HEXSIGHT™

Quick Start Guide

HexSight™
Quick Start Guide

Table Of Contents

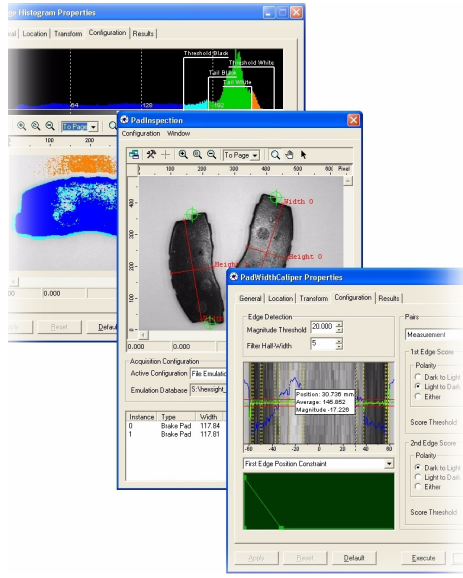
What is HexSight?

HexSight is a comprehensive library of machine vision tools, available as ActiveX controls, C++ DLL libraries, or a .NET DLL assembly.

At the heart of HexSight, the unique and powerful Locator tool provides accurate and robust object recognition through geometric object location. The Locator finds the contours of parts in the image, and quickly identifies each part's position and orientation. This information can be sent to a robot or machine controller to guide part handling.

In addition, HexSight provides integrated calibration and a complete set of inspection tools.

HexSight tools can be model-based to leverage the Locator’s results by automatically transposing tools on parts regardless of their scale or rotation.



System Requirements

To run HexSight on your computer, the following hardware and software are required.

Software Requirements

- Operating system: Windows XP with Service Pack 3, Windows Vista (x86 or x64) or Windows 7 (x86 or x64).
- Visual Studio .NET (2003 or higher), or other ActiveX-compliant environments.
- DirectX 8.0 or later is required for IEEE 1394 (FireWire) or USB cameras
- Carnegie Mellon University (CMU) DCAM driver is required for IEEE 1394 (FireWire) DCAM cameras
- Adobe Acrobat Reader 5.0 or higher and Internet Explorer 5.0 or higher are recommended to respectively access PDF format documentation and online HexSight Help.

Hardware Requirements

- Processor: Pentium II or higher
- Hard Drive space: From 15 MB to 200 MB depending on selected components
- Memory (RAM): 128 MB
- Display: Minimum resolution: 1024 x 768 pixels. Colors: 65,536

Device and Frame Grabber Support

Frame Grabbers

HexSight supports the following frame grabbers:

- Dalsa IC-ASYNC
- Dalsa PC-DIG
- Dalsa PCVision
- Data Translation DT-3152
- Data Translation DT-3155
- Data Translation DT-3157
- Imagenation PX610A
- Imagenation PCX200
- Matrox Meteor-II/Standard
- Matrox Meteor-II/Multi-Channel
- Matrox Meteor-II/Digital
- Matrox Meteor-II/Camera Link
- Matrox Orion
- Opteon DepictMaster 4X1

Devices

The following devices are supported:

- All cameras compatible with the frame grabbers supported by HexSight.
- IEEE 1394 (Firewire) and USB cameras with a DirectShow driver.
- IEEE 1394 (Firewire) cameras using the DCAM protocol.
- GigE Vision™ cameras
- LMI Technologies Gocator™ sensors

Installing HexSight Software

Before Installing HexSight

1. Ensure that your system meets the minimum system requirements specified in the previous section.
2. Ensure that you have Administrator privileges on your PC. You must have permission to write to the necessary system registry sections.
3. To acquire live images in HexSight you will need a compatible camera and frame grabber or a compatible IEEE 1394 camera.
4. If you are using a frame grabber, we strongly recommend that you install the original driver for the frame grabber before installing HexSight. Please refer to the documentation provided with the frame grabber to ensure that you follow the appropriate procedure
5. If you are using HexSight in demonstration (Demo) mode and/or you do not have a dongle, you will only be able to run existing applications, such as the example applications and the tutorial example provided with HexSight. Any changes made to parameters in an application will be lost once you close the application.
6. Visual Studio 2003 or higher should be installed on your PC before installing HexSight.
7. Close any other open applications.

Installing HexSight

1. *If you have an older version of HexSight installed on your system, you must first uninstall this version before installing the new one.* See Uninstalling HexSight.
2. Insert the HexSight USB Flash Drive and wait for the Autorun to begin, or run the downloaded installer package.
3. When the welcome window is displayed, choose **Next**.
4. Read the Licence agreement and then click **I Agree.**
5. Chose and installation folder and click **Next**.
6. Select a start menu folder and click **Install**.
7. HexSight will now install. When the installation is complete, click **Close** to exit the installation program.

Uninstalling HexSight

1. In the Control Panel of your computer, double-click **Add/ Remove Programs**.
2. Find **HexSight 4.3** in the list of programs, select it and then click **Remove**.
3. Click **Uninstall** to confirm the removal.
4. When the removal is completed, close the Add/Remove Programs window.

Installing Frame Grabbers and other Devices

Data Translation Frame Grabber

The interface DLLs needed to connect to the frame grabber driver are installed with HexSight. No specific installation procedure is required.

Imagenation Frame Grabber

In order to connect to a Imagenation frame grabber driver, you will need to copy the following interface DLL files from their installation directory (**bin** subfolder of the installation path) to the Windows system path (**...\system32**).

- PX610A: WPX5_NT.DLL
- PCX200: FRAME_32.DLL, PXC_NT.DLL, PXC2_NT.DLL

Dalsa Frame Grabber

To configure a Dalsa frame grabber, HexSight uses configuration files built with the appropriate Dalsa configuration application. These configuration applications include the device drivers for the frame grabber:

- IC-ASYNC: ITEX Camera Configurator, provided with the ITEX Library SDK (Version 4.1 or higher)
- PCVision: ITEX Camera Configurator, provided with the ITEX Library SDK (Version 4.1 or higher)
- PC-DIG: IFC Camera Configurator, provided with the IFC Library SDK (Version 5.0 or higher)

Itex and IFC software is not provided with the frame grabber and must be purchased separately from Dalsa. In accordance with the Dalsa license agreement, you can however redistribute the device drivers and configuration files as long as you include a frame grabber with your application, so only one copy of the software is required for the development system.

Matrox Frame Grabber

To configure the frame grabber, HexSight uses configuration files built with the Intellicam application provided with MIL-Lite (Version 6.01 or higher), which also includes the device drivers for the frame grabber.

When you install HexSight on a development system, the Intellicam configuration utility is necessary for building the configuration files used to control the frame grabber.

MIL-Lite is not provided with the frame grabber and must be purchased separately from Matrox. In accordance with the Matrox license agreement, you can however redistribute the device

drivers and configuration files as long as you include a frame grabber with your application, so only one copy of MIL-Lite is required for the development system.

Opteon Frame Grabbers

The interface DLL, needed to connect to the frame grabber driver, is installed with HexSight. The frame grabber can be fully configured from the HexSight user interface. No external tool is needed.

IEEE 1394 Device - DirectShow Device

Any IEEE 1394 camera (FireWire) with a DirectShow driver is supported. All DirectShow standard parameters are accessible programmatically. No specific installation procedure is required.

IEEE 1394 Device - DCAM Device

Any IEEE 1394 camera (FireWire) using the DCAM standard is supported. It allows for a greater control over the device than with simply the DirectShow driver, but requires the installation of the CMU DCAM Driver.

GigE Vision™ Cameras

HexSight supports all GigE Vision™ cameras. The drivers are installed automatically when you add your camera to your application. For more information, please see the Acquisition Device chapter of the HexSight User's Manual.

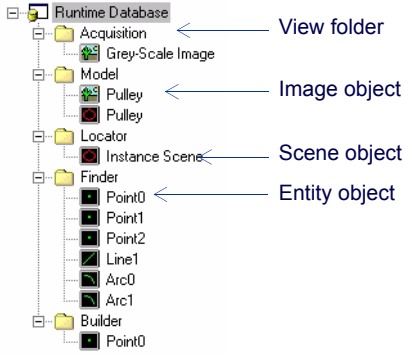
Gocator™ Sensors

Gocator™ sensors 2000 and 2300 series are natively supported in HexSght. The 3D information is captured directly in HexSight for additional vision inspection capabilities. Initially, only a limited number of tools are able to process the heightmaps (Locator, Image tools). Other tools can use the conversion method in the ImageProcessing tool to obtain greyscale images. For more information, please see the Acquisition Device chapter of the HexSight User's Manual.

Basic HexSight Terminology

Configuration: A configuration is a set of HexSight tool instances and the configured properties of each tool. Configuration files are saved to the HexSight CFG format. In the Acquisition Device, a Configuration is a set of parameters for a given input source.

Database: A collection of Views, which can contain any number of database objects: Scenes, Images and Entities. The runtime database acts as a database exchange container for the HexSight processes.



GUI representation of a runtime database

Disambiguate: Applied to object location, disambiguation means to resolve the ambiguity between many possible solutions by using one or more distinguishing characteristics or features of an object. For example: The Locator tool can easily disambiguate between very similar objects.

Entity: An object that is created and used by HexSight Finder tools. Basic entities are Points, Lines and Arcs. These entities can be saved to proprietary file formats and accessed through the Class Library.

Image: A matrix of pixels that represents the 2D appearance of an object. An image is a raster (bitmap) graphic, as opposed to a Scene, which is a vector description.

Input: Data used by the process algorithm to compute a result, which can be an Image, a Scene, or a set of numeric values.

Instance: A representation or an occurrence. For example: each occurrence of a specific object is an instance.

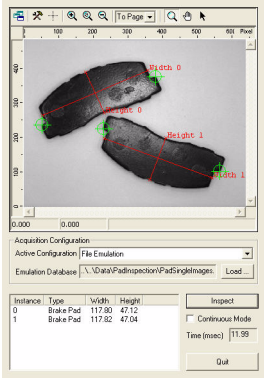
Instantiate: To actualize an abstract representation by defining a particular configuration of the process, giving it a name, and including it in the process list. For example, if you add a Caliper tool to an application, you are instantiating a HexSight HSCaliper process.

Model: A Model is a representation of an object, based on the geometry of its contours. An object Model is the reference

used to identify and locate instances of objects of the same type.

Model-Based: Model-based refers to positioning a HexSight tool on a model of a given object. Each time an instance of this object is found, the model-based tool is applied on the correct position on the found object.

Output: Data generated by the process algorithm, e.g. an Image, a Scene, an Entity or a set of numeric values.



Model-Based Tools on Found Parts

Properties: The characteristics of an object.

Scene: A vectorized description of an object's contours, as opposed to an Image, which is made up of pixels.

View: A collection of Scenes, Images and Entities. By analogy, one could say that a View is a folder that can contain Scenes, Images or Entities.

What is a HexSight Application?

A HexSight application is composed of a sequence of processes, which can also be visualized as tools or modules, each having its own function or purpose.

Each of the processes exchange data through the runtime database. A typical HexSight application contains at least the following two processes: the **Acquisition Device tool** and the **Locator tool**.

These processes perform the following basic operations:

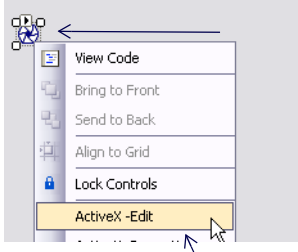
1. The **Acquisition Device** acquires an image of the work surface. Acquired images can be calibrated within the Acquisition Device interface.
2. The **Locator** tool generates a vectorized Contour Scene from the Images provided by the Acquisition Device. The Locator finds and locates, in the Contour Scene, all occurrences of objects of defined in the Model database. Models are created and edited in the Locator interface.

Inspection tools such as the Caliper can be added to these two basic processes depending on the task to be performed.

Creating a HexSight Application

A typical HexSight application is created by placing a HexSight Application control on a blank Visual Studio form. The steps required to complete such a task are:

1. Select the HexSight Application control icon in the Visual Studio toolbox and place this control on a blank form. If the icon is not available in the toolbox, see ***Adding a Component to the Toolbox*** for more details.

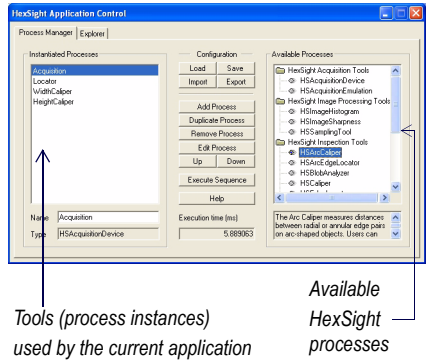


Opening the Process Manager

2. Drag a square with the mouse on the form and release the left mouse button to place the HexSight Application Control.
3. Right-click the HexSight Application Control in the form and select **ActiveX - Edit** to display the Process Manager window. The Process Manager is used to instantiate and configure the sequence of processes.

The Process Manager interface contains two lists:

- **Available Processes**, which lists all HexSight process types and
- **Instantiated Processes**, which contains the names of the processes used in the current application.



Process Manager Window

Adding a Component to the Toolbox

To modify the content of a HexSight application, you must add a HexSight Application control to a Windows form. This enables you to access the Process Manager and the Explorer interfaces in design mode or in runtime.

You should also add at least one HexSight Display control to the form. This will allow you to visualize results in runtime mode.

To add a control to the toolbox:

1. From the toolbox context menu, select: Choose Items.
2. Select the COM Components tab and scroll down to find HSApplication Control. Enable its check box and click OK. The HexSight Application Control appears as a lens shutter icon in the toolbox.

Note: In a Visual Basic program, a variable cannot be declared as a given HexSight control type, if it is declared as a component. Nevertheless, any object created by pasting a control (selected as a component) on a Visual Basic form can be accessed programmatically using the name of the object.

The following HexSight control types should be used as components:

- Application control
- Display control

Referencing a Control

To create HexSight objects programmatically, all HexSight control types used in the user's program must be referenced. This is accomplished the following way:

1. From the Microsoft Visual Studio menu bar, select Project > Add Reference...
2. Select the COM tab, and enable the required HexSight controls.
3. Click OK. The controls are added to project's references and can now be accessed programmatically.

Note: A HexSight control cannot be declared as a component and as a reference at the same time.

HexSight Toolkit

HexSight provides comprehensive set of vision tools for object finding, inspection, image processing and more.

Acquisition Tools

Acquisition Device: The Acquisition Device acquires images from a frame grabber or from a database of images.

Locator Tools

Locator: The Locator tool finds and locates objects that are defined by models. Models are created using the integrated Editor and saved to database files. The Locator provides the scale factor, orientation, and position for each located object.

Arc Finder: The Arc Finder finds and locates circular features on objects and returns the coordinates of the center of the arc, the start and end angles, and the radius.

Line Finder: The Line Finder finds and locates linear features on objects and returns the line angle and point coordinates.

Point Finder: The Point Finder finds and locates point features on objects and returns the angle as well as the coordinates of the found point.

Builder: The Builder creates geometric entities such as points, lines or arcs from user-defined parameters or by combining other geometric entities.

Image Processing Tools

Image Processing Tool: The Image Processing Tool processes images by applying arithmetic, assignment, logical, filtering, morphological or histogram operators. Users can define custom filtering operators.

Image Sharpness Tool: The Image Sharpness Tool computes the sharpness of preponderant edges in a user-defined region of interest.

Image Histogram Tool: The Image Histogram tool computes greylevel statistics within a user-defined region of interest.

Sampling Tool: The sampling tool is used to extract a section of an image and output it as a separate Image. The sampling tool can be configured to operate as model-based. It can be used by a HexSight application to apply a custom processing on a region of interest on an object.

Inspection Tools

Edge Locator: The Edge Locator finds and locates an edge or a set of edges that meet user-defined criteria.

Arc Edge Locator: The Arc Edge Locator finds and locates an edge or a set of edges on an arc or circular-shaped area.

Caliper: The Caliper finds and locates one or more edge pairs and measures distances between the two edges within each pair.

Arc Caliper: The Arc Caliper finds and locates one or more edge pairs on an arc or circular-shaped area and measures distances between the two edges within each pair.

Pattern Locator: The Pattern Locator finds and locates instances of a pattern occurring within an Image. Pattern images are provided by the HexSight Locator control.

Blob Analyzer: The Blob Analyzer finds and locates blobs within a defined area on an Image and returns various results for each blob.

ColorFinder: The Color Finder searches and analyzes images to find areas of color that match user-defined filters.

Symbology Tools

Barcode Reader: The Barcode Reader reads and extracts information from 1D symbologies commonly known as bar codes.

Data Matrix Reader: The Data Matrix Reader reads and extracts information from 2D symbologies commonly known as data matrixes.

OCR Fixed Font: The OCR Fixed Font tool recognizes and reads OCR-A, OCR-B, or SEMI font character strings.

Color Support and Functionality in HexSight

HexSight provides support for acquiring and using color images. The color license adds color capabilities to various HexSight tools and processes.

Color Image Acquisition

The Acquire Image tool accepts and outputs color images provided by supported color cameras.

Color Calibration

In applications where accurate color is required, color calibration ensures that the input images contain accurate color information. The built-in Color Calibration Wizard steps you through color calibration and requires a standard GretagMacbeth ColorChecker target.

Color Locator

The Locator can be configured to differentiate between object based on their color. In the model-building process, a custom color shading area can be defined for each object. This shading area allows the Locator to use color information when locating objects.

Color Processing with Inspection Tools

Most inspection and finder tools can process images on the basis of color.

- The Edge Locator, Arc Edge Locator, Caliper, and Arc Caliper tools use color information to extract edges.
- The Arc Finder, Line Finder, and Point Finder tools use color information to extract edges for finding geometric entities (arc, lines, points.)
- The Blob Analyzer can use Hue, Saturation, and Luminance values to define blobs.

Color Finder Tool

The Color Finder tool can identify the presence or absence of defined colors, and analyze the predominance of colors in images. For example, it can be used to differentiate similarly shaped objects of different colors.

Project Examples Provided with HexSight

The HexSight Development package provides complete examples of machine vision applications. These examples show how to implement specific features such as automated calibration and model teaching.

Examples of application projects are provided for the Microsoft Visual Studio 2003 (or higher) development environment.

Board Location

This application identifies and locates a printed circuit board (PCB). The application demonstrates how to create a model in which one big feature is used to locate the object and a second smaller feature is required to correctly identify the orientation of the object. A Sampling tool samples the image from a model-based region on the located boards. This sampled image is used as an input to a second Locator tool which then checks for the presence of a small label printed on the PCB.

Board Location Model-Based

This application is similar to the Board Location example. However, this example uses a Model-Based Locator, instead of a Sampling tool, to identify subfeatures on the object.

Bracket Inspection

This application identifies, locates and inspects four different models of flat brackets. The Blob Analyzer tool finds the center of mass of each hole on the bracket. Parts are rejected if the position of any of its holes are not within tolerance ranges.

Calibration Demo

This application shows how to use the calibration objects: HSCalibrationXYScaleInterface, HSCalibrationPerspectiveInterface, and HSCalibrationDistortionInterface, to implement automated calibration programmatically, without showing the interactive user interface.

Contour Detection

This application shows how to access the calibrated and uncalibrated contour data extracted from an image by the Locator. The application also shows how to use properties of the extracted contour data, such as perimeter and area, to manually detect circular shapes in the image. This can be used, for example, to detect dots during a custom calibration procedure using a non-standard target.

Contour Draw Demo

This application shows how to manually draw the detected contours (model and instance) into a standard Visual Studio graphics object by directly accessing the contour data generated by the Locator through the HsScene object.

Image Processing

This application shows how the Image Processing tool can be used to apply operations to an image, such as noise filtering, before the image is output to the Locator tool.

Locator Demo

This demo provides a ready-built application, useful for testing HexSight with your own images. This demo also contains configurations to illustrate Model Optimization and the Shading consistency feature.

Finder Demo

This demo provides an example of the use of HexSight Finder Tools: Arc Finder, Line Finder, Point Finder and Builder.

Model Demo

This application shows how to use the HSModelEditorInterface object to implement automated model teaching programmatically, without showing the interactive user interface.

Pad Inspection

This application identifies and locates brake pads on the work surface and performs two measurements on each located pad. This application provides an example of how the Caliper tool can be used to perform linear measurements.

Part Sort

This application provides the type, position and orientation of 4 different parts that are randomly placed on the work surface. Multiple parts can be identified and located at the same time. These parts do not have to be of the same type. In addition to multi-model capabilities, this application also illustrates how to configure the **Locator** tool for robustness to occlusions.

Pulley Location

This application identifies and locates a pulley on the work surface. The Locator identifies the four possible rotational symmetries. An Edge Locator tool locates the pulley's keyway to identify the correct orientation. An Arc Edge Locator is used to measure the angular position of the four spokes of the pulley, with respect to its keyway.

Regulator Inspection

This application sorts and inspects electronic parts. The Pattern Locator tool

is used to disambiguate the two models of T0220 regulators that are used. Distances between the legs of each regulator are measured using the Edge Locator tool and part is rejected if the distance between two legs is not within tolerance ranges.

OEMLocatorGeneric

This application shows typical Locator integration for OEM projects using ActiveX layer.

OEMLocatorDLL

This application shows typical Locator integration for OEM projects using DLL layer only. Since DLL layer do not expose any Graphical User Interface, this example also shows how to display HexSight images and Locator results. It also includes an implementation of a custom model edition window using programmatic calls to the DLL interface.

Color Locator Demo

This demo shows the potential of the Locator in regards with Color detection. Not only can the Locator detect edges that may be invisible when converted to greyscale, but it can also differentiate the instance based on the color of the model.

Color Target Inspection

This application demonstrates the use of the ColorFinder tool to inspect the color of specific areas of the image. By adding filters corresponding to particular colors, you can easily say if a colored area is within the acceptable tolerance defined.

Demonstration Mode

The HexSight Demonstration Mode is provided for the purpose of evaluating our software. This mode is automatically enabled when you run HexSight without the hardware protection key that is provided with the purchase of a HexSight software package.

This demonstration mode has some features removed, yet will allow you to evaluate the functionality and power of HexSight.

Opening and Running in Demo Mode

1. When you first open HexSight in demo mode, a pop-up window appears, advising you that the hardware protection key was not found and that the software will run in the demonstration mode.
2. The application then pauses for approximately 10 seconds. After this delay, the application resumes once you click **OK**.
3. This reminder pop-up window will appear again at random intervals during execution; simply click OK and continue your evaluation.

Demo Mode Limitations

In the Demo Mode, the application development capabilities of the HexSight environment are deactivated.

- You can load and run the examples that are installed from the HexSight CD.
- You cannot operate in design mode i.e. you cannot build an application by selecting and instantiating tools.
- You cannot save a configuration file. You can however open an existing application.
- Since you cannot run in design mode you cannot fully do the tutorial. You can however open the Tutorial files that are installed from the HexSight CD.

Using the Demo Mode with the Tutorial

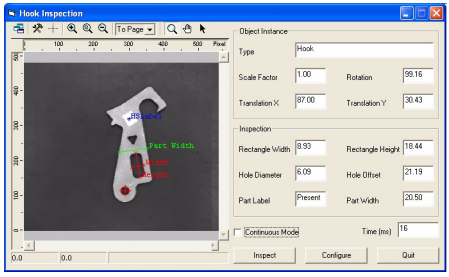
Even though you cannot use the demo mode to build an application as explained and illustrated through the tutorials, you can run any of the applications provided on the HexSight CD, either in the Examples or the Tutorial folders.

To best familiarize yourself with the functionalities of HexSight, we

recommend that you start out with the tutorial application, as follows:

1. Open the final application provided for the tutorials. The VB.NET, Visual C++, and C# tutorial files are located in the Tutorials folder, under the HexSight installation folder. For example, the C# tutorial files are located in: **Tutorial/C#/HookInspection/Part6**

2. To run the application, in the Hook Inspection interface, click the **Inspect** button to launch a first execution.



Hook Inspection Tutorial

Getting Help

Online Help

There are four ways to access the HTML Online Help:

1. Click on a HexSight control icon, either in the Visual Basic toolbox or on one of the forms of the application, then press **F1**.
2. Place the cursor of the Visual Basic Editor in the middle of a string that corresponds to a HexSight control type, property, method or event, then press **F1**. This will work if the icon of the control appears on the toolbox or if the control is referenced.
3. From the Process Manager (Application Control interface), click the **Help** command button.
4. In any property window of a HexSight tool, place the cursor on an object and press **F1**.

Technical Support

We provide technical support for our customers via our web page as well as by telephone.

E-mail Queries

Send your technical questions by e-mail, directly to our technical support team at **support@hexsight.com**.

Your e-mail queries will be directed to the appropriately qualified personnel and promptly answered.

Telephone

You can contact the HexSight technical support between 9:00 A.M. and 5:00 P.M. PST at +1 604 636 1011.

www.LMItechnologies.com