once by clicking the 'Execute Tool' icon: 📈
5. From the context menu, select **Add > Blob Analyzer**.
6. Double-click on the tool title and rename tool from **Blob Analyzer** to **Hole Blob**.
7. **Acquire Image** is by default selected as the tool that will provide the **Input** image.
8. Under **Location**, select **Locator** as **Frame Input** and click on **Location** button.
9. Set the frame-based position of the tool as shown in **Figure 97**.

![Figure 97 Setting Location of the Blob Analyzer Tool](image)

Configuring the Blob Analyzer

You will now set up constraints that will allow the Blob Analyzer to find only valid blobs, that is, those that meet the criteria you have determined. These constraints can be set from Blobs parameters.

1. Under **Blobs**, click **Configure**.
2. Set **Minimum Area** to 0 and **Maximum Area** to 100.
3. For **Image Mode**, select the **Dark** segmentation mode.
4. In the histogram window, use the mouse to drag and set segmentation limits: top to 90 and bottom to 110 as shown in figure.

This instructs the process to keep pixels with a greylevel lower than 90 and reject those with a greylevel higher than 110. Pixels between these values will receive a weight between 0 and 1.

Alternatively you can set the segmentation limits in the Advanced Parameters grid by setting Segmentation Dark Top and Bottom to 90 and 110.

5. Click OK to close the Blob Settings window.

6. In the Advanced Parameters grid, under Results, select Object Coordinate System.

7. Keep default values for all other Advanced Parameters.

8. Close the Sequence Editor.

9. Click the 'Save Project' icon to save changes made to the tutorial vision project.

![Figure 98 Configuring Blob detection parameters](image)

Now that the Blob Analyzer is properly configured to inspect the circular hole feature, you will want to test and observe the Blob Analyzer results on other instances of the Object. Code must now be added to your application in order to highlight Blob Analyzer results.

Next: Add Code for the Blob Analyzer
Add Code for the Blob Analyzer

You will finish up by adding code to your Visual C# application that will display the measurements of the Blob Analyzer in the interface. You will also add a target marker on the display to show the position and diameter of the hole.

1. From the Solution Explorer, select the Add Reference … command from context menu.
2. From the displayed dialog, click on the Browse … button.
3. Move to [Common Files]\Adept Technology\AdeptSight\PlugIns\Tool.
4. Select BlobAnalyzerPlugIn.dll file and click OK twice to accept adding new reference.
5. Select newly added reference and from Properties, change Copy Local to false.
   Now that appropriate reference have been added, code referencing the Blob Analyzer can be added.
7. Locate the ExecuteButton_Click method and insert the lines of code shown in bold:
   try
   {
       Locator lLocator = null;
       Caliper lWidthCaliper = null;
       Caliper lHeightCaliper = null;
       BlobAnalyzer lHoleBlob = null;
       mExecuteButton.Enabled = false;
   }
   This code simply defines a new variable ready to reference a Blob Analyzer tool.
8. In the existing do loop, add the lines of code shown in bold:
   do
   {
      ...
   // Retrieving / Showing Blob results
   lHoleBlob = mVisionProjectControl.VisionProject.Sequences[0][4] as BlobAnalyzer;
   if ( lLocator.GetInstanceCount( 0 ) > 0 && lHoleBlob.GetBlobCount( 0 ) > 0 )
   {
      // Output the blob results
      double lDiameter = Math.Sqrt( 4 * lHoleBlob.GetBlobArea( 0, 0 ) / 3.14159 );
      mDiameter.Text = lDiameter.ToString( "0.00" );
      double lOffset = Math.Sqrt( lHoleBlob.GetBlobPositionX( 0, 0 ) * lHoleBlob.GetBlobPositionX( 0, 0 ) + lHoleBlob.GetBlobPositionY( 0, 0 ) * lHoleBlob.GetBlobPositionY( 0, 0 ) );
      mOffset.Text = lOffset.ToString( "0.00" );
      // Showing blob marker in Display
MarkerTarget lMarker = new MarkerTarget(
    "Hole",
    lHoleBlob.GetBlobPositionXWorld( 0, 0 ),
    lHoleBlob.GetBlobPositionYWorld( 0, 0 ),
    (float) (lDiameter / 2.0),
    true );
mDisplay.Markers.Add( lMarker );
lMarker.Constraints = TargetMarkerConstraints.TargetNoEdit;
lMarker.Color = HSColor.Red;
}
else
{
    mDiameter.Text = "";
    mOffset.Text = "";
}

// Retrieving / Showing sequence execution timing
mTime.Text = mVisionProjectControl.VisionProject.Sequences[0].ElapsedTime.ToString( "0.00 ms" );
mDisplay.RefreshDisplay();
Application.DoEvents();
}
while ( mCheckContinuous.Checked );

This code first creates a reference to the Blob Analyzer tool to enable specific programmatic access. From this reference, results are retrieved and displayed in corresponding controls. If the blob is found, the offset from the origin of the Object coordinate system and the diameter of the hole are computed and displayed. Using the World coordinate system, the calibrated X-Y position of blob is used to draw a non-editable target marker on the Display.

9. Coding is now completed. Save your work and test the application. You should see the measurement in the Inspection Diameter and Offset text boxes of your application. You should also see the target marker you added on the Display highlighting the hole measurements.

**Testing the Application**

You are ready to test your application.

1. To start the running mode, press the F5 key. Click **Execute Inspection** a few times.
   The properties of the found instance should be updated after each inspection.
2. Enable the **Continuous Mode** check box and click **Execute Inspection**.
3. The application should run in continuous mode.
4. The application should now look like **Figure 99**.
This concludes this tutorial step. After debugging, save your work and move on to the next tutorial step where you will add another inspection tool to your application.

Next: Add a Pattern Locator Tool
Add a Pattern Locator Tool

In this tutorial step, you will learn how to set up, configure and use a Pattern Locator tool to find and locate instances of the HS label printed on the part.

Positioning the Pattern Locator

In the previous steps, you have created and modified the vision project file for this tutorial. The application has been configured to automatically load the vision project when the application starts. In this step you will modify the vision sequence to include a Pattern Locator tool.

1. Restart the application
2. In the Sequence Manager, select the Hook Inspection sequence.
3. In the toolbar, click the 'Edit Sequence' icon to start the Sequence Editor.
4. Execute the sequence once by clicking 'Execute' icon.
5. From the context menu, select Add > Pattern Locator.
6. Double-click on the tool title and rename tool from Pattern Locator to Label Locator.
7. Acquire Image is by default selected as the tool that will provide the Input image.
8. Under Location, select Locator as Frame Input and click on Location button.
9. Set the frame-based position of the tool as shown in Figure 100.

![Figure 100](image)

Creating the Pattern Image

You will now create the pattern that you want the tool to search for in the defined Location area. The pattern can be defined from any input image.

1. Under Pattern, click Create to start a pattern creation process, from the current input image.
2. Set the location of the pattern as shown in Figure 101.

![Figure 101 Setting the Location of the Pattern Image.](image)

### Configuring Pattern Locator Constraints

You will now set up constraints that will allow the Pattern Locator to match pattern defined in defined frame-based location. These constraints can be changed in the Advanced Parameters property grid.

1. In **Advanced Parameters**, under **Results**, set **Coordinate System** to **Object**.
2. Under **Search**, set **Match Threshold** to **0.4**.
3. Keep default values for all other Advanced Parameters.
4. Close the Sequence Editor.
5. Click the 'Save Project' icon to save changes made to the tutorial vision project.

Now that the Pattern Locator is properly configured to check presence of the HS label, you will want to test and observe the Pattern Locator results on other instances of the Object. Code must now be added to your application in order to highlight Pattern Locator results

**Next:** Add Code for the Pattern Locator
Add Code for the Pattern Locator

You will finish up by adding code to your Visual C# application that will display the result of the pattern search in the interface. You will also add a point marker on the display to show the position of the label located by the Pattern Locator.

1. From Solution Explorer context menu, select the Add Reference ....
2. In the displayed dialog, click the Browse ... button.
3. Move to [Common Files]\Adept Technology\AdeptSight\PlugIns\Tool.
4. Select PatternLocatorPlugIn.dll file and click OK twice to accept adding new reference.
5. Select the newly added reference and in Properties, change Copy Local to false.
6. Now that appropriate reference have been added, code referencing the Pattern Locator can be added.
7. Select the HookInspection.cs window.
8. Locate the ExecuteButton_Click method and insert the lines of code shown in bold:

```csharp
try {
    Locator lLocator = null;
    Caliper lWidthCaliper = null;
    Caliper lHeightCaliper = null;
    BlobAnalyzer lHoleBlob = null;
    PatternLocator lLabelLocator = null;
    mExecuteButton.Enabled = false;

    // Retrieving / Showing Pattern results
    lLabelLocator = mVisionProjectControl.VisionProject.Sequences[0][5] as PatternLocator;
    if (lLocator.GetInstanceCount(0) > 0 && lLabelLocator.GetMatchCount(0) > 0)
    {
        // Output the pattern results
        mPartLabel.Text = "Present";
        // Showing pattern marker in Display
        MarkerPoint lMarker = new MarkerPoint("HSLabel",
            lLabelLocator.GetMatchPositionX(0, 0, false),
            lLabelLocator.GetMatchPositionY(0, 0, false),
            true);
        mDisplay.Markers.Add(lMarker);
    }
}
```

This code simply defines a new variable ready to reference a Pattern Locator tool.

9. In the existing do loop, add the lines of code shown in bold:

```csharp
do {
    ...
    // Retrieving / Showing Pattern results
    lLabelLocator = mVisionProjectControl.VisionProject.Sequences[0][5] as PatternLocator;
    if (lLocator.GetInstanceCount(0) > 0 && lLabelLocator.GetMatchCount(0) > 0)
    {
        // Output the pattern results
        mPartLabel.Text = "Present";
        // Showing pattern marker in Display
        MarkerPoint lMarker = new MarkerPoint("HSLabel",
            lLabelLocator.GetMatchPositionX(0, 0, false),
            lLabelLocator.GetMatchPositionY(0, 0, false),
            true);
        mDisplay.Markers.Add(lMarker);
    }
}
```
lMarker.Color = HSColor.Blue;
}
else
{
    mPartLabel.Text = "Absent";
}

// Retrieving / Showing sequence execution timing
mTime.Text =

mVisionProjectControl.VisionProject.Sequences[0].ElapsedTime.ToString("0.00 ms");
    mDisplay.RefreshDisplay();
    Application.DoEvents();
} while ( mCheckContinuous.Checked );

This code first creates a reference to the Pattern Locator tool to enable specific programmatic access. From this reference, Match Count is retrieved and appropriate status is displayed in corresponding control. If a match is found, using the World coordinate system, the calibrated X-Y position of match is used to draw a non-editable point marker on the Display.

10. Coding is now completed. Save your work and test the application. You should see the match status in the Inspection Part Label text box of your application. You should also see the point marker you added on the Display highlighting the match position.

**Testing the Application**

1. You are ready to test your application.
2. To start the running mode, press the F5 key. Click **Execute Inspection** a few times.
3. The measurements on the found instance should be updated after each inspection.
4. Enable the **Continuous Mode** check box and click **Execute Inspection**.
5. The application should run in continuous mode.
6. Application should now look as shown in Figure 102:
Figure 102 Application Interface showing Part HS Label match

This concludes this tutorial step. After debugging, save your work and move on to the next tutorial step where you will add other inspection tools to your application.

Next: Add Two Edge Locator Tools
Add Two Edge Locator Tools

In this tutorial step, you will learn how to set up, configure and use two Edge Locator tools to measure non-parallel edges on the part.

Positioning the First Edge Locator

In the previous steps, you have created and modified the vision project file for this tutorial. The application has been configured to automatically load the vision project when the application starts.

In this step you will modify the vision sequence to include two Edge Locator tools.

1. Restart the application
2. In the Sequence Manager, select the Hook Inspection sequence.
3. In the toolbar, click the 'Edit Sequence' icon to start the Sequence Editor.
4. Execute once by clicking Execute icon from toolbar.
5. From context menu select Add > Edge Locator command.
6. Double-click on the tool title and rename tool, from Edge Locator to Part Left Edge.
7. Acquire Image is automatically selected as the Input image provider.
8. Under Location, select Locator as Frame Input and click on Location button.
9. Set the frame-based position of the tool as shown in Figure 103. You can either enter values manually or edit the location bounding box in the display, with the mouse.

The Edge Detector detects edges that are parallel to its Y-axis: adjust the rotation to best match the inclination of the edges you want to measure.

Figure 103 Setting the Location of the Edge Locator Tool

Configuring First Edge Locator Constraints

You will now set up constraints that will allow the Edge Locator to find edges in defined frame-based location. These constraints can be changed in the Advanced Parameters property grid.
1. In **Advanced Parameters**, under **Results**, set **Coordinate System** to **Object**.

2. Under **Edge Constraints**, set **Constraints** to **Position**.

3. Under **Edge Constraints**, set **Polarity Mode** to **Dark To Light**.

4. Under **Constraints**, set Position Constraint to $0, 1, 0.509, 0.560$.

5. Keep default values for all other **Advanced Parameters**.

6. Close the Sequence Editor.

7. Click the 'Save Project' icon to save changes made to the tutorial vision project.

Now that the first Edge Locator is properly configured, the second one must be added.

**Positioning the Second Edge Locator**

The vision sequence must now be modified to include a second Edge Locator tool.

1. In the toolbar, click the 'Edit Sequence' icon to start the Sequence Editor.

2. Execute once by clicking the 'Execute' icon from toolbar.

3. From context menu select **Add > Edge Locator** command.

4. Double-click on tool title, rename tool from **Edge Locator** to **Part Right Edge**.

5. **Acquire Image** is automatically selected as the **Input** image provider.

6. Under Location, select **Locator** as **Frame Input** and click on **Location** button.

7. Set the frame-based position of the tool as shown in **Figure 104**. You can either enter values manually or edit the location bounding box in the display, with the mouse.

The Edge Detector detects edges that are *parallel to its Y-axis*: adjust the rotation to best match the inclination of the edges you want to measure. Skew can also be modified to configure the tool to match the orientation of the edge.

For the current example, the *skew was set to -18 degrees* to adapt to the edge slope in the inspected area. The same result could be obtained with Rotation=72 and Skew=0 degrees.
You will now set up constraints that will allow the Edge Locator to find edges in defined frame-based location. These constraints can be changed in the Advanced Parameters property grid.

**Configuring Second Edge Locator Constraints**

You will now set up constraints that will allow the Edge Locator to find edges in defined frame-based location. These constraints can be changed in the Advanced Parameters property grid.

1. In Advanced Parameters, under Results, set Coordinate System to Object.
2. Under Edge Constraints, set Constraints to Position.
3. Under Edge Constraints, set Polarity Mode to Light to Dark.
4. Under Constraints, set Position Constraint to 0, 1, 0.497, 0.497.
5. Keep default values for all other Advanced Parameters.
6. Close the Sequence Editor.
7. Click the 'Save Project' icon to save changes made to the tutorial vision project.

Now that the Edge Locator tools are properly configured to measure part width, you will want to test and observe the Edge Locator results on other instances of the Object. Code must now be added to your application in order to highlight Edge Locator results.
Add Code for the Edge Locators

You will finish up by adding code to your Visual C# application that will display the measurement computed from the Edge Locator tools in the interface. You will also add a line marker on the display to show the measurement.

1. From the **Solution Explorer** context menu, select the **Add Reference ...**
2. In the displayed dialog, click the **Browse ...** button.
3. Move to **[Common Files]\Adept Technology\AdeptSight\PlugIns\Tool**.
4. Select **PatternLocatorPlugIn.dll** file and click OK twice to accept adding new reference.
5. Select the newly added reference and in **Properties**, change **Copy Local** to **false**.
6. Now that appropriate reference have been added, code referencing the Edge Locator tools can be added.
7. Select the **HookInspection.cs** window.
8. Locate the **ExecuteButton_Click** method and insert the lines of code shown in bold:

   ```csharp
   try {
       Locator lLocator = null;
       Caliper lWidthCaliper = null;
       Caliper lHeightCaliper = null;
       BlobAnalyzer lHoleBlob = null;
       PatternLocator lLabelLocator = null;
       EdgeLocator lPartLeftEdge = null;
       EdgeLocator lPartRightEdge = null;
       mExecuteButton.Enabled = false;
       // Retrieving / Showing EdgeLocators results
       lPartLeftEdge =
       mVisionProjectControl.VisionProject.Sequences[0][6] as EdgeLocator;
       lPartRightEdge =
       mVisionProjectControl.VisionProject.Sequences[0][7] as EdgeLocator;
       if (lLocator.GetInstanceCount(0) > 0 &&
           lPartLeftEdge.GetEdgeCount(0) > 0 &&
           lPartRightEdge.GetEdgeCount(0) > 0 ) {
           // Output the computed width results
           float lPartWidth =
           lPartRightEdge.GetEdgePositionY(0, 0) -
   ```

9. In the existing **do** loop, add the lines of code shown in bold:

   ```csharp
   do
   {
   ...
   // Retrieving / Showing EdgeLocators results
   lPartLeftEdge =
   mVisionProjectControl.VisionProject.Sequences[0][6] as EdgeLocator;
   lPartRightEdge =
   mVisionProjectControl.VisionProject.Sequences[0][7] as EdgeLocator;
   if (lLocator.GetInstanceCount(0) > 0 &&
       lPartLeftEdge.GetEdgeCount(0) > 0 &&
       lPartRightEdge.GetEdgeCount(0) > 0 ) {
   ```
AdeptSight Standalone C# Tutorial - Add Code for the Edge Locators

```csharp
lPartLeftEdge.GetEdgePositionY( 0, 0 );
mPartWidth.Text = lPartWidth.ToString( "0.00" );
// Showing part width marker in Display
MarkerLine lMarker = new MarkerLine(
    "Part Width",
    lPartRightEdge.GetEdgePositionXWorld( 0, 0 ),
    lPartRightEdge.GetEdgePositionYWorld( 0, 0 ),
    lPartLeftEdge.GetEdgePositionXWorld( 0, 0 ),
    lPartLeftEdge.GetEdgePositionYWorld( 0, 0 ),
    true);
mDisplay.Markers.Add( lMarker );
lMarker.AnchorStyle = MarkerAnchorStyle.CROSS;
lMarker.Constraints = LineMarkerConstraints.LineNoEdit;
lMarker.PenWidth = PenWidth.Thin;
lMarker.Color = HSColor.Red;
}
else
{
    mPartWidth.Text = "";
}
// Retrieving / Showing sequence execution timing
mTime.Text = mVisionProjectControl.VisionProject.Sequences[0].ElapsedTime.ToString("0.00 ms");
mDisplay.RefreshDisplay();
Application.DoEvents();
}
while ( mCheckContinuous.Checked );
```

This code first creates references to the Edge Locator tools to enable specific programmatic access. From these references, results are retrieved and part width is computed and displayed in corresponding control. Using the World coordinate system, the calibrated X-Y position of the first and second edges are used to draw a non-editable line marker in the Display.

10. Coding is now completed. Save your work and test the application. You should see the measurement in the Inspection Part Width text box of your application. You should also see the line marker you added, highlighting the measurement in the Display.

### Testing the Application

You are ready to test your application.

1. To start the running mode, press the F5 key. Click Execute Inspection a few times.

2. The measurements on the found instance should be updated after each inspection.
3. Enable the Continuous Mode check box and click Execute Inspection.
4. The application should run in continuous mode.
5. Application should now look as shown in Figure 105.

![Figure 105](image)

You have completed the tutorial!

Continue learning about AdeptSight in the following tutorials and online help topics

- [AdeptSight Pick-and-Place Tutorial](#)
- [AdeptSight Conveyor Tracking Tutorial](#)
- [Setting Up System Devices](#)
- [Managing Models in AdeptSight](#)