

HexSight Version 4.4 Release Notes

New Features

- Support for **LMI Technologies Gocator sensors** with firmware **version 4.0**.

Bug Fixes

- Update of the **CMU/DCAM** libraries to version 6.4.6.
- Correction of an issue with the buttons in the ModelEditor interface of the **HSLocator**.
- Correction of an editing issue for the reference points in the ModelEditor interface of the **HSLocator**.
- Correction of an issue for **HSDisplay** control.

HexSight Version 4.3.1 Release Notes

New Features

- Support for **LMI Technologies Gocator sensors 2000-2300 series** is done by copying the **Go2.dll** file correspondant to the device firmware in the HexSight **Controls** folder.
- The **ArcFinder**, **LineFinder** and **PointFinder** have been modified to natively support the **Gocator sensors** heightmap images.
- Addition of the new **EvaluatorTool**. It has the ability to convert results from other tools to Pass or Fail result using threshold comparison or mathematical operations, in order to simplify application development.
- Addition of the new **OutputTool**. It has the ability to schedule a digital IO from a Gocator, based on a Pass/Fail result from a HSEvaluatorTool. The schedule can be an encoder offset or a time delay.
- Replaced the grids in controls with XP themed grids.

Bug Fixes

- Removed Basler/GenICam devices from being detected as DirectShow, and only allow them to be detected as GigE Vision devices.
- Better device matching on load in the **AcquisitionDevice**.
- Correction to allow mask images to be of different size than models.
- Reduced the memory footprint of the **ArcFinder** to a minimum.
- Reduced the memory footprint of the **EdgeDetection** module.
- Fixed a crash occuring in the **HSImageSharpness** tool.

HexSight Version 4.3.0 Release Notes

New Features

- **LMI Technologies Gocator sensors 2000-2300 series** are now supported. They are discovered when on the same subnet, or can be added by using the drop down list of the Add Device dialog box.
- The **Locator** has been modified to natively support the **Gocator sensors** heightmap images.
- The **ImageProcessing** and **ImageHistogram** tools have been modified to natively support the **Gocator sensors** heightmap images.
- The **ImageProcessing** tool can also convert those images to greyscale images, in order to use them directly in any other tool.
- Addition of a new calibration to contain the Z dimension introduced by the heightmap support. The new Z-offset and Z-resolution can be used to compute the height of any given pixel value.
- Addition of the orientation range in the **PatternLocator** tool.
- Update of the GigE drivers, for newer devices.
- Support for color mask images during Model build process.
- Support for XP themes in all controls.

HexSight Version 4.2 Release Notes

Licensing Modification

- Developer and Runtime versions of **HexSight** have now been consolidated into a single library. All development features are enabled and the naming of the various **HexSight** packages has been simplified to **Express**, **Professional**, and **Premium** editions.

AcquisitionDevice

- **IDS uEye** cameras are now supported using DirectShow. The discovery of new unidentified DirectShow devices is also improved.
- Updated GigE drivers to the latest, resolving an issue with Basler devices.
- Added DCAM 1394b Support flag checkbox in the interface.
- All 3 newest devices, **GigE-DCAM-DirectShow**, are now grabbing concurrently. Initiating a Start from multiple devices will block the Wait function until any device request is finished, not just the oldest request.
- Fixed an issue with Calibration Dialog with multi-threaded acquisitions.
- Fixed the issue where selecting back and forth a configuration changes the selected device for a configuration.
- Fixed the order of the DCAM modes.
- Better handling using GigE devices without SingleFrame option (JAI).
- Reduced the amount of memory used for GigE devices; the allocation of the pipeline is done only when really needed (no decrease in performance).

ArcFinder

- The match quality of the found Arc has been improved in cases where the computed center is different from the user specified center.

BlobAnalyzer

- Added **Exclude Rejected Blob** option in the General Tab of the Blob Analyzer.
- Fixed an issue with images of more than 4k x 4k in size, where only a part of the blobs were found.
- Fixed an issue for the HSLInside values not being displayed correctly when changed programmatically.
- Fixed an issue with the computation of the weighted area when a slope is used in the configuration.

Display

- Added a new method, `SaveDisplay()`, to save the current window to a file, with all the entities, instances, markers and overlays.
- Fixed GDI leaks if device contexts are not released correctly.

ImageProcessing

- Masking with a Blob image is now working in color.

LineFinder

- The match quality of the found Line has been improved in cases where the orientation of the computed Line is different from the orientation of the Tool Position.

Locator

- Added a Mask Image to the search area of the Locator. Using **`LoadMaskImage()`**, **`ClearMaskImage()`** and **`GetMaskImageFilename()`**, it is now possible to block specific areas from the search. Multiple file formats are supported: jpg, bmp, png, gif, hig, hdb.
- Added new **`InstanceOrdering`** modes to obtain exactly ordered instances, instead of on a best-effort basis.

General Improvement

- Added resizable property pages and calibration dialogs.
- Added symbols generation in HexSight packages, to simplify the support.
- Added logging to a log file for investigation purposes. Use these environment variables:
 - Use **`HEXSIGHT_LOG_LEVEL`** (0 (none) - 6 (fatal)) to enable the logging.
 - Use **`HEXSIGHT_LOG_FILE`** (c:\log.log) to specify the location of the log file.

HexSight Version 4.1.2 Release Notes

Licensing Modification

- Developer license owners can now create Runtime systems with on-demand configuration changes, using the SaveConfiguration() method.

DCAM Support

- **HexSight 4.1** now supports the DCAM protocol, allowing it to interface with any camera that understands the IIDC 1394-based Digital Camera Specification Version 1.31.
 - Please refer to the **Acquisition Device** section of the **Reference Manual** for details on controlling these cameras with **HexSight**. The installation of the CMU Carnegie Mellon 1394 Digital Camera Driver is required.
 - Some 1394 Cameras may already possess a DirectShow interface, therefore, HexSight may recognize the same camera twice, using both interfaces. However, only one interface may be working at a time, depending on the driver installed for the device.

Windows 7 Support

- **Windows 7, 32 bits version** is now fully supported, in addition to **Windows XP Service Pack 3** and **Windows Vista**.

Bug Fixes

- Memory issue fixed in Locator, for Model-Based mode and for multiple models situations.

HexSight Version 4.1.1 Release Notes

Color Support for HSImageProcessing Tool

- The **HSImageProcessing** tool has been updated to allow for color image processing.

Color Image Creation

- Shapes in HSImage can now be created in color.

Sony Intelligent Camera Support

- **HexSight 4.1** now supports a range of Sony Intelligent cameras and their control APIs. The following cameras are now supported, monochrome and color:
 - **XCI-SX100**
 - **XCI-SX100C**
 - **XCI-V100**
 - **XCI-V100C**
- Please refer to the **Acquisition Device** section of the **Reference Manual** for details on controlling these cameras with **HexSight**.

Installer Updates

- The installers now support additional command line options to further their flexibility. For more information, please consult the the on-line FAQ.

Bug Fixes

- Improved GigE Vision continuous acquisition by adding flow control parameters.
- Fixed Color Images subsampling issue.
- Fixed Best Filter name and index in HSColorFinder result page.
- Fixed Apply and Cancel buttons behaviour in HSColorFinder when modifying color filters.
- Increased robustness of SetParameter function when issuing Objects of different types as arguments.
- HSDisplay Control now supports Load/Save functions.
- Fixed AddProcess/RemoveProcess programmatic functions memory leak.
- Fixed the persistence with Basler GigE devices.
- Removed ActiveMovie window appearance.
- Fixed issue with Reference Points TranslationY property in Models.

HexSight Version 4.0.15 Release Notes

GigE Camera Support

- **HexSight** now supports all GigE Vision™ cameras that are compatible with the Pleora GigE driver (www.pleora.com).

Windows Vista Support

- **HexSight** can now be installed and run on all versions of Windows Vista32.

DirectShow Driver Update

- The **DirectShow** driver in **HexSight** has been updated to work with **DirectX 9** and to support newer devices.

Color Image Creation

- The **AddImage** methods for **HSDatabase** and **HSView** classes now allow the creation of color images through the use of an additional parameter (see **Reference Manual** for details).

Silent Runtime Installer

- A **runtime-only** installer is now available and can be run silently with command line switches. This allows you to integrate the **HexSight Runtime** into your own installer.

Bug Fixes

- The Matrox MIL Library errors no longer occur.
- The AcquisitionDevice no longer fails to recover after a LAN or power failure, when connected to a GigE camera.
- There is no longer a memory leak when adding color images to a database.
- Fixed ActiveX control registration issues

HexSight Version 4.0.14 Release Notes

HexSight Environment.

- **HexSight tools are available from the global .NET assembly library (DLL) *HexsightNet.dll*.**
All tools, except the Acquisition Device, can now be integrated using an ActiveX, a DLL C++ Layer or a .NET Layer. For OEM projects, static C++ libraries can also be made available for integration into the required platform.
- **Visual C++ .NET compatibility.** The HexSight C++ wrappers are .NET compatible. (HexSight 4.0 is built using the latest Visual Studio 2005 development tools.)

Color Support

- **HexSight 4.0 adds support for color image acquisition and processing.**

The tool properties **InputGreyscaleImage** and **InputGreyscaleView** have been updated to support the new Color features. Even though the properties still exist, new corresponding properties have been added: **InputImage** and **InputView**. Those new properties should be used, as the old ones will be rendered obsolete.

The **Acquisition Device** tool accepts and outputs color images provided by supported frame grabbers and color cameras. Additionally, an optional color calibration is available to provide accurate color images.

The **Locator** tool natively works on color images using color information to extract contours used for detection. It can also be configured to differentiate objects based on their color by defining a **Custom Shading Area** for a model and performing analysis in **Shading Consistency** ordering mode. Additionally, most inspections can now 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 find geometric entities.

The **Blob Analyzer** tool can be configured to extract blobs based on color segmentation.

The **Sampling** tool can natively process images to generate sampled color images.

Finally, the new **Color Finder** tool allows the definition of color filters to extract and analyze color areas in an input image.

Locator Tool

- **ActiveX and Library interface now export *Shading Quality* for found instances.**
A ***Custom Shading Area*** can be defined within the user interface of the Model Editor. A defined custom shading area is used to perform shading analysis, if the ***Shading Consistency*** ordering mode is selected. It is now possible to get the ***Shading Quality*** value used for hypothesis ordering.

Acquisition Device Tool

Frame Grabbers

- Added support for Matrox Meteor-II / Camera Link

Color Finder Tool

- A new tool that allows the definition of color filters for color image analysis.

Class Library

- Some methods have been added to extend synthetic image creation capabilities.

Display Control

- A ***WheelOperationMode*** property has been added to control interaction with the mouse wheel.

Application Examples

Tutorials and examples of application projects are provided for the Microsoft Visual Basic 6.0, Microsoft Visual C++ 6.0 and Microsoft Visual C++ .NET development environments. Tutorials and examples of application projects developed for Microsoft Visual Basic .NET and Visual C# have been added to this release. A new C# application example has also been added to this release.

OEMLocatorNet

- This application shows typical Locator integration for OEM projects using the ***HexsightNet.dll*** .NET assembly. Because this layer does not expose any Graphical User Interface, this example 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 .NET HexSight assembly.

Hardware Protection Key

Hardware Protection Keys for HexSight version 3.2 and earlier are not compatible with HexSight version 3.3 or 4.0. HexSight 3.3 requires a 3.3 dongle, and HexSight 4.0 requires a 4.0 dongle.

Installing HexSight 4.0

To avoid confusion with ActiveX controls from previous HexSight releases, you should not try to install HexSight 4.0 along with older versions of the software on the same system. If a previous release of HexSight is already installed on a system, uninstall it before installing HexSight 4.0.

- Projects created with HexSight versions 3.3, 3.2, 3.1 and 3.0 will load on systems where HexSight 4.0 is installed and should work without modifications (except for the enums of the Display control's PenWidth property).
- Projects created with HexSight version 2.0 will load on systems where release 4.0 is installed, but will require modifications to the code that manages Display controls.
- Projects created with HexSight versions 1.1 and 1.2 will *not work* on systems where HexSight 4.0 is installed.

You should *not* try to load projects created with HexSight release 4.0 on systems where an older HexSight release is installed. Doing so may crash the development environment and force you to reboot your system.

HexSight Version 3.3 Release Notes

HexSight Environment.

- **All HexSight tools are now available as dynamic C++ libraries (DLL).** All tools, except the Acquisition Device, can now be integrated using an ActiveX or a DLL C++ Layer. For OEM projects, static C++ libraries can also be made available for the integration into the required platform.
- **Visual C++ .NET compatibility.** The HexSight C++ wrappers are now .NET compatible. This enhancement was introduced in HexSight Version 3.2 and latest Version 3.1 builds. HexSight 3.3 is now built using latest DevStudio .NET development tools.

Locator Tool

Basic and Advanced Modes

- The Locator now has two setup modes: **Basic** and **Advanced**. Toggling between the two modes can be done in the interface by enabling/disabling the **Advanced Parameters** checkbox, or programmatically, using the **AdvancedParametersVisible** property.
- The **Basic** mode allows access to only the functionality and essential parameters that are used for most simple applications. This mode is also intended for first-time or less advanced users. When in Basic mode, advanced parameters are hidden and are set to default values.
- The **Advanced** mode, for the more experienced user and/or more complex applications gives access to all Locator parameters.
- Properties that are present in both modes share the same names and same behavior. In fact, the selected **Basic/Advanced** mode affects the user interface only. The ActiveX or DLL programmatic interface gives access to all available Locator parameters *independently of the currently selected mode.*

Model-Based Locator

- The Locator can now be **Model-based** in the same manner as any other HexSight inspection tool. All properties and methods needed for Model-based positioning have been added to the Locator. The **Advanced** mode must be enabled to graphically use Model-based positioning.
- A Model-based Locator requires the input of another Locator tool, or a manually defined transform, to position its Tool Area.
- Adding Model-based functionality to the Locator required making some changes to the Locator interface. In previous HexSight Version 3.2, the **Search Area** was defined using uncalibrated **Left**, **Right**, **Top** and **Bottom** coordinates. In HexSight Version 3.3, **Tool Area** positioning has been normalized and is now defined using uncalibrated or calibrated **Center X**, **Y** positions, **Width**, **Height** and **Rotation**. The user interface reflects these

changes but for backward compatibility, the properties **SearchAreaLeft**, **SearchAreaRight**, **SearchAreaTop**, **SearchAreaBottom** and **SearchAreaUseEntireImage** have been kept. We *highly recommend* that you use the following new properties instead: **ToolPositionX**, **ToolPositionY**, **ToolWidth**, **ToolHeight**, **ToolRotation**, **ToolAreaUseEntireImage** and **CalibratedUnitsEnabled**.

Recognition Effort

- The **Recognition Effort** slider (**RecognitionLevel** property) has been improved for speed enhancement without significant trade-off on robustness. Although the center position is still the recommended value, you can now safely decrease the Recognition level towards *Quick* to reduce the execution time for *simple* applications that require faster processing.

Conformity Tolerance

- When **Model Disambiguation** is *disabled*, changing the **Conformity Tolerance** value or mode no longer triggers an automatic relearn. This new feature can be useful for multiple model applications in which all models are learned in a single step but only a single model is used/enabled for each execution.
- **Conformity Tolerance Range** can now be disabled for more versatility. This can be useful if the required **Conformity Tolerance** value is outside of the (read only) tolerance range.

Contrast Threshold

- When **Model Disambiguation** is *disabled*, changing **Contrast Threshold** value or mode no longer triggers an automatic relearn. This new feature can be useful for multiple model applications in which all models are learned in a single step but only a single model is used/enabled for each execution.
- For versatility, **Contrast Threshold** selection is no longer linked to the **Parameters Based On** selection. Contrast Threshold Mode selection is now possible in any situation; the Parameter Based On selection controls only Outline and Detail level values.
- **Contrast Threshold** selection now supports three new modes: *Models Mode*, *Models Value* and *Models Value and Mode*. These new modes will select a Contrast Threshold value based on the currently enabled models. *ModelsMode* selects the most sensitive *mode* from the enabled models, *Model Value* will select the most sensitive *value* from enabled models and *Models Value and Mode* will select the most sensitive *mode or value* from the enabled models. These new modes have been added to ensure highest sensitivity in any situations. *Models Value and Mode* is in fact the new default value for **ContrastThresholdMode**.

General

- **ShowProperties method now supports new option.** The *hsDisableSaveModelDialogOnOK* option disables the Save Models Database dialog that opens when changes to models have been made through the user interface. This can be useful when the Locator interface is managed programmatically.

Model Edition

- It is now possible to define a **Custom Shading Area** within the user interface of the Model Editor; this functionality is also available programmatically. A defined custom shading area will be used to perform shading analysis if the **Shading Consistency** ordering mode is selected.

Acquisition Device Tool

Frame Grabbers

- Added support for Opteon USB 2.0.
- Added support for Opteon Ethernet.
- Removed support for the MRT Imaging card. This PCMCIA acquisition module is no longer available and was supported on Windows NT only.

Emulation Databases

- When importing BMP or TIFF files, it is now possible to specify an Image Coordinate System. In previous HexSight Version 3.2, Image Coordinate System was set to Right Handed for all imported images.

Class Library

- Many new properties and methods have been added for better management of HSCalibration, HSDatabase, HSView, HSScene, HSImage, HSArc, HSLine and HSPoint objects. Many of these objects can now be created as standalone objects (not linked to any specific database). Some computation capabilities have also been added to HSArc, HSLine and HSPoint objects.

Display Control

- New **hsRectangleDoubleMidpointMode** constraint has been defined for Rectangle markers.
- New **hsMaskUseMarkerHitPriority** flag has been defined for Priority mask enhancement.
- **InteractiveSelectionEnabled**, **EntitySelectionEnabled**, **SceneSize** and **ImageSize** properties have been added to increase Display capabilities.

Builder Tool

- To conform to the nomenclature used by other tools, the **InputGreyScaleImageView** property has been renamed to **InputGreyScaleView**.
- Now that Class Library fully supports Arc, Line and Point objects, results are output as entity objects and their properties are available through these specific object interfaces. Based on this change, the following properties were added: **OutputArc**, **OutputLine** and **OutputPoint**.
- **ParameterName** and **ParameterType** properties have been added.

- ***OutputEntityResultValid*** was removed since it is no longer required.

Application Examples

Examples of application projects are provided for the Microsoft Visual Basic 6.0, Microsoft Visual C++ 6.0 and Microsoft Visual C++ .NET development environments. Examples of application projects developed for Microsoft Visual Basic .NET and Visual C# are actually in development. Three new application examples have been added to this release.

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.

Board Location Model Based

- This application shows how to gain advantage of Locator's new Model-based capabilities to simplify the coding of an already existing application for electronic board inspection.

Installing HexSight 3.3

To avoid confusion with ActiveX controls from previous HexSight releases you should not try to install HexSight 3.3 along with older versions on the same system. If a previous release of HexSight is already installed on a system, uninstall it first before installing HexSight release 3.3.

- Projects created with release 3.2, 3.1 and 3.0 will load on systems where release 3.3 is installed and should work without modifications (except for the enums of the Display control's PenWidth property).
- Projects created with release 2.0 will load on systems where release 3.3 is installed but will require modifications to the code that manages Display controls.
- Projects created with releases 1.1 and 1.2 will *not work* on systems where release 3.3 is installed.

You should *not* try to load projects created with HexSight release 3.3 on systems where an older HexSight release is installed. Doing so might crash Visual Basic or Visual C++ and force you to reboot your system.

HexSight Version 3.2 Release Notes

HexSight Environment

- **New Finder and Builder tools have been added to the HexSight toolbox.** These specialized tools can find and accurately locate basic geometric entities such as arcs, lines and points. The Finder tools can be run as Model-based and their results are directly output to the HexSight runtime database. As a companion for these tools, the Builder tool can combine Finder tool results to build new entities. For example, you can use the Builder to find the intersection between two located lines without typing a single line of code.
- **New symbology inspection tools have been added to the HexSight toolbox.** The Barcode Reader reads and extracts information from Barcode symbology. The Data Matrix Reader reads and extracts information from Data Matrix symbology. Finally, the OCR Fixed Font tool recognizes alphanumerical fixed-font character strings. These tools can be used as Model-based and were introduced in latest HexSight Version 3.1 builds. HexSight Symbology tools are available as an optional toolset to the Standard HexSight license.
- **Visual C++ .NET compatibility.** The HexSight C++ wrappers are now .NET compatible. This enhancement was already introduced in latest HexSight Version 3.1 builds.
- **HexSight tools now available as dynamic libraries (DLL) or static libraries (LIB).** Actually, for OEM projects, the Locator Tool can be made available as a DLL or LIB upon request.

Locator Tool

General

- **ShowProperties method supports new optional parameters.** The *Option* parameter now supports the *hsDisplayTopMost* field value to display the dialog as topmost. The new *DisabledTabs* and *DefaultTab* optional parameters can be used to remove specific tabs from the interface and to select the default tab that is displayed when the dialog is first created. This enhancement was already introduced in latest HexSight Version 3.1 builds.
- **ShowModelEditor method supports new optional parameter.** The *DisplayTopMost* optional parameter can now be used to force window to be displayed topmost. This enhancement was already introduced in latest HexSight Version 3.1 builds.
- **ShowResultsViewer method supports new optional parameter.** The *DisplayTopMost* optional parameter can now be used to force window to be displayed topmost. This enhancement was already introduced in latest HexSight Version 3.1 builds.

Contour Detection

- **Adaptive contrast threshold computation.** It is now possible to select the method used to compute the threshold used for detecting contours in the input image. You can select between adaptive low, normal or high sensitivity or a fixed value threshold. This enhancement was already introduced in latest HexSight Version 3.1 builds.

Model Edition and Optimization

- **Improved automatic model-teaching algorithms.** The automatic model-teaching algorithms have been completely reviewed and enhanced to generate models that will ensure accurate and robust location of objects.
- **Added fully programmatic model-edition capability.** The Locator Tool model edition interface (*HSModelEditorInterface*) now includes all needed properties and methods for full programmatic model edition. All GUI functionality that is in the *HSLocator*'s Model Editor window is now available programmatically.
- **New Model Optimizer capability.** To further improve the quality of models built by the automatic model-teaching algorithms, a Model Optimizer module has been added to the *HSLocator*. This module can adapt and improve a model by analyzing the results of the location process on a specified number of found instances. This can be useful to remove unstable features resulting from specular reflections or to adapt features of slightly deformable objects. This module can be programmatically accessed through the *HSLocator* model optimizer interface (*HSModelOptimizerInterface*)

Search Algorithm

- **Speed optimization.** This new release will run faster than version 3.1 without compromising the robustness or accuracy. This enhancement was partly introduced in latest HexSight Version 3.1 builds.
- **Memory consumption.** This new release will require less memory than version 3.1. This enhancement was partly introduced in latest HexSight Version 3.1 builds.
- **New Shading Consistency ordering mode.** The Locator can now output instances according to their Shading Consistency with respect to the Model, beginning with the strongest hypothesis. In this mode, the hypothesis strength is modulated by a computation of the shading consistency of each hypothesis relative to model shading. This ordering mode can be particularly interesting for locating BGA dot patterns. This enhancement was already introduced in latest HexSight Version 3.1 builds.

Application / Process Manager Controls

- **ShowInterface method now supports new optional parameter.** The *DisplayTopMost* optional parameter can now be used to force a window to be displayed topmost. This enhancement was already introduced in latest HexSight Version 3.1 builds.
- **New Export / Import capabilities.** Methods have been added to the *HSProcessManager* control to support export and import configuration operations. These new capabilities are

available in the GUI and also programmatically using the *ExportConfiguration*, *ImportConfiguration* and *InquireConfiguration* methods. This enhancement was already introduced in latest HexSight Version 3.1 builds.

- **Processes Availability.** It is now possible to programmatically inquire about a specific process availability using the *ProcessAvailable* property. If the specified process is not part of the active license (hardware key), the property will return false. This enhancement was already introduced in latest HexSight Version 3.1 builds.

Class Library

- **HSArc, HSLine and HSPoint object types added.** Since the new Finder and Builder tools now generate entities in the runtime database, new object type were defined to access these new structures. The *HSDatabase* object was also modified accordingly and the following properties/methods were added: *Arc*, *Line*, *Point* and the corresponding *Add*, *Copy*, *Remove* and *Rename* methods.
- **HSDatabase now supports CopyView method.**
- **HSScene now supports FindModelEdgePathFromEdgeI method.** This new method is related to programmatic model edition. It enables the capability to implement interactive selection from an external *HSDisplay* control.
- **HSImage now includes subpixel drawing capabilities.** Methods and properties have been added to support synthetic shape drawings. Such drawings can be very useful for creating a Model Image Mask creation or for programmatically creating a synthetic image from a CAD definition for the purpose of building a model. This enhancement was already introduced in latest HexSight Version 3.1 builds.

Display Control

- **Small Calibration Range support.** The *HSDisplay* control can now easily handle calibrated images having a small calibrated range.
- **Entity drawing and selection support.** The *HSDisplay* control now supports drawing of output entities for the new Finder and Builder tool.

Acquisition Device Tool

- **Added DirectShow device support (IEEE 1394 FireWire cameras).** Any IEEE 1394 camera with a DirectShow driver is now available in the *HSAcquisitionDevice* input section. All DirectShow standard parameters are accessible programmatically. Stream Format and Video Parameters dialogs can also be activated programmatically. The *StartPassThru* method now also supports an optional DirectShow-specific parameter.
- **Added multiple configuration selection capability.** It is now possible to select more than one active configuration both in the user interface as well as programatically, using the added *ConfigurationEnabled* property. Having multiple active configurations permits a single *Execute* command to perform an acquisition on each active configuration.

Image Processing Tool

- FFT and DCT transforms have been added.

Protection Key Tool

- New tool added to this release (introduced in latest HexSight Version 3.1 builds).
- This tool checks for the presence of a HexSight hardware key (dongle) on a running system.

Barcode Reader Tool

- New tool added to this release (introduced in latest HexSight Version 3.1 builds).
- This tool reads and extracts information from Barcode symbology.

Data Matrix Reader Tool

- New tool added to this release (introduced in latest HexSight Version 3.1 builds).
- This tool reads and extracts information from Data Matrix symbology.

OCR Fixed Font Tool

- New tool added to this release (introduced in latest HexSight Version 3.1 builds).
- This tool recognizes alphanumerical fixed-font character strings.

Arc Finder Tool

- New tool added to this release.
- This tool finds and locates circular features on objects. The coordinates of the center of the arc, the start and end angles, as well as the radius are provided.

Line Finder Tool

- New tool added to this release.
- This tool finds and locates linear features on objects. The coordinates of a point on the line, as well as the angle of the line are provided.

Point Finder Tool

- New tool added to this release.
- This tool finds and locates points on objects. The coordinates of the found point are provided.

Builder Tool

- New tool added to this release.
- This tool creates geometric entities such as points, lines or arcs from user defined parameters or by combining other geometric entities. For example, lines and arcs can be created from multiple points while points can be created by intersecting lines and/or arcs.

Installing HexSight 3.2

To avoid confusion with ActiveX controls from previous HexSight releases you should not try to install HexSight 3.2 along with older versions on the same system. If a previous release of HexSight is already installed on a system, uninstall it first before installing HexSight release 3.2. Projects created with release 3.1 and 3.0 will load on systems where release 3.2 is installed and should work without modifications (except for the enums of the Display control's PenWidth property). Projects created with release 2.0 will load on systems where release 3.2 is installed but will require modifications to the code that manages Display controls. Projects created with releases 1.1 and 1.2 will not work on systems where release 3.2 is installed. You should not try to load projects created with HexSight release 3.2 on systems where an older HexSight release is installed. Doing so might crash Visual Basic or Visual C++ and force you to reboot your system.

Application Examples

Examples of application projects are provided for the Microsoft Visual Basic 6.0 and Microsoft Visual C++ 6.0 development environments only. Three application examples have been added to this release and specific configurations demonstrating new *HSLocator* capabilities have also been added to **Locator Demo** sample application program.

Symbology Demo

- This application shows examples of symbology tool applications.

Finder Demo

- This application shows examples of Finder tools and Builder tool applications.

Model Editor Demo

- This application shows how to implement a custom Model Editor interface using new programmatic capabilities of the *HSLocator* model edition interface. It also shows how to exploit subpixel capabilities of the *HSImage* object to create a Model Mask Image.

HexSight Version 3.1 Release Notes

HexSight Environment and Class Library

- **Complete support of left-handed and right-handed image and calibrated coordinate systems.** Left-handed and right-handed image coordinate systems are now directly supported by HexSight. It is no longer necessary to provide an inverting calibration to correct the display of a left-handed image (top-left origin) in the HSDisplay and in the Locator's Model Editor. A user-accessible property for specifying the coordinate system type has been added to images (HSImage) and scenes (HSScene). The display uses this property to set the origin and direction of the Y-axis. With HexSight 3.1, a left-handed (top-left origin) as well as a right-handed (bottom-left origin) image can be calibrated into either a left-handed or a right-handed coordinate system. With a left-handed calibrated coordinate system, the Locator will return positive angle values for clockwise rotations, whereas with a right-handed coordinate system, the Locator will continue to return positive angle values for counterclockwise rotations.
- **The OriginX and OriginY properties of the HSImage class are now read/write.** This property can now be used to specify the origin of an image cropped externally to HexSight.

Locator Tool

Search Algorithm

- **Speed optimizations.** This new release will run significantly faster than version 3.0 (up to twice as fast), without compromising the robustness or accuracy.
- **New Quality ordering mode.** The Locator can now output instances according to their Match Quality, starting with the highest value. Unlike the Evidence instance ordering mode, this mode does not order the instances according to the strength of their hypotheses but rather according to the final Match Quality factor after the pose is computed at the detail level. In images where there are many actual instances of the model or a high level of ambiguity with background features, this mode will obviously be slower because all the potential instances have to be analyzed. However, in difficult applications, such as in semiconductor applications, this ordering mode can help increase the robustness to false detections. With this mode, the return of located instances as they are found, using the Start/Wait methods, is not available. The first call to Wait will wait until the entire image is searched before returning, while subsequent calls to Wait will return immediately because all the instances have already been located.
- **New Distance ordering mode.** The Locator can now order the instances in decreasing order of their distance from a user-defined calibrated or uncalibrated location in the image. This mode can be useful when instances closer to an expected location need to be prioritized, for example to locate a specific feature that can occur at many locations in the search area.

Multi-Model and Disambiguation Capabilities

- **Improved model disambiguation.** The capability to identify and discriminate between inter-model and intra-model ambiguities has been significantly improved. This should improve the robustness to false detections in multi-model applications or when a model is nearly symmetric.
- **Improved handling of model symmetries.** The way model symmetries are handled by the Locator has been completely reworked. The cost in processing time is now very minor when the Locator is configured to output all the rotational symmetries of an object.

Models Database

- **Read-only access to the scenes of the internal models database.** These scenes contain the model contours detected in the image and the model features selected from these contours. Using these objects, it is possible to directly show in a user interface the contours detected in the model image at both the outline and detail resolutions (these contours are shown in blue in the model editor) and the features selected from these contours at both resolutions. The newly added capability to access model contours and features, makes it possible to directly implement in an application's user interface, a model teach interface that contains the contour detection and feature selection settings that are available in the Model Editor. This provides the user of the application the same level of visual feedback that the Model Editor provides.

Display Control

- **Color image display.** The HSDisplay control can now display RGB color images using three separate 8-bit images. These three 8 bit images can either be acquired externally to HexSight or internally using a RGB frame grabber such as the Matrox Meteor II/MC.
- Brightness and contrast properties have been added to control the appearance of a grey-scale image shown in a Display control. This setting does not alter the actual data of the image; it only modifies the palette used to display the image.
- To avoid conflict with other HexSight properties, the hsPenWidth enums of the ScenePenWidth property have been renamed to: hsPenWidthNone, hsPenWidthThin and hsPenWidthThick

Image Histogram Tool

- New tool added to this release.
- This tool builds the histogram of a region of interest on a grey-scale image and computes global statistics on greylevel values such as average, median, minimum, maximum, range, etc.

Image Sharpness Tool

- New tool added to this release.

- This tool computes a sharpness value for a region of interest on a grey-scale image.

Frame Grabbers

Support for the following frame grabbers has been added.

- Matrox Orion
- Coreco Imaging Technology PC-DIG

Installing HexSight 3.1

To avoid confusion with ActiveX controls from previous HexSight releases you should not try to install HexSight 3.1 along with older versions on the same system. If a previous release of HexSight is already installed on a system, uninstall it first before installing HexSight release 3.1. Projects created with release 3.0 will load on systems where release 3.1 is installed and will work without modifications (except for the enums of the Display control's PenWidth property). Projects created with release 2.0 will load on systems where release 3.1 is installed but will require modifications to the code that manages Display controls. Projects created with releases 1.1 and 1.2 will not work on systems where release 3.1 is installed. You should not try to load projects created with HexSight release 3.1 on systems where an older HexSight release is installed. Doing so might crash Visual Basic or Visual C++ and force you to reboot your system.

Application Examples

Examples of application projects are provided for the Microsoft Visual Basic 6.0 and Microsoft Visual C++ 6.0 development environments only. Two new application examples have been added to this release.

Contour Draw Demo

- Using the HSScene object to directly access the contour data generated by the Locator, this application shows how to manually draw, into a standard Visual Basic or Visual C++ graphics object, the detected contours, the model contours and the instance contours.

Contour Detection

- This application shows how to access the calibrated and uncalibrated contour data extracted by the Locator from a grey-scale image.
- This 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 process can be used, for example, to detect dots during a custom calibration procedure using a non-standard target.

HexSight Version 3.0 Release Notes

Documentation

The Locator section of the User Guide has been extensively updated. In particular, the Understanding Locator Concepts section has been improved and includes more information on how the Locator's algorithm works and how to configure it optimally.

Demonstration Mode

HexSight 3.0 includes a demonstration mode. If no hardware protection key is detected when an application is launched, HexSight automatically enters into demonstration mode. A pop-up window appears telling the user that the hardware protection key was not found and that the software will run in the demonstration mode. The application will then pause for approximately 10 seconds. After this delay, the application will resume after the user clicks OK. This pop-up window will appear again at random intervals during execution. In the demonstration mode, application development capabilities of the HexSight environment are deactivated.

Locator Tool

Except for a few contour detection and search parameters that have been removed and/or changed, the user interface and the model editor remain mostly unchanged. The internal structures and algorithms of the Locator in HexSight 3.0 have however been significantly improved. The main improvements are:

Search Algorithm

- The Locator uses leaner internal data structures. It requires less memory to run.
- The Locator uses an entirely new search engine based on the exhaustive search algorithm of version 2.0.
- The effect of the Recognition effort slider has been modified. At its default center position, the search algorithm has a behavior and performance equivalent to the Exhaustive search setting of version 2.0. This slider can be moved towards the Exhaustive setting to improve the robustness in very difficult images. At the Exhaustive setting, the search algorithm of version 3.0 is significantly more robust than the exhaustive mode of version 2.0. With cleaner images, it is also possible to move the Recognition slider towards the Quick setting. This will usually significantly reduce search times, especially in applications that must deal with wide scale factor ranges. Even at a quicker setting, the search algorithm of version 3.0 is generally more robust than the HexSight Locator 2.0 was at its normal Recognition effort level. For most applications, the default center position of the Recognition slider is recommended.
- As opposed to version 2.0, the new search algorithm does not use separate feature extraction and object location steps. Feature detection parameters and Feature Scene display capabilities have therefore been removed from both the user interface and the ActiveX programming interface.

- Alignment tolerance parameters and alignment quality factors have been removed from both the user interface and the ActiveX programmatic interface.
- Minimum Feature Length Percentage and Minimum Object Length Percentage thresholds have been merged into a single threshold called Minimum Model Percentage.
- To release the memory used by the Locator, two new methods were added. The CompactMemory method will compact all the internal structures used by the Locator to their minimal size, without unlearning the models. The next execution will not be slower. The Unlearn method will clear all the structures used by the Locator, including the learned models. The next execution will be slower because the models will be re-learned.

Automatic Model Teach Algorithm

- The automatic teach algorithm of the Model Editor has been significantly improved, especially the default Normal setting for Feature Selection. It is now better matched to the search algorithm. A proper amount of features are now extracted to reliably recognize and locate the object. The “All” setting can still be used for cases where all the contour features must be added to the model. This mode is generally useful in electronics/semiconductor applications.
- After they are taught, models can optionally be cropped to reduce the size of Model database. Cropping reduces the size of the model's grey-scale image to the portion of image contained in the model's bounding area. Only the cropped image is saved to the Model database and it is no longer possible to re-teach the Model on the original grey-scale image.

Multi-Threaded Execution

- The Locator supports a new multi-threaded execution mode. In this mode, the Locator runs the search algorithm in its own thread without blocking the main thread of the application, which is useful in pick-and-place applications where the instances can be processed as soon as they are found without having to wait for the entire Locator process to complete. This mode is only available through programmatic access. In fact, there is nothing to configure to activate this mode. It is automatically activated by using the Start and Wait methods to control the execution instead of calling the Execute method.
- The time needed to recognize and locate each instance has been added to the Results Grid and is also available through the instance properties in the ActiveX interface. The time needed to locate the first instance is usually longer because it includes all of the low-level image preprocessing.
- Configurable instance ordering has been added to the search algorithm. At the default Evidence setting, the instances are ordered according to their hypothesis strength. The instances can also be output in the order they appear in the image, starting with the leftmost, the rightmost, the topmost or the bottommost. This feature is particularly interesting for pick-and-place applications where parts that are farther down a conveyor must be picked first.

- The Results Viewer includes an optional dynamic update mode. This mode enables the user to view the instances as they are recognized. Since this mode slows the search algorithm it should only be used for application troubleshooting.
- Elapsed time properties include separate fields for Learn, Search and Total elapsed times. Learn time is the additional time required on the first execution after a change in parameters. Search time is the total time used by the search algorithm. Total time is the total time elapsed for the execution of the Locator process, including the overhead.

Multi-Model and Multi-Instances Capabilities

- For multiple model applications, the model disambiguation capabilities of the search algorithm have been significantly enhanced. The Locator can now automatically discriminate between two very similar models, even though the disambiguation features are small with respect to the total size of the object. The use of required features is still possible but not necessary for reliable model disambiguation.
- For multiple instances applications, the search algorithm is more robust in images in which multiple objects may touch and/or overlap.

Contour Detection

- The contrast thresholding mode for the contour detection step has been modified. It now features 4 different operating modes. The Locator provides 3 adaptive threshold modes: Low, Normal and High sensitivity, which automatically adjust the contrast threshold to lighting variations. The default Normal sensitivity mode is recommended for most applications. The fourth mode is a Fixed value threshold, useful in controlled environment applications where only contours above a predefined threshold should be processed. As with version 2.0, the contrast thresholding mode for a given model is set in the Model Editor when the model is built but can be overridden at runtime in the Contour panel of the Locator properties window.

Model Database Sharing

- Multiple Locators can now share a same models database in memory. This is useful for applications in which the same models must be searched in different areas and/or with different parameters. Instead of having each Locator keep its own copy of the models database, handles can be used to share one Locator's models database with other Locators.
- The models database handle can also be used to directly display the models contained in a Locator's Model database in a Display control (See the ModelDemo application example).

Display Control

The Display control has been extensively modified in version 3.0. Most of the internal structures have been redesigned to be faster and to use less memory. The ActiveX programming interface has been modified to take advantage of the new internal structures. The main improvements are:

- The Display uses leaner internal data structures. It requires less memory to run.
- The image/scene selection interface has been simplified and improved. Applications developed with HexSight 2.0 will have to be modified slightly to work with the new Display. The old image/scene selection properties (Data, SelectedImageView, SelectedImage, SelectedSceneView, and SelectedScene) were replaced by the following properties:

ImageDatabase(Index): The handle of the database containing the image to display

ImageViewName(Index): The name of the view containing the image to display

ImageName(Index): The name of the image to display

SceneDatabase(Index): The handle of the database containing the scene to display

SceneViewName(Index): The name of the view containing the scene to display

SceneName(Index): The name of the scene to display

- The AskData even has been removed from the ActiveX interface because now the database handle is individually defined for each image/scene to display.
- Since the database handle is individually set for each item to display, images and scenes contained in different databases can now be displayed in the same Display.
- The color selection interface for scenes, markers, etc. has been simplified. All colors are now specified directly using RGB values. Individual color types such as hsMarkerColor have been replaced by a more intuitive set of hsColor RGB constants (hsRed, hsGreen, hsYellow, etc.).
- Images can now be displayed in user-selectable palettes.

Display Control Example

An example of the modifications required to convert the display management code of a HexSight 2.0 for HexSight 3.0 is given below:

Original display management code in HexSight 2.0 (Visual Basic):

```
Set Acquisition =  
ApplicationControl.ProcessManager.Process("Acquisition")  
Set Locator = ApplicationControl.ProcessManager.Process("Locator")  
  
ApplicationDisplay.Data = ApplicationControl.Database.Handle  
ApplicationDisplay.SelectedImageView(0) = Acquisition.OutputView  
ApplicationDisplay.SelectedImage(0) = Acquisition.OutputGreyScaleImage  
ApplicationDisplay.SelectedSceneView(0) = Locator.OutputInstanceSceneView  
ApplicationDisplay.SelectedScene(0) = Locator.OutputInstanceScene  
  
ApplicationDisplay.MarkerColor("Axes0") = hsMarkerColor2  
ApplicationDisplay.MarkerColor("Label0") = hsMarkerColor1
```

Converted display management code in HexSight 3.0 (Visual Basic):

```
Set Acquisition =  
ApplicationControl.ProcessManager.Process("Acquisition")
```

```
Set Locator = ApplicationControl.ProcessManager.Process("Locator")
```

```
ApplicationDisplay.ImageDatabase(0) = ApplicationControl.Database.Handle
ApplicationDisplay.ImageViewName(0) = Acquisition.OutputView
ApplicationDisplay.ImageName(0) = Acquisition.OutputGreyScaleImage
ApplicationDisplay.SceneDatabase(0) = ApplicationControl.Database.Handle
ApplicationDisplay.SceneViewName(0) = Locator.OutputInstanceSceneView
ApplicationDisplay.SceneName(0) = Locator.OutputInstanceScene
```

```
ApplicationDisplay.MarkerColor("Axes0") = hsGreen
ApplicationDisplay.MarkerColor("Label0") = hsRed
```

Pattern Locator Tool

The search algorithm of the grey-scale Pattern Locator has been significantly improved. It still does not support rotated patterns and should generally be used as a model-based inspection tool for detecting the presence of grey-scale patterns on object instances that have been located using the contour-based HexSight Locator. The main improvements are:

- The Pattern Locator now uses a much faster multi-resolution grey-scale correlation pyramidal search algorithm.
- For simple non-model based applications that use grey-scale correlation, model patterns can be loaded and saved directly into files without having to use Locator models.

Installing HexSight 3.0

To avoid confusion with ActiveX controls from previous HexSight releases you should not try to install HexSight 3.0 along with older versions on the same system. If HexSight release 1.1, 1.2 or 2.0 is already installed on a system, uninstall it first before installing HexSight release 3.0. Projects created with release 2.0 will load on systems where release 3.0 is installed but will require minor modifications to the code that manages Display controls. Projects created with releases 1.1 and 1.2 will not work on systems where release 3.0 is installed. You should not try to load projects created with HexSight release 3.0 on systems where an older HexSight release is installed. Doing so might crash Visual Basic or Visual C++ and force you to reboot your system.

Opening a HexSight 2.0 Project with Release 3.0

The code that manages Display controls will require some minor modifications (see the **Display Control** section above).

A HexSight 2.0 project containing a Locator will load correctly with release 3.0. The application will also usually work without changing the code managing the Locator and the settings of the tool. Depending on the original configuration of the HexSight 2.0 Locator, the HexSight 3.0 Locator will be configured as follows:

- If the Recognition slider was originally set to Exhaustive in version 2.0, it will be set to the default center setting. Otherwise, it will be set two clicks towards the Quick setting.
- The Minimum Feature Length Percentage originally set in version 2.0 will be used for Minimum Model Percentage in version 3.0.

- Models built in version 2.0 with Contrast Sensitivity values in the 0-79 range will use the Low Sensitivity Adaptive contrast threshold at runtime. Models built with Contrast Sensitivity values in the 80-89 range will use the Normal Sensitivity Adaptive contrast threshold at runtime. Models built using higher values for Contrast Sensitivity will use the High Sensitivity Adaptive contrast threshold.
- Feature Scene parameters will be ignored.
- Alignment Tolerance parameters will be ignored.

A HexSight 2.0 project containing a Pattern Locator will load correctly with release 3.0. The application will also usually work without changing the code managing the Pattern Locator nor the settings of the tool.

Application Examples

Examples of application projects are provided for the Microsoft Visual Basic 6.0 and Microsoft Visual C++ 6.0 development environments only.