



Reference Manual



Copyright

Copyright (c) 1998-2015 by LMI Technologies All Rights Reserved. No part of this publication may be reproduced or transmitted in any form without the written permission of LMI Technologies

Notice: Effects of Software License Agreement

The license agreement included in the HexSight software package specifies how you may use the product. Any unauthorized duplication of HexSight, including this document, is strictly prohibited and violates the Software License Agreement.

Notice

LMI Technologies shall not be responsible for any errors that may appear in this document and shall not be liable for any damages arising out of or related to this document or the information contained in it, even if LMI Technologies has been advised of the possibility of such damages.

Trademark Information

HexSight is a trademark of LMI Technologies

The following are trademarks or registered trademarks of the associated company:

.NET, ActiveX, COM, C++, C#, DirectShow, DirectX, MFC, Notepad, Visual Basic, Visual C++, Visual Studio, Windows, Windows XP, Windows Vista, HTML Help: Microsoft Corporation

Acrobat Reader, PDF: Adobe Systems Incorporated

DCAM, IIDC: 1394 Trade Association, Institute of Electrical and Electronics Engineers

FireWire: Apple Computer, Inc.

GenICam: European Machine Vision Association

GigE Vision: Automated Imaging Association

GretagMacbeth, ColorChecker: X-Rite, Inc.

Pentium, MMX: Intel Corporation

Gocator: LMI Technologies

Corporate Information

LMI Technologies
Corporate Headquarters
1673 Cliveden Ave.
Delta, British Columbia
Canada V3M 6V5

Phone: +1 604 636 1011
Fax: +1 604 516 8368
Internet: www.hexsight.com
Technical support: support@hexsight.com

Table of Contents

Table of Contents	1
Acquisition Device	59
Properties	59
Calibration	59
CalibrationConfigurationName	59
CalibrationDescription	60
CalibrationMethod	60
CalibrationMethodName	61
CalibrationParameter	62
CalibrationParameterCount	64
CalibrationParameterDescription	65
CalibrationParameterEditable	66
CalibrationParameterName	67
ColorCalibration	67
ColorCalibConfigurationName	68
ColorCalibDescription	68
ColorCalibMethod	69
ColorCalibMethodName	70
ColorCalibParameter	70
ColorCalibParameterCount	71
ColorCalibParameterDescription	71
ColorCalibParameterEditable	72
ColorCalibParameterName	72
ConfigurationCount	73
ConfigurationDefault	73
ConfigurationEnabled	74
ConfigurationIndex	75
ConfigurationName	75
Database	76
DatabasePassword	76
Description	76
ElapsedTime	76
EmulationDatabaseFileName	76
EmulationImageCount	77
EmulationImageEnabled	78
EmulationImageName	78
EmulationNextImageIndex	79
EmulationNextImageName	80
ErrorCode	81
ErrorDescription	81
ErrorObject	81
GrabberDCAMParameter	81
GrabberDCAMParameterSupport	86
GrabberDCAMModeSupport	92
GrabberDepict4x1InputPort	93
GrabberDepict4x1OutputPort	94
GrabberDepict4x1Parameter	94

GrabberDirectShowParameter	97
GrabberDirectShowParameterRange	101
GrabberDt3152OutputPort	103
GrabberDt3152Parameter	103
GrabberDt3155OutputPort	105
GrabberDt3155Parameter	105
GrabberDt3157InputPort	107
GrabberDt3157OutputPort	107
GrabberDt3157Parameter	108
GrabberGocatorParameter	109
GrabberGocatorParameterCategory	110
GrabberGocatorParameterFlags	110
GrabberGocatorParameterImpact	110
GrabberGocatorParameterIndex	111
GrabberGocatorParameterMax	112
GrabberGocatorParameterMin	112
GrabberGocatorParameterName	113
GrabberGocatorParametersCount	113
GrabberGocatorParameterType	113
GrabberItilcAsyncInputPort	114
GrabberItilcAsyncOutputPort	114
GrabberItilcAsyncParameter	115
GrabberItilcAsyncPortParameter	116
GrabberItpcDigInputPort	118
GrabberItpcDigOutputPort	118
GrabberItpcDigParameter	119
GrabberItpcDigPortParameter	121
GrabberItpcVisionInputPort	122
GrabberItpcVisionOutputPort	123
GrabberItpcVisionParameter	123
GrabberItpcVisionPortParameter	124
GrabberMake	126
GrabberMatroxMeteor2dioUserBit	127
GrabberMatroxMeteor2dParameter	128
GrabberMatroxMeteor2dioUserBit	130
GrabberMatroxMeteor2mcioUserBit	131
GrabberMatroxMeteor2Parameter	132
GrabberMatroxMeteor2mcParameter	134
GrabberMatroxOrionIoUserBit	137
GrabberMatroxOrionParameter	138
GrabberModel	140
GrabberGigEParameter	141
GrabberGigEParameterCategory	141
GrabberGigEParameterFlags	142
GrabberGigEParameterImpact	142
GrabberGigEParameterIndex	142
GrabberGigEParameterMax	143
GrabberGigEParameterMin	144
GrabberGigEParameterName	144
GrabberGigEParametersCount	144
GrabberGigEParameterType	145
GrabberPx610aParameter	145

GrabberPxc200InputPort	149
GrabberPxc200OutputPort	149
GrabberPxc200Parameter	150
GrabberSonylCamParameter	157
ImageCoordinateSystemType	163
InputDevice	163
InputDeviceAvailable	164
InputDeviceCount	164
InputDeviceDescription	165
InputDeviceName	165
InputDeviceType	165
Name	166
OutputImage	166
OutputView	167
PassThruRunning	168
SelectionImage	168
SelectionView	168
TimeStampHigh	169
TimeStampLow	169
TypeName	170
Version	170
Methods	171
AboutBox	171
AppendImageToEmulationDatabase	171
ControlPropertyExchange	172
CreateConfiguration	172
DeleteConfiguration	172
EmulationDeleteImage	172
EmulationDisableAll	173
EmulationEnableAll	173
EmulationExport	174
EmulationImport	175
Execute	175
GetDebugParameter	176
GetDispatchPointer	176
GrabberDepict4x1GetBusyCount	176
GrabberDepict4x1ResetBusyCount	177
GrabberDirectShowDialog	178
GrabberPx610aFireStrobe	178
GrabberPxc200FireStrobe	179
HideCalibrationDialog	180
HideColorCalibrationDialog	180
HideProperties	181
HideRealTimeDisplay	181
ImageToWorld	182
LoadCalibration	182
LoadColorCalibration	183
LoadConfiguration	183
LoadEmulationDatabase	183
LoadGrabberParameters	184
NewEmulationDatabase	184
SaveCalibration	185

SaveColorCalibration	186
SaveConfiguration	186
SaveEmulationDatabase	186
SaveGrabberParameters	187
SetDebugParameter	187
ShowCalibDistortionLUTDialog	187
ShowCalibDistortionModelDialog	188
ShowCalibPerspectiveDialog	189
ShowCalibXYScaleDialog	189
ShowColorCalibrationDialog	190
ShowProperties	191
ShowRealTimeDisplay	192
Start	192
StartPassThru	193
StopPassThru	194
ValidateCalibration	195
ValidateColorCalibration	196
Wait	196
WorldToImage	197
Events	198
AskDatabase	198
DatabaseChange	198
Done	198
Error	198
ItiInportInterrupt	198
PropertiesChange	198
SingleExecute	198
HSCalibrationDistortionInterface	199
Properties	199
AverageImageError	199
AveragePixelHeight	199
AveragePixelWidth	200
AverageWorldError	200
AxisXPositionX	200
AxisXPositionY	201
AxisYPositionX	202
AxisYPositionY	202
CalibrationDotCount	203
CalibrationDotPositionXImage	203
CalibrationDotPositionXWorld	204
CalibrationDotPositionYImage	204
CalibrationDotPositionYWorld	205
DetectionCount	206
DisplayContourSceneEnabled	206
DisplayImageEnabled	206
DisplayMode	207
DotConformity	207
DotPitch	207
EdgeSensitivity	208
MaximumDotRadius	208
MaximumImageError	209

MaximumWorldError	209
MinimumDotRadius	209
OriginPositionX	210
OriginPositionXWorld	210
OriginPositionY	211
OriginPositionYWorld	211
UnitsLength	212
ValidationDotCount	212
ValidationDotPositionXImage	213
ValidationDotPositionXWorld	213
ValidationDotPositionYImage	214
ValidationDotPositionYWorld	215
WorldCoordinateSystemType	215
Methods	216
Detect	216
EndDialog	216
GetDispatchPointer	217
Hide	217
Show	217
Solve	217
Events	218
HSCalibrationPerspectiveInterface	219
Properties	219
AveragePixelHeight	219
AveragePixelWidth	219
CornerApproximatePositionX	220
CornerApproximatePositionY	220
CornerDetectorHeight	221
CornerDetectorPositionX	222
CornerDetectorPositionY	223
CornerDetectorWidth	223
DetectionCount	224
DetectionMode	224
LockOnCornersEnabled	225
PositionXImage	225
PositionXWorld	226
PositionYImage	227
PositionYWorld	227
UnitsLength	228
WorldCoordinateSystemType	229
Methods	229
Detect	229
EndDialog	230
GetDispatchPointer	230
Hide	230
Show	231
Solve	231
Events	231
HSCalibrationXYScaleInterface	232
Properties	232

AveragePixelHeight	232
AveragePixelWidth	232
DetectionCount	233
DetectionMode	233
HeightImage	234
HeightWorld	234
HorizontalCaliperHeight	234
HorizontalCaliperPositionX	235
HorizontalCaliperPositionY	235
HorizontalCaliperWidth	235
UnitsLength	236
VerticalCaliperHeight	236
VerticalCaliperPositionX	237
VerticalCaliperPositionY	237
VerticalCaliperWidth	237
WidthImage	238
WidthWorld	238
WorldCoordinateSystemType	239
Methods	239
Detect	239
EndDialog	240
GetDispatchPointer	240
Hide	240
Show	241
Solve	241
Events	241
HSColorCalibrationFCMInterface	242
Properties	242
Methods	242
Detect	242
EndDialog	242
GetDispatchPointer	243
Hide	243
Show	243
ShowModal	244
Solve	244
Events	244
Application	245
Properties	245
Database	245
Description	245
DisplayMessageOnError	245
ErrorObject	246
Explorer	246
ExplorerTabVisible	246
InterfaceVisible	246
Name	247
ProcessManager	247
ProcessManagerTabVisible	247
UpdateDisplayOnDone	248

UpdateExplorerOnDatabaseChange	248
Version	249
Methods	249
AboutBox	249
ControlPropertyExchange	249
GetDispatchPointer	250
GetDebugParameter	250
HideInterface	250
HideProperties	250
LoadConfiguration	250
Refresh	251
SaveConfiguration	251
SetDebugParameter	251
ShowInterface	251
ShowProperties	252
Events	252
AskDatabase	252
DatabaseChange	252
Done	252
Error	253
ExplorerSelectionChange	253
ProcessManagerModeChange	253
PropertiesChange	253
ArcCaliper	254
Properties	254
BilinearInterpolationEnabled	254
CalibratedUnitsEnabled	254
CoordinateSystem	255
Database	256
DatabasePassword	256
Description	256
Edge1Constraints	256
Edge1Magnitude	257
Edge1MagnitudeConstraint	257
Edge1MagnitudeScore	258
Edge1Polarity	259
Edge1PositionConstraint	259
Edge1PositionScore	260
Edge1PositionX	261
Edge1PositionY	261
Edge1Radius	262
Edge1Rotation	262
Edge1Score	263
Edge1ScoreThreshold	264
Edge2Constraints	264
Edge2Magnitude	265
Edge2MagnitudeConstraint	265
Edge2MagnitudeScore	266
Edge2Polarity	267
Edge2PositionConstraint	267
Edge2PositionScore	268

Edge2PositionX	269
Edge2PositionY	269
Edge2Radius	270
Edge2Rotation	270
Edge2Score	271
Edge2ScoreThreshold	272
EdgeFilterHalfWidth	272
EdgeMagnitudeThreshold	273
ElapsedTime	273
ErrorObject	273
InputImage	274
InputView	274
InputInstanceScene	274
InputInstanceView	275
InputModelScene	275
InputModelView	275
InstanceSelectionEnabled	276
InstanceSelectionIndex	276
ModelTransformEnabled	277
Name	277
PairCount	277
PairIndex	278
PairName	278
PairPositionX	279
PairPositionY	280
PairRadius	280
PairRotation	281
PairScore	282
PairSize	282
ProcessingFormat	283
ProjectionMode	284
SamplingStep	284
SamplingStepCustom	285
SamplingStepCustomEnabled	285
SamplingStepDefault	285
ToolOpening	286
ToolPositionX	286
ToolPositionY	287
ToolRadius	287
ToolRotation	288
ToolThickness	289
TransformRotation	289
TransformScale	290
TransformTranslationX	290
TransformTranslationY	291
TypeName	291
UnitsLength	291
UnitsOrientation	292
Version	292
Methods	293
AboutBox	293
AddPair	293

ControlPropertyExchange	294
Execute	294
GetDebugParameter	294
GetDispatchPointer	294
GetProcessingFormatEnabled	294
HideProperties	295
LoadConfiguration	295
RemovePair	295
SaveConfiguration	296
SetDebugParameter	296
ShowProperties	296
Events	297
AskDatabase	297
DatabaseChange	297
Done	297
Error	297
PropertiesChange	297
SingleExecute	297
Arc Edge Locator	298
Properties	298
BilinearInterpolationEnabled	298
CalibratedUnitsEnabled	298
Constraints	299
CoordinateSystem	299
Database	300
DatabasePassword	300
Description	300
EdgeCount	301
EdgeMagnitude	301
EdgeMagnitudeScore	301
EdgePositionScore	302
EdgePositionX	302
EdgePositionY	303
EdgeRadius	303
EdgeRotation	304
EdgeScore	305
ElapsedTime	305
ErrorObject	305
FilterHalfWidth	306
InputImage	306
InputView	307
InputInstanceScene	307
InputInstanceView	307
InputModelScene	308
InputModelView	308
InstanceSelectionEnabled	308
InstanceSelectionIndex	309
MagnitudeConstraint	309
MagnitudeThreshold	310
ModelTransformEnabled	310
Name	311

Polarity	311
PositionConstraint	312
ProcessingFormat	312
ProjectionMode	313
SamplingStep	313
SamplingStepCustom	314
SamplingStepCustomEnabled	314
SamplingStepDefault	315
ScoreThreshold	315
SortResultsEnabled	315
ToolOpening	316
ToolPositionX	316
ToolPositionY	317
ToolRadius	317
ToolRotation	318
ToolThickness	319
TransformRotation	319
TransformScale	320
TransformTranslationX	320
TransformTranslationY	321
TypeName	321
UnitsLength	321
UnitsOrientation	322
Version	322
Methods	323
AboutBox	323
ControlPropertyExchange	323
Execute	323
GetDebugParameter	324
GetDispatchPointer	324
GetProcessingFormatEnabled	324
HideProperties	324
LoadConfiguration	324
SaveConfiguration	325
SetDebugParameter	325
ShowProperties	325
Events	326
AskDatabase	326
DatabaseChange	326
Done	326
Error	326
PropertiesChange	326
SingleExecute	326
Arc Finder	327
Properties	327
ArcAverageContrast	327
ArcCenterPointPositionX	327
ArcCenterPointPositionY	328
ArcEndAngle	328
ArcEndPointPositionX	328
ArcEndPointPositionY	329

ArcFitQuality	329
ArcFound	330
ArcMatchQuality	330
ArcMustBeTotallyEnclosed	330
ArcRadius	331
ArcStartAngle	331
ArcStartPointPositionX	331
ArcStartPointPositionY	332
CalibratedUnitsEnabled	332
ConformityTolerance	333
ContrastThreshold	333
ContrastThresholdMode	334
CoordinateSystem	335
Database	335
DatabasePassword	336
DefaultConformityTolerance	336
Description	336
ElapsedTime	336
ErrorObject	337
FitMode	337
InputImage	338
InputGreyScaleImage	338
InputView	338
InputGreyScaleView	339
InputInstanceScene	339
InputInstanceView	339
InputModelScene	340
InputModelView	340
InstanceSelectionEnabled	340
InstanceSelectionIndex	341
MaximumAngleDeviation	341
MaximumConformityTolerance	342
MessageCount	342
MessageDescription	342
MessageNumber	343
MinimumArcPercentage	343
MinimumConformityTolerance	344
ModelTransformEnabled	344
Name	345
OutputEntity	345
OutputEntityEnabled	345
OutputEntityView	346
PolarityMode	346
PositioningLevel	347
ProcessingFormat	347
SearchMode	348
ShowEdges	348
ShowEntityFound	349
ShowEntityFoundAs	349
ShowImage	350
ShowGreyScaleImage	350
ShowSearchArea	350

SubsamplingLevel	351
ToolGuidelineOffset	351
ToolOpening	351
ToolPositionX	352
ToolPositionY	352
ToolRadius	353
ToolRotation	353
ToolThickness	354
TransformRotation	355
TransformScale	355
TransformTranslationX	356
TransformTranslationY	356
TypeName	357
UnitsLength	357
UnitsOrientation	357
UseDefaultConformityTolerance	358
Version	358
Methods	359
AboutBox	359
ControlPropertyExchange	359
Execute	359
GetDebugParameter	359
GetDispatchPointer	359
GetProcessingFormatEnabled	360
HideProperties	360
LoadConfiguration	360
SaveConfiguration	360
SetDebugParameter	361
ShowProperties	361
Events	362
AskDatabase	362
DatabaseChange	362
Done	362
Error	362
PropertiesChange	362
SingleExecute	362
Barcode Reader	363
Properties	363
BarcodeAngle	363
BarcodeCornerX	363
BarcodeCornerY	364
BarcodeDecodeTime	365
BarcodeFoundCount	366
BarcodeStatusCode	367
BarcodeStatusCodeString	368
BarcodeString	369
BarcodeToDecode	370
BarcodeType	370
BarcodeTypeEnabled	372
BarcodeTypeString	373
BilinearInterpolationEnabled	373

CalibratedUnitsEnabled	374
ChecksumEnabled	374
CoordinateSystem	375
Database	376
DatabasePassword	376
DecodeLeftToRightEnabled	376
Description	377
ElapsedTime	378
ErrorObject	378
HorizontalProbeSpacing	378
InputImage	379
InputGreyScaleImage	379
InputView	380
InputGreyScaleView	380
InputInstanceScene	380
InputInstanceView	381
InputModelScene	381
InputModelView	381
InstanceSelectionEnabled	382
InstanceSelectionIndex	382
MaximumBarWidth	382
MaximumCodeHeight	383
MaximumCodeLength	384
MaximumNumberOfBars	384
MinimumBarWidth	385
MinimumCodeHeight	385
MinimumCodeLength	386
MinimumEdgeStrength	386
MinimumNumberOfBars	387
MinimumQuietZone	387
ModelTransformEnabled	388
Name	388
Polarity	389
PolarityEnabled	390
ProcessingAlgorithm	391
ProcessingSize	392
SamplingStep	392
SamplingStepCustom	393
SamplingStepCustomEnabled	393
SamplingStepDefault	394
SearchDirection	394
StartStopEnabled	395
Timeout	396
ToolHeight	396
ToolPositionX	396
ToolPositionY	397
ToolRotation	397
ToolWidth	398
TransformRotation	399
TransformScale	399
TransformTranslationX	400
TransformTranslationY	400

TypeName	401
UnitsLength	401
UnitsOrientation	401
Version	402
VerticalProbeSpacing	402
Methods	403
AboutBox	403
ControlPropertyExchange	403
Execute	403
GetDebugParameter	403
GetDispatchPointer	404
HideProperties	404
Learn	404
LoadConfiguration	405
SaveConfiguration	405
SetDebugParameter	405
ShowProperties	405
Events	406
AskDatabase	406
DatabaseChange	406
Done	406
Error	406
PropertiesChange	406
SingleExecute	406
Blob Analyzer	407
Properties	407
BilinearInterpolationEnabled	407
BlobArea	407
BlobBoundingBoxBottom	408
BlobBoundingBoxCenterX	409
BlobBoundingBoxCenterY	409
BlobBoundingBoxHeight	410
BlobBoundingBoxLeft	411
BlobBoundingBoxRight	411
BlobBoundingBoxRotation	412
BlobBoundingBoxTop	413
BlobBoundingBoxWidth	413
BlobChainCode	414
BlobChainCodeDeltaX	415
BlobChainCodeDeltaY	415
BlobChainCodeLength	416
BlobChainCodeStartX	416
BlobChainCodeStartY	417
BlobConvexPerimeter	418
BlobCount	418
BlobElongation	419
BlobExtentBottom	419
BlobExtentLeft	420
BlobExtentRight	421
BlobExtentTop	421
BlobGreyLevelMaximum	422

BlobGreyLevelMean	422
BlobGreyLevelMinimum	423
BlobGreyLevelRange	423
BlobGreyLevelStdDev	424
BlobHoleCount	424
BlobInertiaMaximum	425
BlobInertiaMinimum	425
BlobInertiaXAxis	426
BlobInertiaYAxis	427
BlobIntrinsicBoundingBoxBottom	428
BlobIntrinsicBoundingBoxCenterX	428
BlobIntrinsicBoundingBoxCenterY	429
BlobIntrinsicBoundingBoxHeight	430
BlobIntrinsicBoundingBoxLeft	431
BlobIntrinsicBoundingBoxRight	431
BlobIntrinsicBoundingBoxRotation	432
BlobIntrinsicBoundingBoxTop	433
BlobIntrinsicBoundingBoxWidth	433
BlobIntrinsicExtentBottom	434
BlobIntrinsicExtentLeft	435
BlobIntrinsicExtentRight	436
BlobIntrinsicExtentTop	436
BlobPositionX	437
BlobPositionY	438
BlobPrincipalAxesRotation	438
BlobRawPerimeter	439
BlobRoundness	440
CalibratedUnitsEnabled	440
ChainCodeResultsEnabled	441
ClearOutputBlobImageEnabled	441
CoordinateSystem	441
Database	442
DatabasePassword	442
Description	442
ElapsedTime	443
ErrorObject	443
ExtrinsicBoxResultsEnabled	443
ExtrinsicInertiaResultsEnabled	444
GreyLevelResultsEnabled	444
HoleFillingEnabled	445
InputImage	445
InputView	446
InputInstanceScene	446
InputInstanceView	446
InputModelScene	447
InputModelView	447
InstanceSelectionEnabled	447
InstanceSelectionIndex	448
IntrinsicBoxResultsEnabled	448
IntrinsicInertiaResultsEnabled	449
MaximumBlobArea	449
MinimumBlobArea	450

ModelTransformEnabled	450
Name	451
OutputBlobImage	451
OutputBlobImageEnabled	451
OutputBlobView	452
PerimeterResultsEnabled	452
ProcessingFormat	453
SamplingStep	453
SamplingStepCustom	454
SamplingStepCustomEnabled	454
SamplingStepDefault	455
SegmentationDark	455
SegmentationDynamicDark	456
SegmentationDynamicInside	456
SegmentationDynamicLight	457
SegmentationDynamicOutside	457
SegmentationHSLInside	458
SegmentationHSLOutside	458
SegmentationInside	459
SegmentationLight	459
SegmentationMode	460
SegmentationOutside	460
SortBlobsBy	461
SortResultsEnabled	463
ToolHeight	463
ToolPositionX	464
ToolPositionY	464
ToolRotation	465
ToolWidth	465
TopologicalResultsEnabled	466
TransformRotation	466
TransformScale	467
TransformTranslationX	467
TransformTranslationY	468
TypeName	468
UnitsLength	469
UnitsOrientation	469
Version	470
Methods	470
AboutBox	470
ControlPropertyExchange	470
Execute	470
GetDebugParameter	471
GetDispatchPointer	471
GetProcessingFormatEnabled	471
HideProperties	471
LoadConfiguration	472
SaveConfiguration	472
SetDebugParameter	472
ShowProperties	472
Events	473
AskDatabase	473

DatabaseChange	473
Done	473
Error	473
PropertiesChange	474
SingleExecute	474
Builder	475
Properties	475
CoordinateSystem	475
Database	475
DatabasePassword	475
Description	475
ElapsedTime	476
ErrorObject	476
InputEntity	477
InputEntityCount	477
InputEntityDatabase	477
InputEntityType	478
InputEntityView	478
InputImage	479
InputGreyScaleImage	479
InputView	480
InputGreyScaleView	480
Intersected	481
Mode	481
Name	484
OutputArc	485
OutputEntity	485
OutputEntityCount	485
OutputEntityDatabase	486
OutputEntityExtrapolated	486
OutputEntityType	487
OutputEntityView	487
OutputLine	488
OutputPoint	488
Parameter	489
ParameterCount	491
ParameterName	491
ParameterType	492
ShowExtrapolations	492
ShowImage	492
ShowGreyScaleImage	493
ShowInputEntities	493
ShowOutputEntities	493
TypeName	494
UnitsLength	494
UnitsOrientation	495
Version	495
Methods	496
AboutBox	496
ControlPropertyExchange	496
Execute	496

GetDebugParameter	496
GetDispatchPointer	497
HideProperties	497
LoadConfiguration	497
SaveConfiguration	497
SetDebugParameter	497
ShowProperties	497
Events	498
AskDatabase	498
DatabaseChange	499
Done	499
Error	499
PropertiesChange	499
SingleExecute	499
Caliper	500
Properties	500
BilinearInterpolationEnabled	500
CalibratedUnitsEnabled	500
CoordinateSystem	501
Database	501
DatabasePassword	502
Description	502
Edge1Constraints	502
Edge1Magnitude	503
Edge1MagnitudeConstraint	503
Edge1MagnitudeScore	504
Edge1Polarity	504
Edge1PositionConstraint	505
Edge1PositionScore	506
Edge1PositionX	507
Edge1PositionY	507
Edge1Rotation	508
Edge1Score	509
Edge1ScoreThreshold	509
Edge2Constraints	510
Edge2Magnitude	510
Edge2MagnitudeConstraint	511
Edge2MagnitudeScore	511
Edge2Polarity	512
Edge2PositionConstraint	513
Edge2PositionScore	514
Edge2PositionX	514
Edge2PositionY	515
Edge2Rotation	515
Edge2Score	516
Edge2ScoreThreshold	517
EdgeFilterHalfWidth	517
EdgeMagnitudeThreshold	518
ElapsedTime	518
ErrorObject	518
InputImage	519

InputGreyScaleImage	519
InputView	519
InputGreyScaleView	520
InputInstanceScene	520
InputInstanceView	520
InputModelScene	521
InputModelView	521
InstanceSelectionEnabled	521
InstanceSelectionIndex	522
ModelTransformEnabled	522
Name	523
PairCount	523
PairIndex	523
PairName	524
PairPositionX	524
PairPositionY	525
PairRotation	526
PairScore	527
PairSize	527
ProcessingFormat	528
SamplingStep	529
SamplingStepCustom	529
SamplingStepCustomEnabled	530
SamplingStepDefault	530
ToolHeight	530
ToolPositionX	531
ToolPositionY	531
ToolRotation	532
ToolSkew	533
ToolWidth	533
TransformRotation	533
TransformScale	534
TransformTranslationX	535
TransformTranslationY	535
TypeName	536
UnitsLength	536
UnitsOrientation	537
Version	537
Methods	537
AboutBox	537
AddPair	538
ControlPropertyExchange	538
Execute	538
GetDebugParameter	539
GetDispatchPointer	539
GetProcessingFormatEnabled	539
HideProperties	539
LoadConfiguration	539
RemovePair	540
SaveConfiguration	540
SetDebugParameter	540
ShowProperties	540

Events	541
AskDatabase	541
DatabaseChange	541
Done	542
Error	542
PropertiesChange	542
SingleExecute	542
Class Library	543
HSArc	544
Properties	544
Calibrated	544
Calibration	545
CenterPoint	545
Circle	546
Description	547
EndAngle	547
EndPoint	548
ErrorObject	548
Information	548
ModifyCount	549
Name	549
Radius	549
StartAngle	550
StartPoint	551
TypeName	551
UnitsLength	551
UnitsOrientation	552
Version	552
Methods	553
AboutBox	553
AngleWithinRange	553
CopyFormatFrom	553
GetDispatchPointer	554
Load	554
NearestPoint	554
Reset	555
Save	555
Events	556
DatabaseChange	556
Error	556
HSCalibration	557
Properties	557
AveragePixelHeight	557
AveragePixelSize	557
AveragePixelWidth	557
CalibrationPointer	558
Description	558
ErrorObject	558
ImageHeight	559
ImageWidth	559

Information	559
Parameter	559
ParameterCount	561
ParameterDescription	562
ParameterName	562
State	563
Type	563
TypeName	564
UnitsLength	564
Version	565
WorldCoordinateSystemType	565
Methods	566
AboutBox	566
Create	566
GetDispatchPointer	566
ImageToWorld	567
Load	567
Save	568
Solve	568
WorldToImage	569
Events	569
Error	569
HSDatabase	570
Properties	570
Arc	570
Description	570
ErrorObject	571
Filename	571
FirstImageName	571
FirstSceneName	572
FirstViewName	572
Handle	573
Image	573
Information	573
LastImageName	574
LastSceneName	574
LastViewName	575
Line	575
ModifyCount	575
Name	576
NextImageName	576
NextSceneName	577
NextViewName	577
Password	577
Point	578
PreviousImageName	578
PreviousSceneName	578
PreviousViewName	579
Scene	579
TypeName	580
Version	580

View	580
ViewCount	581
ViewExists	581
ViewIndex	582
ViewName	582
Methods	583
AboutBox	583
AddArc	583
AddImage	583
AddLine	584
AddPoint	585
AddView	585
CopyArc	586
CopyImage	586
CopyLine	587
CopyPoint	587
CopyScene	588
CopyView	588
GetDispatchPointer	589
Load	589
LoadConfiguration	589
Modify	589
RemoveArc	590
RemoveImage	590
RemoveLine	591
RemovePoint	591
RemoveScene	592
RemoveView	592
RenameArc	592
RenameImage	593
RenameLine	593
RenamePoint	594
RenameScene	594
RenameView	595
Reset	595
Save	596
SaveConfiguration	596
Events	596
DatabaseChange	596
Error	596
HSErrorObject	597
Properties	597
Description	597
HelpContext	597
HelpFile	597
Number	598
Source	598
TypeName	598
Version	599
Methods	599
AboutBoxCustom	599

Clear	599
Events	600
HSI Image	601
Properties	601
AveragePixelHeight	601
AveragePixelWidth	601
BackgroundColor	602
BackgroundImage	602
BoundaryBottom	603
BoundaryLeft	603
BoundaryRight	603
BoundaryTop	604
Calibration	604
Description	604
ErrorObject	605
Height	605
ImageCoordinateSystemType	605
Information	606
ModifyCount	606
Name	607
OriginX	607
OriginY	607
Pixel	608
RawImagePointer	608
ShapesCalibrated	609
SubSamplingFactor	609
Type	609
TypeName	610
UnitsLength	610
UnitsOrientation	611
Version	611
Width	612
Methods	612
AboutBox	612
AddCircleShape	612
AddRectangleShape	613
ApplyTransform	614
ArcTo	615
AttachImagePointer	616
Calibrate	616
ComputeArea	616
ComputeShapes	616
Copy	617
CopyFromSafeArray	617
CopyToSafeArray	618
Create	619
CreateEx	619
GetDispatchPointer	619
ImageToWorld	619
ImageToWorldArray	620
LineTo	620

Load	620
Modify	621
Render	622
ResetShapes	622
Save	623
SetColor	623
SetFirst	624
SetForcePositiveSign	624
StartShape	624
SubSampleFrom	625
WorldToImage	625
WorldToImageArray	626
Events	626
DatabaseChange	626
Error	626
HSLine	627
Properties	627
Angle	627
Calibrated	628
Calibration	629
Description	629
EndDelta	629
EndPoint	630
ErrorObject	630
Information	631
ModifyCount	631
Name	631
StartDelta	632
StartPoint	632
TypeName	633
UnitsLength	633
UnitsOrientation	634
VectorPoint	634
Version	635
Methods	635
AboutBox	635
CopyFormatFrom	635
GetDispatchPointer	636
Load	636
NearestPoint	636
Perpendicular	637
Reset	638
Save	638
Events	638
DatabaseChange	638
Error	639
HSPoint	640
Properties	640
Calibrated	640
Calibration	640

Description	641
ErrorObject	641
ImageX	642
ImageY	642
Information	643
ModifyCount	643
Name	643
TypeName	644
UnitsLength	644
UnitsOrientation	645
Version	645
WorldX	645
WorldY	646
Methods	647
AboutBox	647
Angle	647
CopyFormatFrom	647
Distance	648
GetDispatchPointer	648
Load	649
Reset	649
Save	649
Events	650
DatabaseChange	650
Error	650
HSScene	651
Properties	651
AveragePixelHeight	651
AveragePixelWidth	651
BoundaryBottom	652
BoundaryLeft	652
BoundaryRight	652
BoundaryTop	653
Calibration	653
Description	653
EdgeICount	653
EdgeIImageX	654
EdgeIImageXArray	655
EdgeIImageY	656
EdgeIImageYArray	658
EdgeIWorldX	659
EdgeIWorldXArray	660
EdgeIWorldY	662
EdgeIWorldYArray	663
EdgePathArea	664
EdgePathCentroidX	665
EdgePathCentroidY	666
EdgePathClosed	666
EdgePathCount	667
EdgePathFirstEdgeIIndex	668
EdgePathLastEdgeIIndex	669

EdgePathPerimeter	669
ErrorObject	670
Height	670
ImageCoordinateSystemType	671
Information	671
InstanceCount	672
InstanceFirstEdgePathIndex	674
InstanceLastEdgePathIndex	676
ModelEdgePathArea	678
ModelEdgePathCentroidX	678
ModelEdgePathCentroidY	679
ModelEdgePathCount	680
ModelEdgePathFirstEdgeIndex	680
ModelEdgePathLastEdgeIndex	682
ModelEdgePathNextEdgeIndex	682
ModelEdgePathPerimeter	683
ModelEdgePathPreviousEdgeIndex	683
ModifyCount	684
Name	684
OriginX	685
OriginY	685
Type	686
TypeName	686
UnitsLength	687
Width	687
Version	687
Methods	688
AboutBox	688
ComputeProperties	688
CopyFormatFrom	689
FindModelEdgePathFromEdge	689
GetDispatchPointer	689
ImageToWorld	690
Load	690
Modify	691
Reset	691
Save	691
WorldToImage	692
Events	692
DatabaseChange	692
Error	693
HSVView	694
Properties	694
Arc	694
ArcCount	694
ArcExists	695
ArcIndex	695
ArcName	696
Description	696
ErrorObject	696
Image	697

ImageCount	697
ImageExists	697
ImageIndex	698
ImageName	698
Information	699
Line	699
LineCount	699
LineExists	700
LineIndex	700
LineName	701
ModifyCount	701
Name	702
Point	702
PointCount	702
PointExists	703
PointIndex	703
PointName	704
Scene	704
SceneCount	704
SceneExists	705
SceneIndex	705
SceneName	706
TypeName	706
Version	706
Methods	707
AboutBox	707
AddArc	707
AddImage	707
AddLine	708
AddPoint	709
AddScene	709
GetDispatchPointer	709
Load	710
Modify	710
RemoveArc	710
RemoveImage	711
RemoveLine	711
RemovePoint	712
RemoveScene	712
RenameArc	713
RenameImage	713
RenameLine	713
RenamePoint	714
RenameScene	714
Reset	715
Save	715
Events	716
DatabaseChange	716
Error	716
Color Finder	717
Properties	717

BilinearInterpolationEnabled	717
CalibratedUnitsEnabled	717
ClearOutputFilterImageEnabled	718
ColorFilterBestMatchIndex	718
ColorFilterBestMatchName	718
ColorFilterCount	719
ColorFilterEnabled	719
ColorFilterMatchPixelCount	720
ColorFilterMatchQuality	720
ColorFilterName	721
CoordinateSystem	721
Database	722
DatabasePassword	722
Description	722
ElapsedTime	722
ErrorObject	723
ImageHeight	723
ImagePixelCount	724
ImageSubsampling	724
ImageWidth	724
InputImage	725
InputGreyScaleImage	725
InputView	725
InputGreyScaleView	726
InputInstanceScene	726
InputInstanceView	727
InputModelScene	727
InputModelView	727
InstanceSelectionEnabled	728
InstanceSelectionIndex	728
ModelTransformEnabled	728
Name	729
OutputFilterImage	729
OutputFilterImageEnabled	730
OutputFilterView	730
SamplingStep	730
SamplingStepCustom	731
SamplingStepCustomEnabled	731
SamplingStepDefault	732
ToolHeight	732
ToolPositionX	733
ToolPositionY	733
ToolRotation	734
ToolWidth	734
TransformRotation	735
TransformScale	735
TransformTranslationX	736
TransformTranslationY	736
TypeName	737
UnitsLength	737
UnitsOrientation	738
Version	738

Methods	739
AboutBox	739
AddColorFilter	739
ControlPropertyExchange	740
Execute	740
ExportColorFilter	740
GetColorFilterHSLTolerances	740
GetColorFilterHSLValues	741
GetColorFilterRGBValues	742
GetDebugParameter	742
GetDispatchPointer	742
HideProperties	742
ImportColorFilter	743
LoadConfiguration	743
RemoveColorFilter	743
SaveConfiguration	744
SetColorFilterHSLTolerances	744
SetColorFilterHSLValues	745
SetColorFilterRGBValues	745
SetDebugParameter	746
ShowProperties	746
Events	747
AskDatabase	747
DatabaseChange	747
Done	747
Error	747
PropertiesChange	747
SingleExecute	748
DataMatrix Reader	749
Properties	749
AdaptiveThresholdEnabled	749
AngleVariation	750
AverageElementSize	750
BilinearInterpolationEnabled	751
BorderIntegrity	751
CalibratedUnitsEnabled	751
CoordinateSystem	752
Database	753
DatabasePassword	753
Description	753
ElapsedTime	753
ErrorObject	753
FineTuneMethodEnabled	754
GlobalThresholdEnabled	755
InputImage	756
InputGreyScaleImage	756
InputView	756
InputGreyScaleView	757
InputInstanceScene	757
InputInstanceView	757
InputModelScene	758

InputModelView	758
InstanceSelectionEnabled	758
InstanceSelectionIndex	759
LocalThresholdEnabled	759
MatrixAngle	760
MatrixBorderMatchPercentage	761
MatrixColumns	761
MatrixColumnsEnabled	762
MatrixContrast	762
MatrixCornerX	763
MatrixCornerY	764
MatrixDamagePercentage	765
MatrixDecodeTime	766
MatrixECCLevel	766
MatrixECCLevelEnabled	767
MatrixFoundCount	767
MatrixHeight	768
MatrixHeightEnabled	769
MatrixImageStyle	769
MatrixImageStyleEnabled	770
MatrixOrientation	770
MatrixOrientationEnabled	771
MatrixPolarity	772
MatrixPolarityEnabled	772
MatrixRows	773
MatrixRowsEnabled	773
MatrixStatusCode	774
MatrixStatusCodeString	775
MatrixString	776
MatrixThreshold	777
MatrixToDecode	777
MatrixTotalErrorBits	778
MatrixWidth	779
MatrixWidthEnabled	779
MinimumEdgeStrength	780
ModelTransformEnabled	780
Name	781
ProcessingAlgorithm	781
ProcessingSize	782
SamplesPerCell	783
SamplingStep	783
SamplingStepCustom	784
SamplingStepCustomEnabled	784
SamplingStepDefault	785
SearchDirection	785
SearchSpeed	785
SizeVariation	786
Timeout	787
ToolHeight	787
ToolPositionX	787
ToolPositionY	788
ToolRotation	788

ToolWidth	789
TransformRotation	790
TransformScale	790
TransformTranslationX	791
TransformTranslationY	791
TypeName	792
UnitsLength	792
UnitsOrientation	792
Version	793
WarpMethod	793
WidthHeightRatio	794
WidthHeightRatioEnabled	794
Methods	795
AboutBox	795
ControlPropertyExchange	795
Execute	795
GetDebugParameter	796
GetDispatchPointer	796
HideLearnInterface	796
HideProperties	796
Learn	797
LoadConfiguration	798
SaveConfiguration	798
SetDebugParameter	798
ShowLearnInterface	798
ShowProperties	799
Events	800
DatabaseChange	800
Done	800
Error	800
PropertiesChange	800
SingleExecute	800
Display	801
Properties	801
Appearance	801
AutoPanEnabled	801
AxesMarkerConstraints	802
AxesMarkerLabel1	802
AxesMarkerLabel2	803
AxesMarkerLength1	803
AxesMarkerLength2	804
AxesMarkerMidAngle	805
AxesMarkerRotation	805
BorderStyle	806
Brightness	806
CalibratedUnitsEnabled	807
CircularArcMarkerConstraints	807
CircularArcMarkerOpening	808
CircularArcMarkerRadius	809
CircularArcMarkerRotation	809
CircularArcMarkerThickness	810

ColorBackground	810
ColorForeground	811
ColorGrid	812
ContextMenuEnabled	812
ContextMenuMask	813
Contrast	813
CrossHairVisible	814
Description	814
EntityColor	815
EntityDatabase	816
EntityDatabasePassword	816
EntityKValues	816
EntityName	816
EntityPenWidth	817
EntitySelectionEnabled	817
EntitySize	818
EntityViewName	818
ErrorObject	819
FocusX	819
FocusY	820
Font	820
GridVisible	820
ImageDatabase	821
ImageDatabasePassword	821
ImageName	821
ImagePalette	822
ImagePaletteColor	823
ImagePaletteCycle	824
ImagePaletteMaskMode	825
ImagePaletteTransparencyIndex	825
ImagePaletteType	826
ImageSize	827
ImageViewName	828
InteractionMode	828
InteractiveSelectionEnabled	829
IsMarkerDefined	829
LabelMarkerText	829
LabelMarkerTextAlign	830
LineMarkerConstraints	830
LineMarkerX2	831
LineMarkerY2	832
MarkerAnchorColor	832
MarkerAnchorStyle	833
MarkerCalibrated	834
MarkerColor	834
MarkerDefined	835
MarkerDisplayName	836
MarkerGroup	836
MarkerLineStyle	836
MarkerPenWidth	837
MarkerX	838
MarkerY	838

MaximalBoundaryHeight	839
MaximalBoundaryWidth	839
MultiSelectionEnabled	840
PointArrayMarkerSize	840
PointArrayMarkerX	840
PointArrayMarkerY	841
PointMarkerConstraints	842
PriorityMask	842
RectangleMarkerConstraints	843
RectangleMarkerHeight	844
RectangleMarkerRotation	844
RectangleMarkerSkew	845
RectangleMarkerWidth	846
RulersVisible	846
ScaleToFitEnabled	847
SceneColor	847
SceneDatabase	848
SceneDatabasePassword	848
SceneName	848
ScenePenWidth	849
SceneSelection	850
SceneSelectionColor	851
SceneSelectionIndex	852
SceneSelectionPenWidth	852
SceneSelectionType	853
SceneSize	854
SceneViewName	855
ScrollBarsVisible	855
StatusBarVisible	856
TargetMarkerConstraints	856
TargetMarkerRadius	857
TargetMarkerRotation	857
Title	858
TitleVisible	858
ToolBarVisible	859
UnitsLength	859
UnitsOrientation	859
Version	860
WheelOperationMode	860
Zoom	861
ZoomToPage	861
Methods	862
AboutBox	862
AddAxesMarker	862
AddCircularArcMarker	863
AddLabelMarker	864
AddLineMarker	865
AddPointArrayMarker	866
AddPointMarker	867
AddRectangleMarker	868
AddTargetMarker	869
ChangeAxesMarker	870

ChangeCircularArcMarker	871
ChangeLabelMarker	872
ChangeLineMarker	873
ChangePointArrayMarker	874
ChangePointMarker	875
ChangeRectangleMarker	876
ChangeTargetMarker	876
ClearSceneSelection	877
ControlPropertyExchange	878
DoClick	878
Drag	878
GetDispatchPointer	878
GetDisplayCoordinates	878
GetFreeBackgroundSelection	879
HideProperties	879
LoadConfiguration	879
Move	879
Refresh	879
RefreshChangedMarker	880
RefreshDisplay	880
RefreshMarker	881
RefreshMarkers	881
RemoveAllMarker	881
RemoveMarker	882
RemoveMarkerGroup	882
SaveConfiguration	883
SaveDisplay	883
SetFocus	883
SetFocusXY	883
SetMarkerPosition	884
SetPointArrayMarkerPosition	884
ShowProperties	885
Events	886
AxesMarkerChange	886
CalibratedUnitsChange	886
CircularArcMarkerChange	887
Click	887
DbClick	888
DisplayMouseDown	888
DisplayMouseMove	888
DisplayMouseUp	889
EntitySelectionChange	889
Error	889
FocusChange	890
InteractionModeChange	890
InteractiveSelectionChange	890
KeyDown	890
KeyPress	891
KeyUp	891
LineMarkerChange	891
PointMarkerChange	892
PropertiesChange	892

RectangleMarkerChange	892
TargetMarkerChange	893
ZoomChange	894
Edge Locator	895
Properties	895
BilinearInterpolationEnabled	895
CalibratedUnitsEnabled	895
Constraints	896
CoordinateSystem	896
Database	897
DatabasePassword	897
Description	897
EdgeCount	898
EdgeMagnitude	898
EdgeMagnitudeScore	898
EdgePositionScore	899
EdgePositionX	899
EdgePositionY	900
EdgeRotation	900
EdgeScore	901
ElapsedTime	901
ErrorObject	902
FilterHalfWidth	902
InputImage	903
InputView	903
InputInstanceScene	903
InputInstanceView	904
InputModelScene	904
InputModelView	904
InstanceSelectionEnabled	905
InstanceSelectionIndex	905
MagnitudeConstraint	906
MagnitudeThreshold	906
ModelTransformEnabled	906
Name	907
Polarity	907
PositionConstraint	908
ProcessingFormat	908
SamplingStep	909
SamplingStepCustom	909
SamplingStepCustomEnabled	910
SamplingStepDefault	910
ScoreThreshold	911
SortResultsEnabled	911
ToolHeight	911
ToolPositionX	912
ToolPositionY	912
ToolRotation	913
ToolSkew	914
ToolWidth	914
TransformRotation	914
TransformScale	915

TransformTranslationX	916
TransformTranslationY	916
TypeName	917
UnitsLength	917
UnitsOrientation	918
Version	918
Methods	918
AboutBox	918
ControlPropertyExchange	919
Execute	919
GetDebugParameter	919
GetDispatchPointer	919
GetProcessingFormatEnabled	919
HideProperties	920
LoadConfiguration	920
SaveConfiguration	920
SetDebugParameter	920
ShowProperties	920
Events	921
AskDatabase	921
DatabaseChange	922
Done	922
Error	922
PropertiesChange	922
SingleExecute	922
EvaluatorTool	923
Properties	923
Database	923
DatabasePassword	923
Description	923
ElapsedTime	923
ErrorObject	924
LeftInputConstant	924
LeftInputToolFunction	924
LeftInputToolFunctionIndex	925
LeftInputToolName	925
LeftInputType	925
LeftResult	926
Mode	926
Name	927
ProcessManager	927
Result	927
RightInputConstant	928
RightInputToolFunction	928
RightInputToolFunctionIndex	928
RightInputToolName	929
RightInputType	929
RightResult	929
TypeName	930
Version	930
Methods	930

AboutBox	930
ControlPropertyExchange	931
Execute	931
GetDebugParameter	931
GetDispatchPointer	931
HideProperties	931
LoadConfiguration	932
SaveConfiguration	932
SetDebugParameter	932
ShowProperties	932
Events	933
AskDatabase	933
DatabaseChange	933
Done	933
Error	933
PropertiesChange	933
SingleExecute	933
OCRFixedFont	934
Properties	934
BilinearInterpolationEnabled	934
CalibratedUnitsEnabled	934
CharacterConfidence	935
CharacterCornerX	935
CharacterCornerY	937
CharacterHeight	938
CharactersOnRow	939
CharacterSpacing	939
CharacterStrokeWidth	940
CharacterWidth	940
ConfidenceThreshold	941
CoordinateSystem	941
CustomFielding	942
Database	943
DatabasePassword	943
Description	944
ElapsedTime	944
ErrorObject	944
FineTuneMethodEnabled	945
FontType	946
InputImage	946
InputGreyScaleImage	947
InputView	947
InputGreyScaleView	947
InputInstanceScene	948
InputInstanceView	948
InputModelScene	948
InputModelView	949
InstanceSelectionEnabled	949
InstanceSelectionIndex	950
LearnElapsedTime	950
LearnFontType	950

LearnOcrMethod	951
LearnPrimaryCharacter	951
LearnPrimaryCharacterBoxBottom	952
LearnPrimaryCharacterBoxLeft	952
LearnPrimaryCharacterBoxRight	953
LearnPrimaryCharacterBoxTop	953
LearnStatusCode	953
LearnStatusCodeString	954
LearnTextOrientation	956
LearnTextPolarity	956
LearnTrainingBoxBottom	957
LearnTrainingBoxLeft	957
LearnTrainingBoxRight	958
LearnTrainingBoxTop	958
LearnTrainingString	958
ModelTransformEnabled	959
Name	959
OcrMethod	960
ProcessingAlgorithm	960
ProcessingSize	961
RowsToRead	962
SamplingStep	962
SamplingStepCustom	963
SamplingStepCustomEnabled	963
SamplingStepDefault	964
StatusCode	964
StatusCodeString	965
String	966
StringConfidence	966
StringLengthVariation	967
TextOrientation	967
TextPolarity	968
ToolHeight	969
ToolPositionX	969
ToolPositionY	970
ToolRotation	970
ToolWidth	971
TransformRotation	971
TransformScale	972
TransformTranslationX	972
TransformTranslationY	973
TypeName	973
UnitsLength	974
UnitsOrientation	974
Version	975
Methods	975
AboutBox	975
ControlPropertyExchange	975
Execute	975
GetDebugParameter	976
GetDispatchPointer	976
HideLearnInterface	976

HideProperties	976
Learn	977
LoadConfiguration	977
SaveConfiguration	977
SetDebugParameter	977
ShowLearnInterface	978
ShowProperties	978
Events	979
AskDatabase	979
DatabaseChange	979
Done	979
Error	979
PropertiesChange	980
SingleExecute	980
Image Histogram	981
Properties	981
BilinearInterpolationEnabled	981
CalibratedUnitsEnabled	981
Database	982
DatabasePassword	982
Description	982
ElapsedTime	982
ErrorObject	983
GreylevelRange	983
Histogram	983
HistogramPixelCount	984
ImageHeight	984
ImagePixelCount	985
ImageSubsampling	985
ImageWidth	986
InputImage	986
InputGreyScaleImage	986
InputView	987
InputGreyScaleView	987
InputInstanceScene	987
InputInstanceView	988
InputModelScene	988
InputModelView	988
InstanceSelectionEnabled	989
InstanceSelectionIndex	989
MaximumGreylevelValue	990
Mean	990
Median	990
MinimumGreylevelValue	991
Mode	991
ModelTransformEnabled	991
ModePixelCount	992
Name	992
SamplingStep	993
SamplingStepCustom	993
SamplingStepCustomEnabled	993

SamplingStepDefault	994
StandardDeviation	994
TailBlack	995
TailBlackGreylevelValue	995
TailWhite	996
TailWhiteGreylevelValue	996
ThresholdBlack	997
ThresholdWhite	997
ToolHeight	998
ToolPositionX	998
ToolPositionY	999
ToolRotation	999
ToolWidth	1000
TransformRotation	1001
TransformScale	1001
TransformTranslationX	1002
TransformTranslationY	1002
TypeName	1003
UnitsLength	1003
UnitsOrientation	1004
Variance	1004
Version	1004
Methods	1005
AboutBox	1005
ControlPropertyExchange	1005
Execute	1005
GetDebugParameter	1006
GetDispatchPointer	1006
HideProperties	1006
LoadConfiguration	1006
SaveConfiguration	1006
SetDebugParameter	1006
ShowProperties	1006
Events	1007
AskDatabase	1007
DatabaseChange	1008
Done	1008
Error	1008
PropertiesChange	1008
SingleExecute	1008
Image Processing Tool	1009
Properties	1009
ArithmeticClippingMode	1009
ArithmeticConstant	1009
ArithmeticScale	1010
AssignmentConstant	1010
AssignmentHeight	1010
AssignmentWidth	1011
Database	1011
DatabasePassword	1011
Description	1011

ElapsedTime	1012
ErrorObject	1012
FilteringClippingMode	1012
FilteringCustomKernelAnchorX	1013
FilteringCustomKernelAnchorY	1013
FilteringCustomKernelHeight	1014
FilteringCustomKernelValue	1014
FilteringCustomKernelWidth	1014
FilteringKernelSize	1015
FilteringScale	1015
HistogramThreshold	1016
InputImage	1016
InputView	1016
LogicalConstant	1017
MorphologicalNeighborhoodSize	1017
Name	1017
OperandImage	1018
OperandView	1018
Operation	1018
OutputImage	1020
OutputView	1020
OverrideType	1021
OverrideTypeEnabled	1021
ProcessingFormat	1022
ResultsDisplayMode	1022
TransformFlags	1023
TypeName	1023
Version	1024
Methods	1024
AboutBox	1024
ControlPropertyExchange	1024
Execute	1024
GetDebugParameter	1025
GetDispatchPointer	1025
HideProperties	1025
LoadConfiguration	1025
SaveConfiguration	1025
SetDebugParameter	1025
ShowProperties	1025
Events	1026
AskDatabase	1026
DatabaseChange	1026
Done	1027
Error	1027
PropertiesChange	1027
SingleExecute	1027
Image Sharpness	1028
Properties	1028
AutomaticCandidateCountEnabled	1028
BilinearInterpolationEnabled	1028
CalibratedUnitsEnabled	1029

CandidatePointsCount	1029
Database	1030
DatabasePassword	1030
Description	1030
ElapsedTime	1030
ErrorObject	1031
InputImage	1031
InputGreyScaleImage	1031
InputView	1032
InputGreyScaleView	1032
InputInstanceScene	1032
InputInstanceView	1033
InputModelScene	1033
InputModelView	1033
InstanceSelectionEnabled	1034
InstanceSelectionIndex	1034
KernelSize	1035
MeasurementPointsCount	1035
ModelTransformEnabled	1036
Name	1036
SamplingStep	1036
SamplingStepCustom	1037
SamplingStepCustomEnabled	1037
SamplingStepDefault	1038
Sharpness	1038
SharpnessHistory	1039
SharpnessHistoryScale	1039
SharpnessHistorySize	1040
SharpnessPeak	1040
StandardDeviationThreshold	1040
ToolHeight	1041
ToolPositionX	1041
ToolPositionY	1042
ToolRotation	1042
ToolWidth	1043
TransformRotation	1044
TransformScale	1044
TransformTranslationX	1045
TransformTranslationY	1045
TypeName	1046
UnitsLength	1046
UnitsOrientation	1047
Version	1047
Methods	1047
AboutBox	1047
ControlPropertyExchange	1048
Execute	1048
GetDebugParameter	1048
GetDispatchPointer	1048
HideProperties	1048
LoadConfiguration	1049
ResetHistory	1049

SaveConfiguration	1049
SetDebugParameter	1049
ShowProperties	1049
Events	1050
AskDatabase	1050
DatabaseChange	1050
Done	1050
Error	1050
PropertiesChange	1051
SingleExecute	1051
Line Finder	1052
Properties	1052
CalibratedUnitsEnabled	1052
ConformityTolerance	1052
ContrastThreshold	1053
ContrastThresholdMode	1053
CoordinateSystem	1054
Database	1055
DatabasePassword	1055
DefaultConformityTolerance	1055
Description	1056
ElapsedTime	1056
ErrorObject	1056
InputImage	1057
InputGreyScaleImage	1057
InputView	1057
InputGreyScaleView	1058
InputInstanceScene	1058
InputInstanceView	1058
InputModelScene	1059
InputModelView	1059
InstanceSelectionEnabled	1059
InstanceSelectionIndex	1060
LineAngle	1060
LineAverageContrast	1061
LineEndPointPositionX	1061
LineEndPointPositionY	1061
LineFitQuality	1062
LineFound	1062
LineMatchQuality	1063
LineStartPointPositionX	1063
LineStartPointPositionY	1064
LineVectorPointPositionX	1064
LineVectorPointPositionY	1065
MaximumAngleDeviation	1065
MaximumConformityTolerance	1065
MessageCount	1066
MessageDescription	1066
MessageNumber	1067
MinimumConformityTolerance	1067
MinimumLinePercentage	1068

ModelTransformEnabled	1068
Name	1069
OutputEntity	1069
OutputEntityEnabled	1069
OutputEntityView	1070
PolarityMode	1070
PositioningLevel	1071
ProcessingFormat	1071
SearchMode	1072
ShowEdges	1073
ShowEntityFound	1073
ShowEntityFoundAs	1073
ShowImage	1074
ShowGreyScaleImage	1074
ShowSearchArea	1074
SubsamplingLevel	1075
ToolGuidelineOffset	1075
ToolHeight	1075
ToolPositionX	1076
ToolPositionY	1076
ToolRotation	1077
ToolWidth	1078
TransformRotation	1078
TransformScale	1079
TransformTranslationX	1079
TransformTranslationY	1080
TypeName	1080
UnitsLength	1080
UnitsOrientation	1081
UseDefaultConformityTolerance	1081
Version	1082
Methods	1082
AboutBox	1082
ControlPropertyExchange	1083
Execute	1083
GetDebugParameter	1083
GetDispatchPointer	1083
GetProcessingFormatEnabled	1083
HideProperties	1084
LoadConfiguration	1084
SaveConfiguration	1084
SetDebugParameter	1084
ShowProperties	1084
Events	1085
AskDatabase	1085
DatabaseChange	1085
Done	1085
Error	1085
PropertiesChange	1086
SingleExecute	1086

Locator	1087
Properties	1087
AdvancedParametersVisible	1087
CalibratedUnitsEnabled	1087
ConformityTolerance	1088
ConformityToleranceRangeEnabled	1088
ContrastPolarity	1089
ContrastThreshold	1090
ContrastThresholdMode	1090
CoordinateSystem	1091
Database	1092
DatabasePassword	1092
DefaultConformityTolerance	1092
Description	1092
DetailLevel	1092
DynamicUpdate	1093
ElapsedTime	1093
ErrorObject	1094
FirstModelName	1094
InitialDirectory	1095
InputImage	1095
InputView	1095
InputInstanceScene	1096
InputInstanceView	1096
InputModelScene	1096
InputModelView	1097
InstanceClearQuality	1097
InstanceCount	1098
InstanceFitQuality	1098
InstanceMatchQuality	1098
InstanceModel	1099
InstanceModelName	1099
InstanceOrdering	1100
InstanceOrderingReferenceX	1101
InstanceOrderingReferenceY	1102
InstanceReferencePointCount	1102
InstanceReferencePointPositionX	1102
InstanceReferencePointPositionY	1103
InstanceRotation	1104
InstanceScaleFactor	1104
InstanceSelectionEnabled	1105
InstanceSelectionIndex	1105
InstanceSymmetry	1106
InstanceTime	1106
InstanceTranslationX	1107
InstanceTranslationY	1107
InstanceVisible	1108
LastModelName	1108
LearnTime	1109
MaskImageFileName	1109
MaximumConformityTolerance	1110
MaximumInstanceCount	1110

MaximumInstanceCountEnabled	1110
MaximumRotation	1111
MaximumScaleFactor	1111
MessageCount	1112
MessageDescription	1112
MessageNumber	1113
MinimumClearPercentage	1113
MinimumClearPercentageEnabled	1114
MinimumConformityTolerance	1114
MinimumModelPercentage	1114
MinimumRequiredFeatures	1115
MinimumRotation	1115
MinimumScaleFactor	1116
ModelAutomaticLevels	1116
ModelBasedMaximumRotation	1117
ModelBasedMaximumScaleFactor	1117
ModelBasedMinimumRotation	1118
ModelBasedMinimumScaleFactor	1118
ModelBasedRotationMode	1119
ModelBasedScaleFactorMode	1119
ModelBoundingAreaBottom	1120
ModelBoundingAreaLeft	1121
ModelBoundingAreaRight	1121
ModelBoundingAreaTop	1122
ModelContrastThreshold	1123
ModelContrastThresholdMode	1123
ModelCount	1124
ModelDatabase	1124
ModelDatabaseFileName	1125
ModelDatabaseModified	1125
ModelDetailLevel	1125
ModelDisambiguationEnabled	1126
ModelEditorActive	1126
ModelEditorEnabled	1127
ModelEnabled	1127
ModelFeatureSelection	1128
ModelName	1128
ModelOptimizerActive	1129
ModelOptimizerEnabled	1129
ModelOriginPositionX	1130
ModelOriginPositionY	1130
ModelOriginRotation	1131
ModelOutlineLevel	1132
ModelReferencePointCount	1132
ModelReferencePointPositionX	1133
ModelReferencePointPositionY	1133
ModelShadingAreaBottom	1134
ModelShadingAreaLeft	1135
ModelShadingAreaRight	1135
ModelShadingAreaTop	1136
ModelTrackingInertia	1136
ModelTransformEnabled	1137

Name	1137
NextModelName	1138
NominalRotation	1138
NominalRotationEnabled	1138
NominalScaleFactor	1139
NominalScaleFactorEnabled	1139
OutlineLevel	1140
OutputDetailScene	1140
OutputDetailSceneEnabled	1141
OutputDetailSceneView	1141
OutputInstanceScene	1141
OutputInstanceSceneEnabled	1142
OutputInstanceSceneView	1142
OutputMode	1143
OutputModelEnabled	1143
OutputModelView	1144
OutputOutlineScene	1144
OutputOutlineSceneEnabled	1145
OutputOutlineSceneView	1145
OutputSymmetricInstances	1146
ParametersBasedOn	1146
PercentageOfPointsToAnalyze	1147
PositioningLevel	1147
PreviousModelName	1148
ProcessingFormat	1148
RecognitionLevel	1149
RespectContrastPolarity	1149
SearchAreaBottom	1150
SearchAreaLeft	1150
SearchAreaRight	1150
SearchAreaTop	1150
SearchAreaUseEntireImage	1150
SearchBasedOnOutlineLevelOnly	1150
SearchTime	1151
ShowContourSceneDetail	1151
ShowContourSceneOutline	1152
ShowCoordinateSystem	1152
ShowImage	1152
ShowInstanceScene	1153
ShowInstanceSceneAs	1153
ShowReferencePoints	1154
ShowSearchArea	1154
Timeout	1155
TimeoutEnabled	1155
ToolHeight	1156
ToolPositionX	1156
ToolPositionY	1157
ToolRotation	1157
ToolUseEntireImage	1158
ToolWidth	1158
TrackingInertia	1159
TransformRotation	1159

TransformScale	1160
TransformTranslationX	1160
TransformTranslationY	1161
TypeName	1161
UnitsLength	1162
UnitsOrientation	1162
UseDefaultConformityTolerance	1163
UseExternalModelDatabase	1163
UseMinimumClearPercentage	1164
Version	1164
Methods	1165
AboutBox	1165
AddModel	1165
ClearMaskImage	1165
CompactMemory	1166
ControlPropertyExchange	1166
CopyModel	1166
Execute	1167
GetDebugParameter	1167
GetDispatchPointer	1167
GetInstanceIntrinsicBoundingBox	1168
GetProcessingFormatEnabled	1168
HideModelEditor	1169
HideModelOptimizer	1169
HideProperties	1169
HideResultsViewer	1170
ImportModel	1170
Learn	1171
LoadConfiguration	1171
LoadMaskImage	1172
LoadModelDatabase	1172
ModelOptimizerReset	1173
NewModelDatabase	1173
RemoveModel	1173
RenameModel	1174
SaveConfiguration	1174
SaveModelDatabase	1174
SetDebugParameter	1175
ShowModelEditor	1175
ShowModelEditorModal	1176
ShowModelOptimizer	1176
ShowModelOptimizerModal	1177
ShowProperties	1178
ShowResultsViewer	1179
Start	1180
Stop	1181
Unlearn	1182
Wait	1182
Events	1184
AskDatabase	1184
DatabaseChange	1184
Done	1184

Error	1184
PropertiesChange	1184
SingleExecute	1184
HSMModelEditorInterface	1185
Properties	1185
AutomaticLevels	1185
BoundingAreaBottom	1185
BoundingAreaLeft	1186
BoundingAreaRight	1186
BoundingAreaTop	1187
ContrastThreshold	1187
ContrastThresholdMode	1187
Database	1188
DetailLevel	1188
DisplayCalibrationMode	1188
FeatureSelection	1189
IconVisible	1190
ImageOriginY	1190
MessageCount	1190
MessageDescription	1190
MessageNumber	1191
ModelEdgePathCount	1192
ModelEdgePathLocation	1192
ModelEdgePathRequired	1193
ModelImage	1193
ModelMask	1194
ModelSceneDetail	1195
ModelSceneOutline	1195
Name	1195
OriginPositionX	1196
OriginPositionY	1196
OriginRotation	1197
OutlineLevel	1197
ReferencePointCount	1198
ReferencePointPositionX	1198
ReferencePointPositionY	1198
ShadingAreaBottom	1199
ShadingAreaLeft	1199
ShadingAreaRight	1200
ShadingAreaTop	1200
ShowBars	1200
ShowImage	1201
ShowImageMode	1201
TrackingInertia	1202
UseCustomShadingArea	1202
Methods	1202
AddModelEdgePath	1202
AddReferencePoint	1203
AnalyzeModel	1204
Apply	1204
BuildModel	1204

CreateMask	1205
CropModel	1206
EndDialog	1206
GetDispatchPointer	1207
Hide	1207
RefreshDisplay	1207
RemoveModelEdgePath	1208
RemoveReferencePoint	1208
RemoveMask	1209
Revert	1209
Show	1210
ShowModal	1210
Events	1211
AnalyzeModelClick	1211
ApplyClick	1211
BoundingAreaChange	1211
BuildModelClick	1211
CancelClick	1211
ContourDetectionParametersChange	1211
CoordinateSystemChange	1212
FeatureSelectionChange	1212
FeatureSelectionSliderChange	1212
OKClick	1212
ReferencePointsChange	1212
HSMModelOptimizerInterface	1213
Properties	1213
Database	1213
DisplayCalibrationMode	1213
IconVisible	1213
InstancesAnalyzed	1214
LinkOutlineWithDetail	1214
MinimumRecurrenceDetail	1215
MinimumRecurrenceOutline	1215
MorphingDetail	1216
MorphingOutline	1216
Name	1216
ShowBars	1217
ShowImage	1217
ShowImageMode	1217
Methods	1218
Apply	1218
EndDialog	1218
GetDispatchPointer	1219
Hide	1219
Reset	1219
Show	1219
ShowModal	1220
Update	1220
Events	1221
ApplyClick	1221
DetailParametersChange	1221

DoneClick	1221
OutlineParametersChange	1221
ResetClick	1221
ResultSelectionChange	1221
OutputTool	1222
Properties	1222
Database	1222
DatabasePassword	1222
Description	1222
ElapsedTime	1222
ErrorObject	1223
InputImage	1223
InputToolFunction	1223
InputToolFunctionIndex	1224
InputToolName	1224
InputView	1224
LastResult	1225
LastScheduledEncoder	1227
LastScheduledTime	1227
Name	1227
Offset	1228
OutputAssertOn	1228
OutputIndex	1228
OutputIP	1228
OutputUnits	1229
ProcessManager	1229
Result	1229
ScheduledEncoder	1231
ScheduledTime	1231
TypeName	1232
Version	1232
Methods	1232
AboutBox	1232
ControlPropertyExchange	1233
Execute	1233
GetDebugParameter	1233
GetDispatchPointer	1233
HideProperties	1233
LoadConfiguration	1234
SaveConfiguration	1234
SetDebugParameter	1234
ShowProperties	1234
Events	1235
AskDatabase	1235
DatabaseChange	1235
Done	1235
Error	1235
PropertiesChange	1236
SingleExecute	1236

Pattern Locator	1237
Properties	1237
AutoCoarsenessSelectionEnabled	1237
BilinearInterpolationEnabled	1237
CalibratedUnitsEnabled	1238
CoordinateSystem	1238
Database	1239
DatabasePassword	1239
Description	1239
ElapsedTime	1239
ErrorObject	1240
InputImage	1240
InputGreyScaleImage	1241
InputView	1241
InputGreyScaleView	1241
InputInstanceScene	1242
InputInstanceView	1242
InputModelScene	1242
InputModelView	1243
InstanceSelectionEnabled	1243
InstanceSelectionIndex	1243
MatchCount	1244
MatchPositionX	1244
MatchPositionY	1245
MatchRotation	1245
MatchStrength	1246
MatchThreshold	1246
MaximumInstanceCount	1247
MaximumInstanceCountEnabled	1247
ModelTransformEnabled	1247
Name	1248
OverrideModelImageEnabled	1248
PatternDatabaseFileName	1249
PatternHeight	1249
PatternPositionX	1250
PatternPositionY	1250
PatternRotation	1251
PatternWidth	1252
PositioningCoarseness	1252
SamplingStep	1252
SamplingStepCustom	1253
SamplingStepCustomEnabled	1253
SamplingStepDefault	1254
SearchCoarseness	1254
ToolHeight	1255
ToolPositionX	1255
ToolPositionY	1256
ToolRotation	1256
ToolWidth	1257
TransformRotation	1257
TransformScale	1258
TransformTranslationX	1258

TransformTranslationY	1259
TypeName	1259
UnitsLength	1260
UnitsOrientation	1260
Version	1261
Methods	1261
AboutBox	1261
ControlPropertyExchange	1261
CreatePatternFromInput	1262
Execute	1262
GetDebugParameter	1262
GetDispatchPointer	1262
HideProperties	1262
ImportPatternFromDatabase	1263
LoadConfiguration	1263
LoadPatternDatabase	1263
NewPatternDatabase	1264
SaveConfiguration	1264
SavePatternDatabase	1264
SetDebugParameter	1264
ShowProperties	1265
Events	1266
AskDatabase	1266
DatabaseChange	1266
Done	1266
Error	1266
PropertiesChange	1266
SingleExecute	1266
Point Finder	1267
Properties	1267
CalibratedUnitsEnabled	1267
Connectivity	1267
ConnectivityEnabled	1268
ContrastThreshold	1268
ContrastThresholdMode	1269
CoordinateSystem	1269
Database	1270
DatabasePassword	1270
Description	1270
ElapsedTime	1271
ErrorObject	1271
InputImage	1271
InputView	1272
InputInstanceScene	1272
InputInstanceView	1273
InputModelScene	1273
InputModelView	1273
InstanceSelectionEnabled	1274
InstanceSelectionIndex	1274
InterpolatePositionMode	1274
InterpolatePositionModeEnabled	1275

MessageCount	1275
MessageDescription	1276
MessageNumber	1276
ModelTransformEnabled	1277
Name	1277
OutputEntity	1278
OutputEntityEnabled	1278
OutputEntityView	1279
PointAverageContrast	1279
PointFound	1279
PointPositionX	1280
PointPositionY	1280
PolarityMode	1280
PositioningLevel	1281
ProcessingFormat	1281
SearchMode	1282
ShowEdges	1283
ShowEntityFound	1283
ShowImage	1283
ShowSearchArea	1284
SubsamplingLevel	1284
ToolGuidelineOffset	1284
ToolHeight	1285
ToolPositionX	1285
ToolPositionY	1286
ToolRotation	1286
ToolWidth	1287
TransformRotation	1287
TransformScale	1288
TransformTranslationX	1288
TransformTranslationY	1289
TypeName	1289
UnitsLength	1290
UnitsOrientation	1290
Version	1291
Methods	1291
AboutBox	1291
ControlPropertyExchange	1291
Execute	1292
GetDebugParameter	1292
GetDispatchPointer	1292
GetProcessingFormatEnabled	1292
HideProperties	1292
LoadConfiguration	1293
SaveConfiguration	1293
SetDebugParameter	1293
ShowProperties	1293
Events	1294
AskDatabase	1294
DatabaseChange	1294
Done	1294
Error	1295

PropertiesChange	1295
SingleExecute	1295
Process Manager	1296
Properties	1296
Database	1296
Description	1296
ElapsedTime	1296
ErrorObject	1297
FirstProcessName	1297
InteractiveConfigurationEnabled	1297
InterfaceVisible	1298
LastProcessName	1298
Mode	1298
Name	1299
NextProcessName	1299
PreviousProcessName	1300
Process	1300
ProcessAvailable	1301
ProcessCount	1301
ProcessIndex	1301
ProcessListEditable	1302
ProcessTypeName	1302
Version	1303
Methods	1303
AboutBox	1303
AddProcess	1304
ControlPropertyExchange	1304
DuplicateProcess	1304
Execute	1305
ExportConfiguration	1305
GetDebugParameter	1306
GetDispatchPointer	1306
HideInterface	1306
HideProperties	1306
ImportConfiguration	1307
InquireConfiguration	1308
LoadConfiguration	1308
MoveProcess	1309
Refresh	1309
RemoveProcess	1310
Reset	1310
SaveConfiguration	1311
SetDebugParameter	1311
ShowInterface	1311
ShowProperties	1312
Events	1312
AskDatabase	1312
DatabaseChange	1312
Done	1312
Error	1313
ModeChange	1313

PropertiesChange	1313
Sampling Tool	1314
Properties	1314
BilinearInterpolationEnabled	1314
CalibratedImageHeight	1314
CalibratedImageWidth	1315
CalibratedUnitsEnabled	1315
CoordinateSystem	1315
Database	1316
DatabasePassword	1316
Description	1316
ElapsedTime	1317
ErrorObject	1317
ImageBottomLeftX	1317
ImageBottomLeftY	1318
ImageBottomRightX	1319
ImageBottomRightY	1319
ImageHeight	1320
ImageTopLeftX	1320
ImageTopLeftY	1320
ImageTopRightX	1321
ImageTopRightY	1322
ImageWidth	1322
InputImage	1322
InputView	1323
InputInstanceScene	1323
InputInstanceView	1323
InputModelScene	1324
InputModelView	1324
InstanceSelectionEnabled	1324
InstanceSelectionIndex	1325
ModelTransformEnabled	1325
Name	1326
OutputImage	1326
OutputView	1326
PixelHeight	1327
PixelWidth	1327
ProcessingFormat	1328
SamplingStep	1329
SamplingStepCustom	1329
SamplingStepCustomEnabled	1330
SamplingStepDefault	1330
ToolHeight	1330
ToolPositionX	1331
ToolPositionY	1331
ToolRotation	1331
ToolWidth	1332
TransformRotation	1333
TransformScale	1333
TransformTranslationX	1333
TransformTranslationY	1334
TypeName	1334

UnitsLength	1335
UnitsOrientation	1335
Version	1336
Methods	1336
AboutBox	1336
ControlPropertyExchange	1336
Execute	1336
GetDebugParameter	1337
GetDispatchPointer	1337
HideProperties	1337
LoadConfiguration	1337
SaveConfiguration	1337
SetDebugParameter	1337
ShowProperties	1338
Events	1338
DatabaseChange	1338
Done	1339
Error	1339
PropertiesChange	1339
SingleExecute	1339
Index	1340

Acquisition Device

Description

The Acquisition Device acquires Images from a device or from a database of Images. Integrated image calibration methods can compensate for pixel aspect ratio, perspective and lens distortion.

Properties

Calibration

Description

Returns a copy of the internal **HSCalibration** object of the specified control configuration. Used to access calibration information. Read only.

Syntax

control.**Calibration**([*Configuration* As Variant])

Type

HSCalibration

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Calibration As HSCalibration
Calibration = AcquisitionDevice.Calibration(1)
```

CalibrationConfigurationName

Description

Name of the configuration that contains the calibration when the **From Another Configuration** calibration mode is selected.

Syntax

control.**CalibrationConfigurationName**([*Configuration* As Variant]) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property is used only if the **CalibrationMethod** property of the selected configuration is set to `hsOtherConfiguration`. This property applies to the default configuration if the Configuration parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ConfName As String
CfgName = AcqDevice1.CalibrationConfigurationName("Config0")
AcqDevice1.CalibrationConfigurationName("Config0") = "Config1"
```

CalibrationDescription

Description

Text description of the selected calibration method. Read only.

Syntax

```
control.CalibrationDescription(Method As hsCalibrationMethod)
```

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Method</i>	Selected calibration method.	hsCalibrationMethod	Not applicable.

Example

```
' Description of the default configuration's calibration
Dim Method As hsCalibrationMethod
Method = AcqDevice1.CalibrationMethod
Dim Description As String
Description = AcqDevice1.CalibrationDescription(Method)
```

CalibrationMethod

Description

Calibration method used by the selected configuration. Can be set to `hsAuto` to directly create a new calibration programmatically without using the **HSCalibrationDistortionInterface**, the **HSCalibrationPerspectiveInterface** or the **HSCalibrationXYScaleInterface** objects.

When a new **hsAuto** calibration is created, the actual calibration method that is used depends on the number of pairs of World-Image coordinates that are provided upon its creation. With 2 to 3 pairs of World-Image coordinates, X and Y scale factors calibration is used. With 4 to 8 pairs of World-Image coordinates, perspective correction calibration is used. With 9 or more pairs of World-Image coordinates, distortion correction calibration is used. The parameters of a **hsAuto** calibration, including the pairs of World-Image coordinates, are set through the indexed **CalibrationParameter** property.

Syntax

control.**CalibrationMethod**([*Configuration* As Variant]) [= *Value* As *hsCalibrationMethod*]

Type

hsCalibrationMethod

Range

The valid range for this property is as follows:

hsCalibrationMethod	Value	Description
hsOriginal	0	The original calibration of the image is used (useful for file emulation and selection modes).
hsOtherConfiguration	1	Calibration borrowed from another configuration.
hsXYScale	2	X and Y scale calibration factors.
hsDistortionLUT	3	Lens and perspective distortion correction by lookup table.
hsDistortionModel	4	Lens and perspective distortion correction using a non-linear model.
hsPerspective	6	Perspective correction.
hsAuto	7	Calibration method automatically selected according to the number of calibration points provided. Available only programmatically.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Method As hsCalibrationMethod
Method = AcqDevice1.CalibrationMethod("Configuration0")
AcqDevice2.CalibrationMethod("Configuration0") = hsOriginal
```

CalibrationMethodName

Description

Name of a selected calibration method. Read only.

Syntax

control.**CalibrationName**(*Method* As hsCalibrationMethod)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Method</i>	Selected calibration method.	hsCalibrationMethod	Not applicable.

Example

```
Dim Method As hsCalibrationMethod
Method = AcqDevice1.CalibrationMethod
Dim Name As String
Name = AcqDevice1.CalibrationMethodName(Method)
```

CalibrationParameter

Description

Value of the indexed parameter of the current calibration method for a given configuration.

Syntax

control.**CalibrationParameter**(*Index* As Integer , [*Configuration* As Variant]) [= *Value* As Single]

Type

Single

Range

Depends on the selected parameter and the current calibration method.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: CalibrationParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. The calibration parameters should be set programmatically only with the **hsAuto** method. They should be considered read only otherwise and the calibration dialogs should be used to define the calibration parameters. To load and save a set of calibration parameters, the **LoadCalibration** and **SaveCalibration** methods can be used. When a new calibration is created programmatically using the **hsAuto** method (see **CalibrationMethod**), the indexed

CalibrationParameter property is used to set its parameters. A description of each parameter index is provided in the table below.

Index	Description
0	Number of pairs of calibration points used for the calibration
1	World coordinate system type. A value of 0 defines a right-handed coordinate system and value of 1 defines a left-handed coordinate system
2	Reserved for future use
3	Reserved for future use
4	Reserved for future use
5	Reserved for future use
6	Width of the image in pixels
7	Height of the image in pixels
8	Read only: average width of an image pixel
9	Read only: average height of an image pixel
10	X coordinate of point 1 expressed with respect to the Image coordinate system
11	Y coordinate of point 1 expressed with respect to the Image coordinate system
12	X coordinate of point 1 expressed with respect to the World coordinate system
13	Y coordinate of point 1 expressed with respect to the World coordinate system
14	X coordinate of point 2 expressed with respect to the Image coordinate system
15	Y coordinate of point 2 expressed with respect to the Image coordinate system
16	X coordinate of point 2 expressed with respect to the World coordinate system
17	Y coordinate of point 2 expressed with respect to the World coordinate system
...	...
$10+4(n-1)$	X coordinate of point n expressed with respect to the Image coordinate system
$10+4(n-1)+1$	Y coordinate of point n expressed with respect to the Image coordinate system
$10+4(n-1)+2$	X coordinate of point n expressed with respect to the World coordinate system
$10+4(n-1)+3$	Y coordinate of point n expressed with respect to the World coordinate system

Example

```
' Create a new hsAuto calibration in the current
' Acquisition Device configuration
```

```

AcquDev.CalibrationMethod = hsAuto
' Number of pairs of calibration points
AcquDev.CalibrationParameter(0) = 4
' Calibrate into a left-handed coordinate system
AcquDev.CalibrationParameter(1) = 1
' Calibration point 1
AcquDev.CalibrationParameter(10) = 100.0
AcquDev.CalibrationParameter(11) = 50.0
AcquDev.CalibrationParameter(12) = -1500.0
AcquDev.CalibrationParameter(13) = 2300.0
' Calibration point 2
AcquDev.CalibrationParameter(14) = 500.0
AcquDev.CalibrationParameter(15) = 50.0
AcquDev.CalibrationParameter(16) = 1500.0
AcquDev.CalibrationParameter(17) = 2300.0
' Calibration point 3
AcquDev.CalibrationParameter(18) = 100.0
AcquDev.CalibrationParameter(19) = 400.0
AcquDev.CalibrationParameter(20) = -1500.0
AcquDev.CalibrationParameter(21) = 6100.0
' Calibration Point 4
AcquDev.CalibrationParameter(22) = 500.0
AcquDev.CalibrationParameter(23) = 400.0
AcquDev.CalibrationParameter(24) = 1500.0
AcquDev.CalibrationParameter(25) = 6100.0
' Execute the Acquisition Device process in order to
' create the calibration internally
AcquDev.Execute
' Validate the newly created calibration
Dim CalibrationOk As Boolean
CalibrationOk = AcquDev.ValidateCalibration

```

CalibrationParameterCount

Description

Number of parameters of the current calibration method for a given configuration.

Syntax

control.**CalibrationParameterCount**([*Configuration* As Variant])

Type

Integer

Range

Greater than 0.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

CalibrationParameterCount is writable only with the *hsDistortionLUT* and *hsAuto* calibration methods. It is read only otherwise.

Example

```
' Dynamic copy of a DistortionLUT calibration between
' two configurations of the same Acquisition Device
If AcqDevice1.CalibrationMethod(0) = hsDistortionLUT Then
    Dim ParamCount As Integer
    ParamCount = AcqDevice1.CalibrationParameterCount(0)
    AcqDevice1.CalibrationMethod(1) = hsDistortionLUT
    AcqDevice1.CalibrationParameterCount(1) = ParamCount
    Dim i As Integer
    For i = 0 To ParamCount - 1
        AcqDevice1.CalibrationParameter(i, 1) = _
            AcqDevice2.CalibrationParameter(i, 0)
    Next i
End If
```

CalibrationParameterDescription**Description**

Text description of the indexed parameter of the current calibration method for a given configuration. Read only.

Syntax

control.**CalibrationParameterDescription**(*Index* As Integer , [*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: CalibrationParameterCount - 1

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Description As String
Description = AcqDevice1.CalibrationParameterDescription _
    (2, "Configuration0")
```

CalibrationParameterEditable

Description

Editable status of the indexed parameter of the current calibration method for the given configuration. An editable parameter can be modified directly by the user in the properties interface. Read only.

Syntax

control.**CalibrationParameterEditable**(*Index* As Integer , [*Configuration* As Variant])

Type

Boolean

Range

True, false.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: CalibrationParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the selected parameter is not editable, it still can be modified programmatically, even though it cannot be modified in the user interface.

Example

```
Dim ParamEditable As Boolean
ParamEditable = AcqDevice1.CalibrationParameterEditable _
    (3, "Configuration0")
```


CalibrationParameterName

Description

Name of the indexed parameter of the current calibration method for the given configuration.
Read only.

Syntax

control.**CalibrationParameterName**(*Index* As Integer , [*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: CalibrationParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Name As String
Name = AcqDevice1.CalibrationParameterName (4, "Configuration0")
```

ColorCalibration

Description

Returns a copy of the internal **HSCalibrationColor** object of the specified control configuration. Used to access color calibration information. Read only.

Syntax

control.**ColorCalibration**([*Configuration* As Variant])

Type

HSCalibrationColor

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Calibration As HSCalibrationColor
Calibration = AcquisitionDevice.ColorCalibration(1)
```

ColorCalibConfigurationName

Description

Name of the configuration that contains the color calibration when the **From Another Configuration** color calibration mode is selected.

Syntax

control.**ColorCalibConfigurationName**([*Configuration* As Variant]) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property is used only if the **ColorCalibMethod** property of the selected configuration is set to *hsOtherConfiguration*. This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ConfName As String
CfgName = AcqDevice1.ColorCalibConfigurationName("Config0")
AcqDevice1.ColorCalibConfigurationName("Config0") = "Config1"
```

ColorCalibDescription

Description

Text description of the selected color calibration method. Read only.

Syntax

control.**ColorCalibDescription**(*Method* As *hsColorCalibrationMethod*)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Method</i>	Selected calibration method.	hsColorCalibrationMethod	Not applicable.

Example

```
' Description of the default configuration's color calibration
Dim Method As hsColorCalibrationMethod
Method = AcqDevice1.ColorCalibMethod
Dim Description As String
Description = AcqDevice1.ColorCalibDescription(Method)
```

ColorCalibMethod**Description**

Color calibration method used by the selected configuration.

Syntax

control.**ColorCalibMethod**([*Configuration* As Variant]) [= *Value* As hsColorCalibrationMethod]

Type

hsColorCalibrationMethod

Range

The valid range for this property is as follows:

hsCalibrationMethod	Value	Description
hsOriginal	0	The original calibration of the image is used (useful for file emulation and selection modes).
hsOtherConfiguration	1	Calibration borrowed from another configuration.
hsFilterCharacteristicsModel	2	Filter characteristics based color calibration model.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Method As hsColorCalibrationMethod
Method = AcqDevice1.ColorCalibMethod("Configuration0")
AcqDevice2.ColorCalibMethod("Configuration0") = hsOriginal
```

ColorCalibMethodName

Description

Name of a selected color calibration method. Read only.

Syntax

control.**ColorCalibName**(*Method* As hsColorCalibrationMethod)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Method</i>	Selected calibration method.	hsColorCalibrationMethod	Not applicable.

Example

```
Dim Method As hsColorCalibrationMethod
Method = AcqDevice1.ColorCalibMethod
Dim Name As String
Name = AcqDevice1.ColorCalibMethodName(Method)
```

ColorCalibParameter

Description

Value of the indexed parameter of the current color calibration method for a given configuration.

Syntax

control.**ColorCalibParameter**(*Index* As Integer , [*Configuration* As Variant]) [= *Value* As Single]

Type

Single

Range

Depends on the selected parameter and the current calibration method.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ColorCalibParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. The calibration parameters should be considered read only and the color calibration dialog should be used to

define the calibration parameters. To load and save a set of color calibration parameters, the **LoadColorCalibration** and **SaveColorCalibration** methods can be used.

ColorCalibParameterCount

Description

Number of parameters of the current color calibration method for a given configuration.

Syntax

control.**ColorCalibParameterCount**([*Configuration* As Variant])

Type

Integer

Range

Greater than 0.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

ColorCalibParameterCount should be considered read only.

ColorCalibParameterDescription

Description

Text description of the indexed parameter of the current color calibration method for a given configuration. Read only.

Syntax

control.**ColorCalibParameterDescription**(*Index* As Integer , [*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ColorCalibParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Description As String
Description = AcqDevice1.ColorCalibParameterDescription _
    (2, "Configuration0")
```

ColorCalibParameterEditable

Description

Editable status of the indexed parameter of the current color calibration method for the given configuration. An editable parameter can be modified directly by the user in the properties interface. Read only.

Syntax

control.**ColorCalibParameterEditable**(*Index* As Integer , [*Configuration* As Variant])

Type

Boolean

Range

True, false.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ColorCalibParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the selected parameter is not editable, it still can be modified programmatically, even though it cannot be modified in the user interface.

Example

```
Dim ParamEditable As Boolean
ParamEditable = AcqDevice1.ColorCalibParameterEditable _
    (3, "Configuration0")
```

ColorCalibParameterName

Description

Name of the indexed parameter of the current color calibration method for the given configuration. Read only.

Syntax

control.**ColorCalibParameterName**(*Index* As Integer , [*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ColorCalibParameterCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim Name As String
Name = AcqDevice1.ColorCalibParameterName (4, "Configuration0")
```

ConfigurationCount

Description

Number of configurations defined in the Acquisition Device tool. Read only.

Syntax

control.**ConfigurationCount**

Type

Integer

Range

Any positive Integer.

Parameter(s)

Not applicable.

Remark(s)

Even though **ConfigurationCount** is read only, it is possible to add or remove configurations from the tool by programming the **CreateConfiguration** and **DeleteConfiguration** methods.

Example

```
Dim NConfig As Integer
NConfig = AcqDevice1.ConfigurationCount
```

ConfigurationDefault

Description

Default configuration of the tool.

Syntax

control.**ConfigurationDefault**

Type

Variant

Range

Minimum: 0

Maximum: **ConfigurationCount** - 1**Parameter(s)**

Not applicable.

Remark(s)

Since most properties and methods have an optional configuration parameter that is interpreted as **ConfigurationDefault** when omitted, this parameter can be used to simplify the code when performing a series of operations on a specific configuration or when working with an Acquisition Device that has just a single configuration.

Example

```
Dim Default As Integer
Default = AcqDevice1.ConfigurationDefault
' Set Configuration0 As default and execute
AcqDevice1.ConfigurationDefault = "Configuration0"
AcqDevice1.Execute
' Set Configuration1 As default and execute
AcqDevice1.ConfigurationDefault = "Configuraiton1"
AcqDevice1.Execute
```

ConfigurationEnabled

Description

Setting **ConfigurationEnabled** to True activates the selected configuration. Activating multiple configurations permits a single **Execute** command to perform an acquisition on each active configuration.

Syntax

control.**ConfigurationEnabled**(*Configuration as Variant*) [= *Value As Boolean*]

Type

Boolean

Range

True, False

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Example

```
' Activate Configuration1 and Configuration2
AcqDevice1.ConfigurationEnabled("Configuration1") = True
AcqDevice1.ConfigurationEnabled("Configuration2") = True
AcqDevice1.Execute
```


ConfigurationIndex

Description

Property used to get the index of a configuration from its name.

Syntax

control.**ConfigurationIndex**(*Name* As String)

Type

Integer

Range

Minimum: 0

Maximum: **ConfigurationCount** - 1

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the configuration to convert to an index.	String	Not applicable.

Example

```
Dim Index As Integer
```

```
Index = AcqDevice1.ConfigurationIndex("Configuration0")
```

ConfigurationName

Description

Name of a given configuration.

Syntax

control.**ConfigurationName**([*Configuration* As Variant]) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ConfigName As String
```

```
ConfigName = AcqDevice1.ConfigurationName("Configuration0")
```

```
AcqDevice2.ConfigurationName(0) = "Configuration0"
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

Description of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim Description As String
Description = AcqDevice1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the Acquisition Device tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Parameter(s)

Not applicable.

Example

```
Dim Time As Single
Time = AcqDevice1.ElapsedTime
```

EmulationDatabaseFileName

Description

File name of the current emulation database for a given configuration. Read only.

Syntax

control.**EmulationDatabaseFileName**([*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. Even though the **EmulationDatabaseFileName** parameter is read only, it is possible to switch to another emulation database by using the **LoadEmulationDatabase** method.

Example

```
AcqDevice1.LoadEmulationDatabase "Acquisitions.hdb"
Dim FileName As String
' If the LoadEmulationDatabase succeeded, FileName
' should be "Acquisitions.hdb"
FileName = _
    AcqDevice1.EmulationDatabaseFileName("Configuration0")
```

EmulationImageCount

Description

Number of images in the emulation database of a given configuration. Read only.

Syntax

control.**EmulationImageCount**([*Configuration* As Variant])

Type

Integer

Range

Greater or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the selected configuration does not have an emulation database loaded, this parameter returns 0.

Example

```
Dim NImage As Integer
NImage = AcqDevice1.EmulationImageCount("Configuration0")
```

EmulationImageEnabled

Description

Enables the status of an image from an emulation database for a given configuration. Disabled images are left in the emulation database but are ignored when the tool is executed in emulation mode. The **EmulationEnableAll** and **EmulationDisableAll** methods can be used to change the state of all the emulation images of a configuration.

Syntax

control.**EmulationImageEnabled**(*Index* As Integer , [*Configuration* As Variant]) [= *Value* As Boolean]

Type

Boolean

Range

True, False

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the image in the emulation database.	Integer	Minimum: 0 Maximum: EmulationImageCount - 1
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim EmulEnabled As Boolean
EmulEnabled = AcqDevice1.EmulationImageEnabled
    (2, "Configuration0")
AcqDevice2.EmulationImageEnabled(1, "Configuration0") = False
```

EmulationImageName

Description

Name of a specified image index in an emulation database for a given configuration. Read only.

Syntax

control.**EmulationImageName**(*Index* As Integer [*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the image in the emulation database.	Integer	Minimum: 0 Maximum: EmulationImageCount - 1
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. Even though this property is read only, it is possible to change the name of an emulation image manually from the property pages of the Acquisition Device.

Example

```
' Fills a combo box with the name of the images of an
' emulation database from the default configuration
ComboBox.Clear
Dim i As Integer
For i = 0 To AcqDevice1.EmulationImageCount - 1
    Dim Name As String
    Name = AcqDevice1.EmulationImageName(i)
    ComboBox.AddItem Name
Next i
```

EmulationNextImageIndex

Description

Index of the image that will be output the next time the selected configuration of the Acquisition Device is executed and the acquisition is set to emulation database.

Syntax

control.**EmulationNextImageIndex**([*Configuration* As Variant])

Type

Integer

Range

Minimum: 0

Maximum: **EmulationImageCount** - 1

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```

' Two configurations with synchronized emulation
' databases (ex:stereo left-right), making sure the
' corresponding images are output regardless of the
' enable/disable state of images in the second
' configuration
Dim Running As Boolean
Running = True
Do
    ' Store index of image to be output
    Dim Index As Integer
    Index = AcqDevice1.EmulationNextImageIndex("Config0")
    ' Execute 1st configuration
    AcqDevice1.ConfigurationDefault = "Config0"
    AcqDevice1.Execute
    ' Make sure image is enabled in 2nd configuration and
    ' set it As next emulation image
    AcqDevice1.EmulationImageEnabled(Index, "Config1") = True
    AcqDevice1.EmulationNextImageIndex("Config1") = Index
    ' Execute 2nd configuration
    AcqDevice1.ConfigurationDefault = "Config1"
    AcqDevice1.Execute
    DoEvents
Loop While Running

```

EmulationNextImageName

Description

Name of the image that will be output the next time the selected configuration of the Acquisition Device will be executed if the acquisition is set to emulation database. Read only.

Syntax

control.**EmulationNextImageName**([*Configuration* As Variant])

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
' Shows the name of the next emulation image to be output
  TextBox.Text = AcqDevice1.EmulationNextImageName
  AcqDevice1.Execute
```

ErrorCode

Description

Reserved for internal use.

ErrorDescription

Description

Reserved for internal use.

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

GrabberDCAMParameter

Description

Indexed property used to specify the parameters of a DCAM IEEE1394 device. If the specified device is not a generic DCAM device, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDCAMParameter**(*Index* As hsDCAMParameterIndex , *Option* As Long , [*Device* As Variant] , [*Configuration* As Variant]) [= *NewValue* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* and *Option* parameters. See tables below:

Index Parameter	Option Parameter	Description	Range and Type
hsDCAMFeatureBrightness hsDCAMFeatureAutoExposure hsDCAMFeatureSharpness hsDCAMFeatureWhiteBalance hsDCAMFeatureHue hsDCAMFeatureSaturation hsDCAMFeatureGamma hsDCAMFeatureShutter hsDCAMFeatureGain hsDCAMFeatureIris hsDCAMFeatureFocus hsDCAMFeatureTemperature hsDCAMFeatureTriggerDelay hsDCAMFeatureWhiteShading hsDCAMFeatureFrameRate hsDCAMFeatureZoom hsDCAMFeaturePan hsDCAMFeatureTilt hsDCAMFeatureOpticalFilter hsDCAMFeatureCaptureSize hsDCAMFeatureCaptureQuality	hsDCAMName hsDCAMUnits hsDCAMID hsDCAMPresence hsDCAMABSControl hsDCAMOnePush hsDCAMReadout hsDCAMONOFF hsDCAMAutoMode hsDCAMManualMode hsDCAMRangeMin hsDCAMRangeMax hsDCAMRangeABSMIn hsDCAMRangeABSMax hsDCAMValueLO hsDCAMValueHI hsDCAMValueABS	Standard DCAM Features.	Depends on option specified. String for Name and Units. Unsigned Long for ID, RangeMin, RangeMax, ValueHI, ValueLo. Float for RangeABSMIn, RangeABSMax, ValueABS. Boolean for Presence, ABSControl, OnePush, Readout, OnOff, AutoMode, ManualMode. Some options are read-only: Name, Units, ID, RangeMin, RangeMax, RangeABSMIn, RangeABSMax.
hsDCAMStrobe hsDCAMStrobeOffset	hsDCAMNone	Strobes	hsDCAMStrobeOffset returns offset register. Read-Only.
hsDCAMFeatureStrobe0 hsDCAMFeatureStrobe1 hsDCAMFeatureStrobe2 hsDCAMFeatureStrobe3	hsDCAMName hsDCAMUnits hsDCAMID hsDCAMPresence hsDCAMABSControl hsDCAMOnePush hsDCAMReadout hsDCAMONOFF hsDCAMAutoMode hsDCAMManualMode hsDCAMPolarity hsDCAMRangeMin hsDCAMRangeMax hsDCAMRangeABSMIn hsDCAMRangeABSMax hsDCAMValueLO hsDCAMValueHI hsDCAMValueABS	Quick access to the 4 DCAM strobes.	Depends on option specified. String for Name and Units Unsigned Long for ID, RangeMin, RangeMax, ValueHI, ValueLo Float for RangeABSMIn, RangeABSMax, ValueABS Boolean for Presence, ABSControl, OnePush, Readout, OnOff, AutoMode, ManualMode, Polarity Some options are read-only: Name, Units, ID, RangeMin, RangeMax, RangeABSMIn, RangeABSMax.

Index Parameter	Option Parameter	Description	Range and Type
hsDCAMFeatureTriggerMode	hsDCAMName hsDCAMUnits hsDCAMID hsDCAMPresence hsDCAMABSControl hsDCAMOnePush hsDCAMReadout hsDCAMONOFF hsDCAMAutoMode hsDCAMManualMode hsDCAMPolarity hsDCAMValueReadout hsDCAMRangeMin hsDCAMRangeMax hsDCAMRangeABSMIn hsDCAMRangeABSMax hsDCAMValueLO hsDCAMValueHI hsDCAMValueABS hsDCAMTriggerMode hsDCAMTriggerModeParameter hsDCAMTriggerMode0 hsDCAMTriggerMode1 hsDCAMTriggerMode2 hsDCAMTriggerMode3 hsDCAMTriggerMode4 hsDCAMTriggerMode5 hsDCAMTriggerModeVendor0 hsDCAMTriggerModeVendor1 hsDCAMTriggerSource hsDCAMTriggerSource0 hsDCAMTriggerSource1 hsDCAMTriggerSource2 hsDCAMTriggerSource3 hsDCAMTriggerSourceSoftware hsDCAMTriggerDoSoftware	Quick access to the DCAM Triggers	Depends on option specified. String for Name and Units Unsigned Long for ID, RangeMin, RangeMax, ValueHI, ValueLO Float for RangeABSMIn, RangeABSMax, ValueABS Boolean for Presence, ABSControl, OnePush, Readout, OnOff, AutoMode, ManualMode, Polarity, ValueReadout Some options are read-only: Name, Units, ID, RangeMin, RangeMax, RangeABSMIn, RangeABSMax. hsDCAMTriggerMode: unsigned short: 0-15 hsDCAMTriggerModeParameter: unsigned short hsDCAMTriggerSource: unsigned short: 0-7 hsDCAMTriggerModeX for direct Mode set hsDCAMTriggerSourceX for direct Source set
hsDCAMVersion hsDCAMCameraName hsDCAMCameraVendor hsDCAMCameraUID hsDCAMMaxSpeed	hsDCAMNone	Standard DCAM informations.	Depend on information requested. Read-Only. Version: unsigned long CameraName: String CameraVendor: String CameraUID: longlong (8 bytes) MaxSpeed: unsigned long
hsDCAMPowerControl hsDCAMIEEE1394B	hsDCAMNone	Standard DCAM settings.	Boolean

Index Parameter	Option Parameter	Description	Range and Type
hsDCAMMEMNumChannels hsDCAMMEMCurrentChannel hsDCAMMEMLoadChannel hsDCAMMEMSaveChannel	hsDCAMNone	Memory channels functions.	Channel number as Integer NumChannels and CurrentChannel are Read-Only. LoadChannel and SaveChannel are Write-Only. Using set to perform operation.
hsDCAMREGLoadSettings hsDCAMREGSaveSettings	hsDCAMNone	Registry functions.	String for registry key LoadSettings and SaveSettings are Write-Only. Using set to perform operation.
hsDCAMQuadlet	Address, as unsigned long	Write or read a quadlet (4 bytes) from the DCAM registers.	Data : 4 bytes as unsigned long
hsDCAMFrameWidth hsDCAMFrameHeight hsDCAMFrameBitsDepth	hsDCAMNone	Standard DCAM information.	Read-Only. Unsigned Long
hsDCAMOptionalFeatures	hsDCAMNone	Standard DCAM information for specific features.	Read-Only. Boolean
hsDCAMAdvancedFeatures hsDCAMAdvancedFeaturesOffset	hsDCAMNone	Standard DCAM information for specific features.	hsDCAMAdvancedFeaturesOffset returns offset register. Read-Only.
hsDCAMPPIO hsDCAMPPIOOffset hsDCAMPPIOInput hsDCAMPPIOOutput	hsDCAMNone	Parallel IOs access.	hsDCAMPPIOOffset: Offset register. Read-Only. hsDCAMPPIOInput: unsigned long. Read-Only. hsDCAMPPIOOutput: unsigned long. Read-Write.
hsDCAMSIO hsDCAMSIOOffset hsDCAMSIOStatus hsDCAMSIOBaud hsDCAMSIODatabits hsDCAMSIOStopbits hsDCAMSIOParity hsDCAMSIOReceive hsDCAMSIOTransmit hsDCAMSIOConfig	hsDCAMNone	Serial IOs access.	hsDCAMSIOOffset: Offset register. Read-Only. hsDCAMSIOStatus: byte. Read-Only hsDCAMSIOBaud: unsigned long. Write-Only hsDCAMSIODatabits: unsigned long. Write-Only hsDCAMSIOStopbits: unsigned long. Write-Only hsDCAMSIOParity: unsigned long. Write-Only hsDCAMSIOReceive: boolean. Write-Only hsDCAMSIOTransmit: boolean. Write-Only hsDCAMSIOConfig: Used to perform configuration operation. Write-Only.
hsDCAMSIO4Bytes	hsDCAMNone	Read or write 4 bytes chunks from the Serial IOs.	Unsigned long representing 4 bytes.

Index Parameter	Option Parameter	Description	Range and Type
hsDCAMVideoFormat hsDCAMVideoMode hsDCAMVideoFrameRate	hsDCAMNone	DCAM video settings.	Short: 0-7
hsDCAMVideoFormat0 hsDCAMVideoFormat1 hsDCAMVideoFormat2 hsDCAMVideoFormat3 hsDCAMVideoFormat4 hsDCAMVideoFormat5 hsDCAMVideoFormat6 hsDCAMVideoFormat7	hsDCAMNone	Specific Video Format.	Write-Only.
hsDCAMMode	hsDCAMNone	DCAM Mode represent a standard DCAM mode. To be used with the enum hsDCAMMode.	See hsDCAMMode Enum below.
hsDCAMPartialScanDialog	hsDCAMNone	Default user interface for setting up proprietary (format 7) frame dimensions.	When get or set, the user interface is displayed for custom settings.
hsDCAMTimeout	hsDCAMNone	Timeout for acquisition.	Unsigned long, expressing Timeout in milliseconds. Above this value, the acquisition is aborted.

hsDCAMMode	Value	Description
hsDCAMMode160x120YUV444	0	160x120 Frame Size, YUV:444 Pixel Format
hsDCAMMode320x240YUV422	1	320x240 Frame Size, YUV:422 Pixel Format
hsDCAMMode640x480YUV411	2	640x480 Frame Size, YUV:411 Pixel Format
hsDCAMMode640x480YUV422	3	640x480 Frame Size, YUV:422 Pixel Format
hsDCAMMode640x480RGB	4	640x480 Frame Size, RGB Pixel Format
hsDCAMMode640x480Mono8	5	640x480 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode640x480Mono16	6	640x480 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMModeReserved	7	Reserved mode.
hsDCAMMode800x600YUV422	8	800x600 Frame Size, YUV:422 Pixel Format
hsDCAMMode800x600RGB	9	800x600 Frame Size, RGB Pixel Format
hsDCAMMode800x600Mono8	10	800x600 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode800x600Mono16	11	800x600 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMMode1024x768YUV422	12	1024x768 Frame Size, YUV:422 Pixel Format
hsDCAMMode1024x768RGB	13	1024x768 Frame Size, RGB Pixel Format
hsDCAMMode1024x768Mono8	14	1024x768 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode1024x768Mono16	15	1024x768 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMMode1280x960YUV422	16	1280x960 Frame Size, YUV:422 Pixel Format

hsDCAMMode	Value	Description
hsDCAMMode1280x960RGB	17	1280x960 Frame Size, RGB Pixel Format
hsDCAMMode1280x960Mono8	18	1280x960 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode1280x960Mono16	19	1280x960 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMMode1600x1200YUV422	20	1600x1200 Frame Size, YUV:422 Pixel Format
hsDCAMMode1600x1200RGB	21	1600x1200 Frame Size, RGB Pixel Format
hsDCAMMode1600x1200Mono8	22	1600x1200 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode1600x1200Mono16	23	1600x1200 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMModePartialScan	24	Vendor specific frame size and pixel format.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsDCAMParameterIndex	Enum, see table above.
<i>Option</i>	Option parameter, dependant on <i>Index</i>	Long	Long, or specific Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger mode
Dim TriggerMode As Integer
TriggerMode = AcqDevice1.GrabberDCAMParameter _
    (hsDCAMFeatureTriggerMode, hsDCAMTriggerMode)
' Sets the timeout value
AcqDevice1.GrabberDCAMParameter _
    (hsDCAMTimeout, hsDCAMNone) = 500
```

GrabberDCAMParameterSupport**Description**

Indexed property used to query if the parameter of a DCAM IEEE1394 device is present. If the specified device is not a generic DCAM device, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDCAMParameterSupport**(*Index* As hsDCAMParameterIndex , *Option* As Long , [*Device* As Variant] , [*Configuration* As Variant])

Type

Boolean

Range

True, False

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsDCAMParameterIndex	Enum, see table above.
<i>Option</i>	Option parameter, dependant on <i>Index</i>	Long	Long, or specific Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

The usage of this property depends on the desired query. See tables below:

<i>Index Parameter</i>	<i>Option Parameter</i>	Description	Option Description
hsDCAMFeatureBrightness hsDCAMFeatureAutoExposure hsDCAMFeatureSharpness hsDCAMFeatureWhiteBalance hsDCAMFeatureHue hsDCAMFeatureSaturation hsDCAMFeatureGamma hsDCAMFeatureShutter hsDCAMFeatureGain hsDCAMFeatureIris hsDCAMFeatureFocus hsDCAMFeatureTemperature hsDCAMFeatureTriggerDelay hsDCAMFeatureWhiteShading hsDCAMFeatureFrameRate hsDCAMFeatureZoom hsDCAMFeaturePan hsDCAMFeatureTilt hsDCAMFeatureOpticalFilter hsDCAMFeatureCaptureSize hsDCAMFeatureCaptureQuality	hsDCAMNone hsDCAMName hsDCAMUnits hsDCAMID hsDCAMPresence hsDCAMABSControl hsDCAMOnePush hsDCAMReadout hsDCAMONOFF hsDCAMAutoMode hsDCAMManualMode hsDCAMRangeMin hsDCAMRangeMax hsDCAMRangeABSMIn hsDCAMRangeABSMax hsDCAMValueLO hsDCAMValueHI hsDCAMValueABS	Standard DCAM Features.	hsDCAMNone for simple presence of feature, others for specific option's presence.
hsDCAMStrobe hsDCAMStrobeOffset	hsDCAMNone Short:0-3	Strobes	Presence of strobes, or specific strobe number if Integer option is provided.

<i>Index Parameter</i>	<i>Option Parameter</i>	Description	Option Description
hsDCAMFeatureStrobe0 hsDCAMFeatureStrobe1 hsDCAMFeatureStrobe2 hsDCAMFeatureStrobe3	hsDCAMNone hsDCAMName hsDCAMUnits hsDCAMID hsDCAMPresence hsDCAMABSControl hsDCAMOnePush hsDCAMReadout hsDCAMONOFF hsDCAMAutoMode hsDCAMManualMode hsDCAMPolarity hsDCAMRangeMin hsDCAMRangeMax hsDCAMRangeABSMin hsDCAMRangeABSMax hsDCAMValueLO hsDCAMValueHI hsDCAMValueABS	Quick access to the 4 DCAM strobes.	hsDCAMNone for simple presence of feature, others for specific option's presence.

Index Parameter	Option Parameter	Description	Option Description
hsDCAMFeatureTriggerMode	hsDCAMNone hsDCAMName hsDCAMUnits hsDCAMID hsDCAMPresence hsDCAMABSControl hsDCAMOnePush hsDCAMReadout hsDCAMONOFF hsDCAMAutoMode hsDCAMManualMode hsDCAMPolarity hsDCAMValueReadout hsDCAMRangeMin hsDCAMRangeMax hsDCAMRangeABSMIn hsDCAMRangeABSMax hsDCAMValueLO hsDCAMValueHI hsDCAMValueABS hsDCAMTriggerMode hsDCAMTriggerModeParameter hsDCAMTriggerMode0 hsDCAMTriggerMode1 hsDCAMTriggerMode2 hsDCAMTriggerMode3 hsDCAMTriggerMode4 hsDCAMTriggerMode5 hsDCAMTriggerModeVendor0 hsDCAMTriggerModeVendor1 hsDCAMTriggerSource hsDCAMTriggerSource0 hsDCAMTriggerSource1 hsDCAMTriggerSource2 hsDCAMTriggerSource3 hsDCAMTriggerSourceSoftware hsDCAMTriggerDoSoftware	Quick access to the DCAM Triggers	hsDCAMNone for simple presence of feature, others for specific option's presence.
hsDCAMVersion hsDCAMCameraName hsDCAMCameraVendor hsDCAMCameraUID hsDCAMMaxSpeed	hsDCAMNone	Standard DCAM informations.	Presence of information. Always true.
hsDCAMPowerControl hsDCAMIEEE1394B	hsDCAMNone	Standard DCAM settings.	Presence of settings.
hsDCAMMEMNumChannels hsDCAMMEMCurrentChannel hsDCAMMEMLoadChannel hsDCAMMEMSaveChannel	hsDCAMNone	Memory channels functions.	Presence of Memory Channels.
hsDCAMREGLoadSettings hsDCAMREGSaveSettings	hsDCAMNone	Registry functions.	Presence of Registry functions. Always true.

Index Parameter	Option Parameter	Description	Option Description
hsDCAMQuadlet	Address, as unsigned long	Write or read a quadlet (4 bytes) from the DCAM registers.	Presence of Quadlet read-write. Always true.
hsDCAMFrameWidth hsDCAMFrameHeight hsDCAMFrameBitsDepth	hsDCAMNone	Standard DCAM information.	Presence of information. Always true.
hsDCAMOptionalFeatures	hsDCAMNone	Standard DCAM information for specific features.	Presence of settings.
hsDCAMAdvancedFeatures hsDCAMAdvancedFeaturesOffset	hsDCAMNone	Standard DCAM information for specific features.	Presence of settings.
hsDCAMPPIO hsDCAMPPIOOffset hsDCAMPPIOInput hsDCAMPPIOOutput	hsDCAMNone	Parallel IOs access.	Presence of Parallel IOs.
hsDCAMSIO hsDCAMSIOOffset hsDCAMSIOStatus hsDCAMSIOBaud hsDCAMSIODatabits hsDCAMSIOStopbits hsDCAMSIOParity hsDCAMSIOReceive hsDCAMSIOTransmit hsDCAMSIOConfig	hsDCAMNone	Serial IOs access.	Presence of Serial IOs.
hsDCAMSIO4Bytes	hsDCAMNone	Read or write 4 bytes chunks from the Serial IOs.	Presence of Serial IOs.
hsDCAMVideoFormat hsDCAMVideoMode hsDCAMVideoFrameRate	Short: 0-7	DCAM video settings.	hsDCAMVideoFormat: Presence of format. hsDCAMVideoMode: Presence of mode under current format. hsDCAMVideoFrameRate: Presence of frame rate under current format and mode.
hsDCAMVideoFormat0 hsDCAMVideoFormat1 hsDCAMVideoFormat2 hsDCAMVideoFormat3 hsDCAMVideoFormat4 hsDCAMVideoFormat5 hsDCAMVideoFormat6 hsDCAMVideoFormat7	hsDCAMNone hsDCAMVideoMode0 hsDCAMVideoMode1 hsDCAMVideoMode2 hsDCAMVideoMode3 hsDCAMVideoMode4 hsDCAMVideoMode5 hsDCAMVideoMode6 hsDCAMVideoMode7 Short: 0-7	Specific Video Format. Can be used in combination with a mode for support inquiry.	hsDCAMNone: Presence of specified format. hsDCAMVideoModeX: Presence of ModeX under specified format. Short: 0-7: Presence of Mode X under specified format.
hsDCAMMode	See hsDCAMMode Enums below.	DCAM Mode represent a standard DCAM mode. To be used with the enum hsDCAMMode.	Presence of predefined DCAM Mode. See hsDCAMMode Enums below.

Index Parameter	Option Parameter	Description	Option Description
hsDCAMPartialScanDialog	hsDCAMNone	Default user interface for setting up proprietary (format 7) frame dimensions.	If presence of Format 7, returns true; user interface is available.
hsDCAMTimeout	hsDCAMNone	Timeout for acquisition.	Always true.

hsDCAMMode	Value	Description
hsDCAMMode160x120YUV444	0	160x120 Frame Size, YUV:444 Pixel Format
hsDCAMMode320x240YUV422	1	320x240 Frame Size, YUV:422 Pixel Format
hsDCAMMode640x480YUV411	2	640x480 Frame Size, YUV:411 Pixel Format
hsDCAMMode640x480YUV422	3	640x480 Frame Size, YUV:422 Pixel Format
hsDCAMMode640x480RGB	4	640x480 Frame Size, RGB Pixel Format
hsDCAMMode640x480Mono8	5	640x480 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode640x480Mono16	6	640x480 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMModeReserved	7	Reserved mode.
hsDCAMMode800x600YUV422	8	800x600 Frame Size, YUV:422 Pixel Format
hsDCAMMode800x600RGB	9	800x600 Frame Size, RGB Pixel Format
hsDCAMMode800x600Mono8	10	800x600 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode800x600Mono16	11	800x600 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMMode1024x768YUV422	12	1024x768 Frame Size, YUV:422 Pixel Format
hsDCAMMode1024x768RGB	13	1024x768 Frame Size, RGB Pixel Format
hsDCAMMode1024x768Mono8	14	1024x768 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode1024x768Mono16	15	1024x768 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMMode1280x960YUV422	16	1280x960 Frame Size, YUV:422 Pixel Format
hsDCAMMode1280x960RGB	17	1280x960 Frame Size, RGB Pixel Format
hsDCAMMode1280x960Mono8	18	1280x960 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode1280x960Mono16	19	1280x960 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMMode1600x1200YUV422	20	1600x1200 Frame Size, YUV:422 Pixel Format
hsDCAMMode1600x1200RGB	21	1600x1200 Frame Size, RGB Pixel Format
hsDCAMMode1600x1200Mono8	22	1600x1200 Frame Size, Monochrome 8 bits Pixel Format
hsDCAMMode1600x1200Mono16	23	1600x1200 Frame Size, Monochrome 16 bits Pixel Format
hsDCAMModePartialScan	24	Vendor specific frame size and pixel format.

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```

' Query the presence of Parallel IOs
Dim ParallelIOSupport As Boolean
ParallelIOSupport = AcqDevice1.GrabberDCAMParameterSupport _
    (hsDCAMPPIO, hsDCAMNone)

' Query the presence of DCAM Mode 160x120
Dim Mode160x120Support As Boolean
Mode160x120Support = AcqDevice1.GrabberDCAMParameterSupport _
    (hsDCAMMode, hsDCAMMode160x120YUV444)

```

GrabberDCAMModeSupport**Description**

Indexed property used to query the presence of a specific VideoFormat/VideoMode/VideoFrameRate of a DCAM IEEE1394 device. If the specified device is not a generic DCAM device, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDCAMModeSupport**(*Format* As Integer , *Mode* As Integer , *FrameRate* As Integer , [*Device* As Variant] , [*Configuration* As Variant])

Type

Boolean

Range

True, False

Parameter(s)

Name	Description	Type	Range
<i>Format</i>	Video Format.	Integer	Minimum: 0 Maximum: 7
<i>Mode</i>	Video Mode.	Integer	Minimum: 0 Maximum: 7
<i>FrameRate</i>	Video Frame Rate.	Integer	Minimum: 0 Maximum: 7 See FrameRate table below.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Frame Rate	Value
1.875 frames per second	0
3.75 frames per second	1
7.5 frames per second	2
15 frames per second	3

Frame Rate	Value
30 frames per second	4
60 frames per second	5
120 frames per second	6
240 frames per second	7

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Query the presence of DCAM Mode 160x120, 1.875 fps
Dim Mode160x120Support As Boolean
Mode160x120Support = AcqDevice1.GrabberDCAMModeSupport(0, 0, 0)
```

GrabberDepict4x1InputPort

Description

Access to the lines of the input port of an Opteon DepictMaster 4x1 controller board. This parameter is not indexed by a configuration, but by a device. If the specified device is not a DepictMaster 4x1 controller board, an error is fired and no internal actions are taken. Read only.

Syntax

control.**GrabberDepict4x1InputPort**(*Device* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the controller board device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Latched</i>	If True, waits for the LatchEnable signal and reads the value of the 8-bit digital input port. If False, returns the present value of the 8-bit digital input port.	Boolean	True, False

Remark(s)

This property is read only since it is not possible to change the status of an input-only port.

Example

```
Dim PortValue As Integer
PortValue = AcqDevice1.GrabberDepict4x1InputPort("DEPICT4X1 0", False)
```

GrabberDepict4x1OutputPort

Description

Access to the lines of the output port of an Opteon DepictMaster 4x1 controller board. This parameter is not indexed by a configuration, but by a device. If the specified device is not a DepictMaster 4x1 controller board, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDepict4x1OutputPort**(*Device* As Variant) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the controller board device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of an Opteon Depict Master 4x1 controller board (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```
AcqDevice1.GrabberDepict4x1("DEPICT4X1 0") = 225
```

```
Dim LastOutput As Integer
```

```
LastOutput = GrabberDepict4x1("DEPICT4X1 0")
```

GrabberDepict4x1Parameter

Description

Indexed parameter of an Opteon DepictMaster 4x1 controller board for a given configuration. If the specified device is not a DepictMaster 4x1 controller board, an error is fired and no internal actions are taken.

Syntax

control.**Depict4x1Parameter**.(*Index* As hsDepict4x1ParameterIndex, [*Device* As Variant], [*Configuration* As Variant]) [= *Value* As Variant]

Type

Variant

Range

The range depends on the selected parameter.

Index	Description	Type	Range
hsDepict4x1CameraPort	Specifies which DepictMaster 4x1 camera port acquires the image.	Short	Minimum: 0 Maximum: 3

Index	Description	Type	Range
hsDepict4x1PExposure	Sets the exposure time, in microseconds, for primary acquisitions by the specified camera.	Long	Minimum: 50 Maximum: 100,000,000 Step: 50
hsDepict4x1CameraGain	Sets the gain for the specified camera.	Short	Minimum: 0 Maximum: 255
hsDepict4x1CameraOffset	Sets the offset for the specified camera.	Short	Minimum: 0 Maximum: 255
hsDepict4x1Strobe	Sets the strobe logic for the specified camera.	hsDepict4x1ParameterStrobe	Enum. See table below.
hsDepict4x1TriggerMode	Acquisition trigger mode for the specified camera.	hsDepict4x1ParameterTriggerMode	Enum. See table below.
hsDepict4x1TriggerPeriod	In the Timebase trigger mode, specifies the minimum time period between two acquisition requests. In ExternalA and Master trigger modes, specifies the hold-off time between the trigger event and the start of the acquisition.	Long	Minimum: 0 Maximum: Unlimited
hsDepict4x1TriggerEdge	In Master trigger mode, sets the polarity of the event on the master trigger input. In ExternalA trigger mode, sets the polarity of the event on the external trigger input of the selected camera.	hsDepict4x1ParameterTriggerEdge	Enum. See table below.
hsDepict4x1Timeout	Timeout value in milliseconds.	Integer	Minimum: 1 Maximum: Unlimited

hsDepict4x1ParameterStrobe	Description
hsDepict4x1StrobeOff	No strobe.
hsDepict4x1StrobePrimaryOnly	Strobe on Primary ROI only.

hsDepict4x1ParameterTriggerMode	Description
hsDepict4x1TriggerModeImmediate	The acquisition request is processed immediately after all other pending acquisition requests are completed; no trigger is used.
hsDepict4x1TriggerModeTimebase	The acquisition request is not be processed until the minimum amount of time has elapsed since the last acquisition request. The minimum time is set by the hsDepict4x1TriggerPeriod parameter.
hsDepict4x1TriggerModeExternalA	Uses the external trigger input for the specified camera.
hsDepict4x1TriggerModeMaster	Uses the external master trigger input for the specified camera.

hsDepict4x1ParameterTriggerMode	Description
hsDepict4x1TriggerModeSlave	The acquisition request on the selected camera will be triggered by the next non-slave acquisition. Synchronized acquisitions are accomplished by placing an acquisition request in the pipeline for the slave camera and placing another acquisition request for the non-slave camera in the pipeline. The two cameras will be acquired simultaneously. The first Wait statement will retrieve the image acquired by the slave camera and the second Wait statement will retrieve the image from the non-slave camera.

hsDepict4x1ParameterTriggerEdge	Description
hsDepict4x1TriggerEdgeFalling	Falling-edge polarity.
hsDepict4x1TriggerEdgeRising	Rising-edge polarity.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsDepict4x1ParameterIndex	Enum, see table above
<i>Device</i>	Index number or name of the controller board device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger mode
Dim Mode As hsDepict4x1ParameterTriggerMode
Mode = AcqDevice1.GrabberDepict4x1Parameter(hsDepict4x1TriggerMode, _
    "DEPICT4X1 0", "Configuration0")

' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberDepict4x1Parameter(hsDepict4x1Timeout)= 5000
```

GrabberDirectShowParameter

Description

Indexed property used to specify the parameters of a generic Direct Show device (typically, a IEEE 1394 camera). If the specified device is not a generic Direct Show device, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDirectShowParameter**(*Index* As hsDirectShowParameterIndex , [*Device* As Variant] , [*Configuration* As Variant]) [= *NewValue* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* parameter. See tables below:

<i>Index</i> Parameter	Description	Property Type	Range
hsDirectShowInputSource	Input Source used for grabbing images.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowOutputSource	Input Source is used to output the acquired image.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowTimeout	Timeout value in milliseconds.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowBrightnessValue	Brightness value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowBrightnessFlags	Flag state for Brightness value.	hsDirectShowParameterIndex	See table below.
hsDirectShowContrastValue	Contrast value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowContrastFlags	Flag state for Contrast value.	hsDirectShowParameterIndex	See table below.
hsDirectShowHueValue	Hue value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowHueFlags	Flag state for Hue value.	hsDirectShowParameterIndex	See table below.
hsDirectShowSaturationValue	Saturation value.	Integer	See GrabberDirectShowParameterRange property.

Index Parameter	Description	Property Type	Range
hsDirectShowSaturationFlags	Flag state for Saturation value.	hsDirectShowParameterIndex	See table below.
hsDirectShowSharpnessValue	Sharpness value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowSharpnessFlags	Flag state for Sharpness value.	hsDirectShowParameterIndex	See table below.
hsDirectShowGammaValue	Gamma value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowGammaFlags	Flag state for Gamma value.	hsDirectShowParameterIndex	See table below.
hsDirectShowColorEnableValue	Color Enable value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowColorEnableFlags	Flag state for Color Enable value	hsDirectShowParameterIndex	See table below.
hsDirectShowWhiteBalanceValue	White Balance value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowWhiteBalanceFlags	Flag state for White Balance value.	hsDirectShowParameterIndex	See table below.
hsDirectShowBacklightValue	Backlight value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowBacklightFlags	Flag state for Backlight value	hsDirectShowParameterIndex	See table below.
hsDirectShowGainValue	Gain value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowGainFlags	Flag state for Gain value.	hsDirectShowParameterIndex	See table below.
hsDirectShowPanValue	Pan value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowPanFlags	Flag state for Pan value.	hsDirectShowParameterIndex	See table below.

Index Parameter	Description	Property Type	Range
hsDirectShowTiltValue	Tilt value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowTiltFlags	Flag state for Tilt value.	hsDirectShowParameterIndex	See table below.
hsDirectShowRollValue	Roll value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowRollFlags	Flag state for Roll value.	hsDirectShowParameterIndex	See table below.
hsDirectShowZoomValue	Zoom value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowZoomFlags	Flag state for Roll value.	hsDirectShowParameterIndex	See table below.
hsDirectShowExposureValue	Exposure value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowExposureFlags	Flag state for Exposure value.	hsDirectShowParameterIndex	See table below.
hsDirectShowIrisValue	Iris value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowIrisFlags	Flag state for Iris value.	hsDirectShowParameterIndex	See table below.
hsDirectShowFocusValue	Focus value.	Integer	See GrabberDirectShowParameterRange property.
hsDirectShowFocusFlags	Flag state for Focus value.	hsDirectShowParameterIndex	See table below.
hsDirectShowGuid	Returns a unique Guid string that identifies the device. Read only.	String	Not applicable.
hsDirectShowStreamFormat Count	Returns the number of valid Stream Formats for the specified device. Read only	Integer	Not applicable.

Index Parameter	Description	Property Type	Range
hsDirectShowStreamFormat	Read: String that identifies the current Stream Format. Write: Index of the StreamFormat to be applied.	Read: String Write: Integer	Read: Not applicable. Write: Minimum: 0 Maximum: X
hsDirectShowStreamFormatString + X	Returns a string that identifies the supported Stream Format of index value X. Read only.	Integer	hsDirectShowStreamFormatString +0, hsDirectShowStreamFormatString +1, ... hsDirectShowStreamFormatString + (result of hsDirectShowStreamFormatCount) -1,

hsDirectShowParameterFlags	Value	Description
hsDirectShowFlagAuto	0x01	Signals that the parameter will be set automatically
hsDirectShowFlagManual	0x02	Signals that the parameter can be set manually by the user.

hsDirectShowParameterInputSource	Value	Description
hsDirectShowInputSourceDevice	0	Device Input Source
hsDirectShowInputSourceNone	1	No grab is performed. The output image will be the last grabbed image on the selected output source.

hsDirectShowParameterOutputSource	Value	Description
hsDirectShowOutputSource1	0	Image output through the Red (Gray) Input Source.
hsDirectShowOutputSource2	1	Image output through the Green Input Source.
hsDirectShowOutputSource3	2	Image output through the Blue Input Source.
hsDirectShowOutputSource4	3	Image output through the Y input source (intensity). Based on the YUV representation.

Parameter(s)

Name	Description	Type	Range
Index	Index of the parameter.	hsDirectShowParameterIndex	Enum, see table above.

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current grabbing output source
Dim OutputSource As Integer
OutputSource = AcqDevice1.GrabberDirectShowParameter _
    (hsDirectShowOutputSource, "Direct Show Device 0")
' Sets the timeout value
AcqDevice1.GrabberDirectShowParameter _
    (hsDirectShowTimeout, "Direct Show Device 0") = 500
```

GrabberDirectShowParameterRange

Description

This property retrieves the Range (Min, Max, SteppingDelta), the default value and the possible flag states (Flags) and returns TRUE if the operation succeeds. Read only.

Syntax

control.**GrabberDirectShowParameterRange**(*Index* As hsDirectShowParameterIndex, *Min* as Long, *Max* As Long, *SteppingDelta* As Long, *Flags* as Long, [*Device*] VARIANT , [*Configuration*] VARIANT)

Type

Boolean

Range

True, False

Parameter(s)

<i>Index</i> Parameter	Description
hsDirectShowInputSource	Input Source used for grabbing images.
hsDirectShowOutputSource	Output Source used to output the acquired image.
hsDirectShowTimeout	Timeout value in milliseconds.
hsDirectShowBrightnessValue	Brightness value.
hsDirectShowBrightnessFlags	Flag state for Brightness value.
hsDirectShowContrastValue	Contrast value.
hsDirectShowContrastFlags	Flag state for Contrast value.

Index Parameter	Description
hsDirectShowHueValue	Hue value.
hsDirectShowHueFlags	Flag state for Hue value.
hsDirectShowSaturationValue	Saturation value.
hsDirectShowSaturationFlags	Flag state for Saturation value.
hsDirectShowSharpnessValue	Sharpness value.
hsDirectShowSharpnessFlags	Flag state for Sharpness value.
hsDirectShowGammaValue	Gamma value.
hsDirectShowGammaFlags	Flag state for Gamma value.
hsDirectShowColorEnableValue	Color Enable value.
hsDirectShowColorEnableFlags	Flag state for Color Enable value
hsDirectShowWhiteBalanceValue	White Balance value.
hsDirectShowWhiteBalanceFlags	Flag state for White Balance value.
hsDirectShowBacklightValue	Backlight value.
hsDirectShowBacklightFlags	Flag state for Backlight value
hsDirectShowGainValue	Gain value.
hsDirectShowGainFlags	Flag state for Gain value.
hsDirectShowPanValue	Pan value.
hsDirectShowPanFlags	Flag state for Pan value.
hsDirectShowTiltValue	Tilt value.
hsDirectShowTiltFlags	Flag state for Tilt value.
hsDirectShowRollValue	Roll value.
hsDirectShowRollFlags	Flag state for Roll value.
hsDirectShowZoomValue	Zoom value.
hsDirectShowZoomFlags	Flag state for Roll value.
hsDirectShowExposureValue	Exposure value.
hsDirectShowExposureFlags	Flag state for Exposure value.
hsDirectShowIrisValue	Iris value.
hsDirectShowIrisFlags	Flag state for Iris value.
hsDirectShowFocusValue	Focus value.
hsDirectShowFocusFlags	Flag state for Focus value.
hsDirectShowGuid	Guid string that identifies the device.

Example

```
' Get the valid range for the input source parameter
Dim Minimum as Long
Dim Maximum as Long
Dim Delta as Long
Dim Flags as Long
```

```

Dim Success as Boolean
Dim i as Integer
Dim Range as String
Success = AcqDevice1.GrabberDirectShowParameterRange _
    ( hsDirectShowInputSource, Minimum, Maximum, Delta, Flags, _
    "Direct Show Device 0" )
' Output the valid range for input source in a string
Range = "The valid range for Input Source is:"
For i = Minimum to Maximum Step Delta
    Range = Range & " " & Cstr(i)
Next i
MsgBox Range

```

GrabberDt3152OutputPort

Description

Access to the lines of the output port of a Data Translation 3152 frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a Data Translation 3152 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDt3152OutputPort**(*Device* As Variant) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index or name of the input device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of a Data Translation 3152 (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```

AcqDevice1.GrabberDt3152OutputPort("DT-3152 0") = 24
Dim LastOutput As Integer
LastOutput = AcqDevice1.GrabberDt3152OutputPort("DT-3152 0")

```

GrabberDt3152Parameter

Description

Indexed parameter of a Data Translation 3152 frame grabber for a given configuration. If the specified device is not a Data Translation 3152 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDt3152Parameter**(*Index* As hsDt3152ParameterIndex, [*Device* As Variant],
[*Configuration* As Variant]) [= *Value* As Variant]

Type

Variant

Range

The range depends on the selected parameter.

Index	Description	Type	Range
hsDt3152InputSource	Input source of the frame grabber.	Integer	Minimum: 0 Maximum: 3
hsDt3152SyncSource	Sync source of the frame grabber.	Integer	Minimum: 0 Maximum: 3
hsDt3152FilteringEnabled	AC filtering enabled/ disabled.	Boolean	True, False
hsDt3152BlackLevel	Black level reference.	Long	Min: 1275 Maximum: 306000
hsDt3152WhiteLevel	White level reference.	Long	Min: 351100 Maximum: 1009000
hsDt3152TriggerType	Trigger type.	hsDtTriggerType	See table below.
hsDt3152Timeout	Time (ms) to wait for an acquisition to complete before failing.	Long	Min: 1 Maximum: 214748

Name	Value	Description
hsDtTriggerNone	0	No trigger.
hsDtTriggerHighToLow	1	Trigger on high to low pulse.
hsDtTriggerLowToHigh	2	Trigger on low to high pulse.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsDt3152ParameterIndex	Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger type
Dim Type As hsDtTriggerType
Type = AcqDevice1.GrabberDt3152Parameter(hsDt3152TriggerType, _
    "DT-3152 0", "Configuration0" )

' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberDt3152Parameter(hsDt3152Timeout)= 5000
```

GrabberDt3155OutputPort

Description

Access to the lines of the output port of a Data Translation 3155 frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a Data Translation 3155 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDt3155OutputPort**(*Device* As Variant) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of a Data Translation 3155 (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```
AcqDevice1.GrabberDt3155OutputPort("DT-3155 0") = 12
Dim LastOutput As Integer
LastOutput = AcqDevice1.GrabberDt3155OutputPort("DT-3155 0")
```

GrabberDt3155Parameter

Description

Indexed parameter of a Data Translation 3152 frame grabber for a given configuration. If the specified device is not a Data Translation 3152 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDt3155Parameter**(*Index* As hsDt3155ParameterIndex , [*Device* As Variant]
 , [*Configuration* As Variant]) [= *Value* As Variant]

Type

Variant

Range

The range depends on the selected parameter.

Index	Description	Type	Range
hsDt3155InputSource	Input source of the frame grabber.	Integer	Minimum: 0 Maximum: 3
hsDt3155SyncSource	Sync source of the frame grabber.	Integer	Minimum: 0 Maximum: 3
hsDt3155FilteringEnabled	AC filtering enabled/ disabled.	Boolean	True, False
hsDt3155BlackLevel	Black level reference.	Long	Min: 1275 Maximum: 306000
hsDt3155WhiteLevel	White level reference.	Long	Min: 351100 Maximum: 1009000
hsDt3155TriggerType	Trigger type.	hsDtTriggerType	See table below.
hsDt3155Timeout	Time (ms) to wait for an acquisition to complete before failing.	Long	Min: 1 Maximum: 214748

Name	Value	Description
hsDtTriggerNone	0	No trigger.
hsDtTriggerHighToLow	1	Trigger on high to low pulse.
hsDtTriggerLowToHigh	2	Trigger on low to high pulse.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsDt3155ParameterIndex	Enum, see table above
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationName - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger type
Dim Type As hsDtTriggerType
Type = AcqDevice1.GrabberDt3155Parameter(hsDt3155TriggerType, _
    "DT-3155 0", "Configuration0")

' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberDt3155Parameter(hsDt3155Timeout)= 5000
```

GrabberDt3157InputPort

Description

Access to the lines of the input port of an Data Translation 3157 frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a Data Translation 3157 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDt3157InputPort**(*Device* As Variant).

Type

Integer

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Range

Minimum: 0x0

Maximum: 0xF

Remark(s)

This property is read only As it is not possible to change the status of an input-only port.

Example

```
Dim PortValue As Integer
PortValue = AcqDevice1.GrabberDt3157InputPort("DT3157 0")
```

GrabberDt3157OutputPort

Description

Access to the lines of the output port of a Data Translation 3157 frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a Data Translation 3157 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberDt3157OutputPort**(*Device* As Variant) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of a Data Translation 3157 (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```
AcqDevice1.GrabberDt3157OutputPort("DT-3157 0") = 102
```

```
Dim LastOutput As Integer
```

```
LastOutput = AcqDevice1.GrabberDt3157OutputPort("DT-3157 0")
```

GrabberDt3157Parameter

Description

Indexed parameter of a Data Translation 3157 frame grabber for a given configuration. If the specified device is not a Data Translation 3157 frame grabber, an error is fired and no internal actions are taken.

Syntax

```
control.GrabberDt3157Parameter(Index As hsDt3157ParameterIndex , [ Device As Variant]  
 , [ Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The valid range for this property is as follows:

Index	Description	Type	Range
hsDt3157TriggerType	Trigger type.	hsDtTriggerType	See table below.
hsDt3157Timeout	Time (ms) to wait for an acquisition to complete before failing.	Long	Min: 1 Maximum: 214748

Name	Value	Description
hsDtTriggerNone	0	No trigger.
hsDtTriggerHighToLow	1	Trigger on high to low pulse.
hsDtTriggerLowToHigh	2	Trigger on low to high pulse.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsDt3157Parameter Index	Enum, see table above
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger type
Dim Type As hsDtTriggerType
Type = AcqDevice1.GrabberDt3157Parameter(hsDt3157TriggerType, _
    "DT-3157 0", "Configuration0")

' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberDt3157Parameter(hsDt3157Timeout)= 5000
```

GrabberGocatorParameter

Description

Indexed parameter of a device for a given configuration.

Syntax

```
control.GrabberGocatorParameter(Index As Integer, [Device As Variant]  
    , [Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The range depends on the selected parameter. See **GrabberGocatorParameterIndex** for additional parameters.

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the *ConfigurationDefault* property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

GrabberGocatorParameterCategory

Description

Indexed parameter of a device for a given configuration, to retrieve the category of a parameter.

Syntax

control.**GetGrabberGocatorParameterCategory**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

String as Variant

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterFlags

Description

Indexed parameter of a device for a given configuration, to retrieve the flags of a parameter.

Syntax

control.**GetGrabberGocatorParameterFlags**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

Integer

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterImpact

Description

Indexed parameter of a device that gives the impacted parameters when changed.

Syntax

control.**GetGrabberGocatorParameterImpact**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

String as Variant

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterIndex

Description

This property is used to retrieve the index of a given parameter's name for a device.

Syntax

control.**GrabberGocatorParameterIndex**(*Name* As Variant, [*Device* As Variant], [*Configuration* As Variant])

Type

Integer

Range

0 or more
-1 if not found

Return

Can be one of these parameters:

Name	Description	Type	Range
<i>hsGocatorIP0</i> <i>hsGocatorIP1</i> <i>hsGocatorIP2</i> <i>hsGocatorIP3</i>	IP Address, expressed as 4 values. It takes the form: IP0.IP1.IP2.IP3	long	Minimum: 0 Maximum: 255 Default: 192.168.1.10
<i>hsGocatorDatatype</i>	Current data type or acquisition mode of the Gocator. The data type needs to match the current Gocator setting, and the output must be properly set in the Gocator web-interface, before starting the acquisition.	hsGocatorSupportedDataType	Enum, see following table.
<i>hsGocatorRotation</i>	Allows to transpose or rotate the image captured from the Gocator.	hsGocatorSupportedImageRotation	Enum, see following table.
<i>hsGocatorScanLines</i>	The number of profiles to capture in profile mode (normal or intensity). The profiles will be stitched to create the final image.	long	Minimum: 0 Maximum: 1000000 Default: 100
<i>hsGocatorTimeout</i>	Timeout value in milliseconds (ms).	long	Minimum: 0 Maximum: 1000000 Default: 30000
<i>hsGocatorEncoderSync</i>	If True, the image will be reordered to maintain the profile acquired at the encoder's tick at the top (or first line) of the image. Useful in rotating applications.	boolean	True, False
<i>hsGocatorConfiguration</i>	Gocator configuration file name.	String	Not applicable
<i>hsGocatorStart</i>	If True, the Gocator is started. If False, the Gocator is stopped.	boolean	True, False

hsGocatorSupportedDataType Parameter	Value	Description
hsGocatorProfileMode	11	Profile acquisition mode
hsGocatorProfileIntensityMode	12	Intensity Profile acquisition mode
hsGocatorProfileAndIntensityMode	13	Profile and Intensity Profile acquisition mode

hsGocatorSupportedDataType Parameter	Value	Description
hsGocatorPartMode	14	Whole Part acquisition mode
hsGocatorPartIntensityMode	15	Intensity Whole Part acquisition mode
hsGocatorPartAndIntensityMode	16	Whole Part and Intensity Whole Part acquisition mode

hsGocatorSupportedImageRotation Parameter	Value	Description
hsGocatorImageRotationNormal	0	Final image is normal.
hsGocatorImageRotationTranspose	1	Final image is transposed.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterMax

Description

Indexed parameter of a device to retrieve the maximum value of a setting.

Syntax

```
control.GetGrabberGocatorParameterMax(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

Variant

Range

The range depends on the selected parameter.

Remark(s)

In case of an enumerated property, the function will return the list of possible values.
This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterMin

Description

Indexed parameter of a device to retrieve the minimum value of a setting.

Syntax

```
control.GetGrabberGocatorParameterMin(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

Variant

Range

The range depends on the selected parameter.

Remark(s)

In case of an enumerated property, the function will return the list of possible values.
This property applies to the default configuration if the Configuration parameter is not provided.

provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterName

Description

Indexed property to retrieve the name of the parameter of a device.

Syntax

control.**GrabberGocatorParameterName**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

String as Variant

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParametersCount

Description

This property is used to get the number of available parameters for a device.

Syntax

control.**GrabberGocatorParametersCount**([*Device* As Variant], [*Configuration* As Variant])

Type

Integer

Range

0 or more.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGocatorParameterType

Description

Indexed parameter of a device for a given configuration, to retrieve the type of a parameter.

Syntax

control.**GetGrabberGocatorParameterType**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

Integer

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberItIcAsyncInputPort

Description

Access to the lines of the input port of an Coreco Imaging IC-ASYNC frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not an IC-ASYNC frame grabber, an error is fired and no internal actions are taken. Read only.

Syntax

control.**GrabberItIcAsyncInputPort**(*Device* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

This property is read only As it is not possible to change the status of an input-only port.

Example

```
Dim PortValue As Integer
```

```
PortValue = AcqDevice1.GrabberItIcAsyncInputPort("IC-ASYNC 0")
```

GrabberItIcAsyncOutputPort

Description

Access to the lines of the output port of an Coreco Imaging IC-ASYNC frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not an IC-ASYNC frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberItIcAsyncOutputPort**(*Device* As Variant) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of an Coreco Imaging IC-ASYNC frame grabber (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```
AcqDevice1.GrabberItIcAsyncOutputPort("IC-ASYNC 0") = 225
Dim LastOutput As Integer
LastOutput = AcqDevice1.GrabberItIcAsyncOutputPort("IC-ASYNC 0")
```

GrabberItIcAsyncParameter**Description**

Indexed parameter of an Coreco Imaging IC-ASYNC frame grabber for a given configuration. If the specified device is not an IC-ASYNC frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberItIcAsyncParameter**(*Index* As C , [*Device* As Variant] , [*Configuration* As Variant]) [= *Value* As Variant]

Type

Variant

Range

The range depends on the selected parameter.

hsPx610aParameterIndex	Description	Type	Range
hsItIcAsyncConfigurationFile	Configuration file name.	String	Not applicable.
hsItIcAsyncGrabChannel0	Enabled/disabled status of Channel 0.	Boolean	True, False.
hsItIcAsyncGrabChannel1	Enabled/disabled status of Channel 1.	Boolean	True, False.
hsItIcAsyncGrabChannel2	Enabled/disabled status of Channel 2.	Boolean	True, False.
hsItIcAsyncGrabChannel3	Enabled/disabled status of Channel 3.	Boolean	True, False.
hsItIcAsyncOutputChannel	Selects the output to be read.	Integer	0, 1, 2, 3.
hsItIcAsyncExtTrigChannel0	Specifies if an external trigger is needed on this channel.	Boolean	True, False.
hsItIcAsyncExtTrigChannel1	Specifies if an external trigger is needed on this channel.	Boolean	True, False.
hsItIcAsyncExtTrigChannel2	Specifies if an external trigger is needed on this channel.	Boolean	True, False.
hsItIcAsyncExtTrigChannel3	Specifies if an external trigger is needed on this channel.	Boolean	True, False.
hsItIcAsyncTimeout	Timeout value in milliseconds (ms).	Integer	Min: 1 Maximum: unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsItIcAsyncParameterIndex	Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current output channel
Dim Channel As Integer
Channel = AcqDevice1.GrabberItIcAsyncParameter( _
    hsItIcAsyncOutputChannel, _
    "IC-ASYNC 0", _
    "Configuration0" )
' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberItIcAsyncParameter _
    (hsItIcAsyncTimeout) = 5000
```

GrabberItIcAsyncPortParameter

Description

Indexed property used to specify the parameters of the I/O ports of an Coreco Imaging IC-ASYNC frame grabber. If the specified device is not an IC-ASYNC frame grabber, an error is fired and no internal actions are taken.

Syntax

control.GrabberItIcAsyncPortParameter(*Index* As hsItIcAsyncPortParameterIndex, *Device* As Variant) [= *Value* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* parameter.

<i>Index Parameter</i>	Description	GrabberItIcAsyncPortParameter Property Range
hsItIcAsyncInputPortInterruptMode	Specifies what kind of transition will fire a ItIcInputInterrupt event.	hsItIcAsyncInputPortInterruptDisabled (=0) hsItIcAsyncInputPortInterruptRisingEdge (=1) hsItIcAsyncInputPortInterruptFallingEdge (=2)
hsItIcAsyncInputPortMode	Specifies if the 8-bit input port is not latched or if it is latched on a rising or a falling edge transition.	hsItIcAsyncInputPortNotLatched (=0) hsItIcAsyncInputPortLatchedRisingEdge (=1) hsItIcAsyncInputPortLatchedFallingEdge (=2)
hsItIcAsyncOutputPortStrobeLevel	Value for the output strobe on the 8-bit port for devices that require a strobe to latch data.	hsItIcAsyncOutputPortStrobeLo (=0) hsItIcAsyncOutputPortStrobeHi (=1)

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsItIcAsyncPortParameterIndex	Enum, see following table.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

<i>Index Parameter</i>	Value	Description
hsItIcAsyncInputPortInterruptMode	0	Specifies what transition will fire an ItIcInputInterrupt event.
hsItIcAsyncInputPortMode	1	Specifies if the 8-bit input port is not latched or if it is latched on a rising or a falling edge transition.
hsItIcAsyncOutputPortStrobeLevel	2	Value for the output strobe on the 8-bit port for devices that require a strobe to latch data.

Example

```

' Reads the input port interrupt mode
Dim IntMode As Integer
IntMode = AcqDevice1.GrabberItIcAsyncPortParameter( _
    hsItIcAsyncInputPortInterruptMode, _
    "IC-ASYNC 0", _
    "Configuration0" )
' Writes a new output port strobe level
AcqDevice1.GrabberItIcAsyncPortParameter _
    (hsItIcAsyncOutputPortStrobeLevels) = _
    hsItIcAsyncOutputPortStrobeLo

```

GrabberItiPcDigInputPort

Description

Access to the lines of the input port of an Coreco Imaging PC-DIG frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a PC-DIG frame grabber, an error is fired and no internal actions are taken. Read only.

Syntax

control.**GrabberItiPcDigInputPort**(*Device* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

This property is read only As it is not possible to change the status of an input-only port.

Example

```
Dim PortValue As Integer
```

```
PortValue = AcqDevice1.GrabberItiPcDigInputPort ("PCDIG 0")
```

GrabberItiPcDigOutputPort

Description

Access to the lines of the output port of an Coreco Imaging PC-DIG frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a PC-DIG frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberItiPcDigOutputPort**(*Device* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of an Coreco Imaging PC-DIG frame grabber (it is impossible to read an output-only port) but it returns the last value that was written to the port.

Example

```
Dim LastOutput As Integer
LastOutput = AcqDevice1.GrabberItiPcDigOutputPort ("PCDIG 0")
```

GrabberItiPcDigParameter**Description**

Indexed parameter of an Coreco Imaging PC-DIG frame grabber for a given configuration. If the specified device is not a PC-DIG frame grabber, an error is fired and no internal actions are taken.

Syntax

```
control.GrabberItiPcDigParameter(Index As hsltiPcDigParameterIndex,[Device As Variant],  
[ Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The range depends on the selected parameter:

hsltiPcDigParameterIndex	Description	Type	Range
hsltiPcDigConfigurationFile	Configuration file name.	String	Not applicable.
hsltiPcDigCameraPort	Camera port of the PC-DIG	Integer	Minimum: 0 Maximum: 0
hsltiPcDigExtTrigger	Specifies if an external trigger is needed.	Boolean	True, False
hsltiPcDigTimeout	Timeout value in milliseconds (ms).	Integer	Min: 1 Maximum: Unlimited
hsltiPcDigImageZoom	Hardware zoom. Default: 1	Integer	Min: 1 Maximum: 1
hsltiPcDigImageOriginX	Window Generator X origin (pixels). Default: 0	Integer	Minimum: 0 Maximum: 16384
hsltiPcDigImageOriginY	Window Generator Y origin (pixels). Default: 0	Integer	Minimum: 0 Maximum: 16384
hsltiPcDigImageWidth	Window Generator Width (pixels). Default: 0 (means config width)	Integer	Min: 64 Maximum: 16384
hsltiPcDigImageHeight	Window Generator Height (pixels). Default: 0 (means config height)	Integer	Min: 2 Maximum: 16384
hsltiPcDigTriggerSource	Trigger source.	hsltiPcDigTriggerInputSource	Enum, see table below.
hsltiPcDigTriggerPolarity	Trigger polarity.	hsltiPcDigTriggerEdgePolarity	Enum, see table below.
hsltiPcDigTriggerStartEvent	Start event.	hsltiPcDigTriggerEvent	Enum, see table below.
hsltiPcDigTriggerStopEvent	Stop event.	hsltiPcDigTriggerEvent	Enum, see table below.

hsItiPcDigTriggerInputSource	Description
hsItiPcDigDifferentialTrigger	Differential trigger
hsItiPcDigTtlTrigger	TTL trigger.
hsItiPcDigOptoCoupledTrigger	Opto Coupled Trigger.

hsItiPcDigTriggerEdgePolarity	Description
hsItiPcDigTriggerPolarityFallingEdge	Falling edge
hsItiPcDigTriggerPolarityRisingEdge	Rising edge

hsItiPcDigTriggerEvent	Description
hsItiPcDigEventDisable	Disabled.
hsItiPcDigHardwareTrigger0Assert	Based on hardware trigger 0 assert.
hsItiPcDigHardwareTrigger0Deassert	Based on hardware trigger 0 deassert.
hsItiPcDigHardwareTrigger1Assert	Based on hardware trigger 1 assert.
hsItiPcDigHardwareTrigger1Deassert	Based on hardware trigger 1 deassert.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter	hsItiPcDigParameterIndex	Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount-1

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

Example

```
' Reads the current Timeout
Dim Timeout As Integer
Timeout = AcqDevice1.GrabberItiPcDigParameter _
    (hsItiPcDigTimeout, "PCDIG 0", "Configuration0" )
' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberItiPcDigParameter (hsItiPcDigTimeout) = 5000
```

GrabberItiPcDigPortParameter

Description

Indexed property used to specify the parameters of the I/O ports of an Coreco Imaging PC-DIG frame grabber. If the specified device is not a PC-DIG frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberItiPcDigPortParameter**(*Index* As hsltiPcDigPortParameterIndex, *Device* As Variant) [= *Value* As Variant]

Type

Variant

Range

The range of this property depends on the Index parameter:

Index Parameter	Description	Type	Range
hsltiPcDigInputPortInterruptMode	The type of transition that will fire an ItiInportInterrupt event.	hsltiPcDigInputPortInterruptModes	Enum, see table below.
hsltiPcDigInputPortMode	Specifies if the 8-bit input port is not latched or if it is latched on a rising or on a falling edge transition.	hsltiPcDigInputPortModes	Enum, see table below.
hsltiPcDigOutputPortStrobeLevel	Value for output strobe on the 8-bit port for devices that require a strobe to latch data.	hsltiPcDigOutputPortStrobeLevels	Enum, see table below.

hsltiPcDigInputPortInterruptMode	Description
hsltiPcDigInputPortInterruptDisabled	Hardware interrupts on the input port do not fire an ItiInportInterrupt event.
hsltiPcDigInputPortInterruptRisingEdge	ItiInportInterrupt events are fired on the rising edge of transitions on the interrupt input of the I/O port.
hsltiPcDigInputPortInterruptFallingEdge	ItiInportInterrupt events are fired on the falling edge of transitions on the interrupt input of the I/O port.

hsltiPcDigInputPortMode	Description
hsltiPcDigInputPortNotLatched	The value read from the input port is the current value present on the port.
hsltiPcDigInputPortLatchedRisingEdge	The value read from the input port corresponds to the value that was present on the input port on the rising edge of the last transition that occurred on the strobe input pin of the I/O port.
hsltiPcDigInputPortLatchedFallingEdge	The value read from the input port corresponds to the value that was present on the input port on the falling edge of the last transition that occurred on the strobe input pin of the I/O port.

hsItiPcDigOutputPortStrobeLevels	Description
hsItiPcDigOutputPortStrobeLo	High
hsItiPcDigOutputPortStrobeHi	Low

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsItiPcDigPortParameterIndex	Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```

' Reads the input port interrupt mode
Dim IntMode As Integer
IntMode = AcqDevice1.GrabberItiPcDigPortParameter( _
    hsItiPcDigInputPortInterruptMode, _
    "PCDIG 0", _
    "Configuration0" )
' Writes a new output port strobe level
AcqDevice1.GrabberItiPcDigPortParameter _
    (hsItiPcDigOutputPortStrobeLevels) = _
    hsItiPcDigOutputPortStrobeLo

```

GrabberItiPcVisionInputPort

Description

Access to the lines of the input port of an Coreco Imaging PCVision frame grabber. This parameter is not indexed by configuration, but by a device. If the specified device is not a PCVision frame grabber, an error is fired and no internal actions are taken. Read only.

Syntax

control.**GrabberItiPcVisionInputPort**(*Device* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

This property is read only since it is not possible to change the status of an input-only port.

Example

```
Dim PortValue As Integer
PortValue = AcqDevice1.GrabberItiPcVisionInputPort("PCVision 0")
```

GrabberItiPcVisionOutputPort

Description

Access to the lines of the output port of an Coreco Imaging PC-Vision frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a PCVision frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberItiPcVisionOutputPort**(*Device* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Remark(s)

Reading this property does not read the output port of an Coreco Imaging PCVision frame grabber (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```
Dim LastOutput As Integer
LastOutput = AcqDevice1.GrabberItiPcVisionOutputPort("PCVision 0")
```

GrabberItiPcVisionParameter

Description

Indexed parameter of an Coreco Imaging PcVision frame grabber for a given configuration. If the specified device is not a PcVision frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberItiPcVisionParameter**(*Index* As hsltiIcAsyncParameterIndex , [*Device* As Variant] , [*Configuration* As Variant]) [= *Value* As Variant]

Type

Variant

Range

The range depends on the selected parameter.

hsltiPCVisionParameter	Description	Type	Range
hsltiPcVisionConfigurationFile	Configuration file name.	String	Not applicable.

hsItiPCVisionParameter	Description	Type	Range
hsItiPcVisionCameraPort	Camera port of the PC-VISION.	Integer	Minimum: 0 Maximum: 3
hsItiPcVisionExtTrigger	Specifies if an external trigger is needed.	Boolean	True, False
hsItiPcVisionTimeout	Timeout value in milliseconds (ms).	Integer	Min: 1 Maximum: Unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsItiPcVisionParameterIndex	Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current camera port
Dim Channel As Integer
Channel = AcqDevice1.GrabberItiPcVisionParameter( _
    hsItiPcVisionCameraPort, _
    "PCVision 0", _
    "Configuration0" )

' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberItiPcVisionParameter _
    (hsItiPcVisionTimeout) = 5000
```

GrabberItiPcVisionPortParameter

Description

Indexed property used to specify the parameters of the I/O ports of an Coreco Imaging PCVision frame grabber. If the specified device is not a PCVision frame grabber, an error is fired and no internal actions are taken.

Syntax

```
control.GrabberItiPcVisionPortParameter(Index As hsItiPcVisionPortParameterIndex, Device As Variant) [= Value As Variant]
```

Type

Variant

Range

The range of this property depends on the Index parameter. See tables below

<i>Index Parameter</i>	<i>Description</i>	<i>Type</i>	<i>Range</i>
hsItiPcVisionInputPortInterruptMode	Type of transition that will fire an ItiInportInterrupt event.	hsItiPcVisionInputPortInterruptModes	Enum, see table below.
hsItiPcVisionInputPortMode	Specifies if the 8-bit input port is not latched or if it is latched on a rising or on a falling edge transition.	hsItiPcVisionInputPortModes	Enum, see table below.
hsItiIcAsyncOutputPortStrobeLevel	Value for output strobe on the 8-bit port for devices that require a strobe to latch data.	hsItiPcVisionOutputPortStrobeLevels	Enum, see table below.

hsItiPcVisionInputPortInterruptMode	Description
hsItiPcVisionInputPortInterruptDisabled	Hardware interrupts on the input port do not result in ItiInportInterrupt event to be fired.
hsItiPcVisionInputPortInterruptRisingEdge	ItiInportInterrupt events are fired on the rising edge of transitions on the interrupt input of the I/O port.
hsItiPcVisionInputPortInterruptFallingEdge	ItiInportInterrupt events are fired on the falling edge of transitions on the interrupt input of the I/O port.

hsItiPcVisionInputPortMode	Description
hsItiPcVisionInputPortNotLatched	The value read from the input port is the current value present on the port.
hsItiPcVisionInputPortLatchedRisingEdge	The value read from the input port corresponds to the value that was present on the input port on the rising edge of the last transition that occurred on the strobe input pin of the I/O port.
hsItiPcVisionInputPortLatchedFallingEdge	The value read from the input port corresponds to the value that was present on the input port on the falling edge of the last transition that occurred on the strobe input pin of the I/O port.

hsItiPcVisionOutputPortStrobeLevels	Description
hsItiIcAsyncOutputPortStrobeLo	High
hsItiIcAsyncOutputPortStrobeHi	Low

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsItiPcVisionPortParameterIndex	Enum, see table above in the Range section.

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```

' Reads the input port interrupt mode
Dim IntMode As Integer
IntMode = AcqDevice1.GrabberItiPcVisionPortParameter( _
    hsItiPcVisionInputPortInterruptMode, _
    "PCVision 0", _
    "Configuration0" )
' Writes a new output port strobe level
AcqDevice1.GrabberItiPcVisionPortParameter _
    (hsItiPcVisionOutputPortStrobeLevels) = _
    hsItiPcVisionOutputPortStrobeLo

```

GrabberMake

Description

Frame grabber make of a selected device. Read only.

Syntax

control.**GrabberMake**(*Device* As Variant)

Type

hsGrabberMake

Range

The valid range for this property is as follows:

hsGrabberMake	Value	Description
hsInvalidMake	-1	Invalid selection
hsGeneric	0	Generic device
hsDataTranslation	1	Data Translation frame grabber
hsImagingTechnology	2	CorecoImaging frame grabber
hsMatrox	4	Matrox frame grabber
hsOpteon	5	Opteon controller board
hsImagenation	6	Imagenation frame grabber
hsDirectShow	7	DirectShow frame grabber
hsLMI	10	LMI Device
hsGigE	11	GigE Device
hsSony	12	Sony Intelligent Camera
hsDCAM	13	DCAM 1394 Device

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```
Dim GrabberMake As hsGrabberMake
GrabberMake = AcqDevice1.GrabberMake(2)
```

GrabberMatroxMeteor2dioUserBit**Description**

This property is used to read or write the specified User Bit, as defined in the dcf file, on a Matrox Meteor-II/Digital frame grabber. By using the User Bits, it is possible to interface with an external device. If the specified device is not a Meteor-II/Digital frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxMeteor2dioUserBit**(*Device* As Variant, *BitId* As Integer)

Type

Integer

Range

Minimum: 0
Maximum: 1

Parameter(s)

Parameter	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>BitId</i>	Bit selected for reading or writing.	hsMatroxMeteor2dioUserBit	Enum, see table below.

hsMatroxMeteor2dioUserBit	Description
hsMatroxMeteor2dioUser0	Bit User Input 0 on a read of property, bit User Output 0 on a write of property.
hsMatroxMeteor2dioUser1	Bit User Input 1 on a read of property, bit User Output 1 on a write of property.
hsMatroxMeteor2dioCameraCtrl0	Camera control bit 0.
hsMatroxMeteor2dioCameraCtrl1	Camera control bit 1.
hsMatroxMeteor2dioCameraCtrl2	Camera control bit 2.

Remark(s)

One acquisition must be performed before the **first** use of this property.

Example

```
Dim BitValue As Integer
```

```
BitValue = AcqDevice1.GrabberMatroxMeteor2dIoUserBit _
    ("METEOR2DIG 0", hsMatroxMeteor2dIoUser0)
AcqDevice1.GrabberMatroxMeteor2dIoUserBit_
    ("METEOR2DIG 0", hsMatroxMeteor2dIoUser1) = 0
```

GrabberMatroxMeteor2dParameter

Description

Indexed property used to specify the parameters of a Matrox Meteor-II/Digital frame grabber. If the specified device is not a Meteor-II/Digital frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxMeteor2dParameter**(*Index* As hsMatroxMeteor2dParameterIndex, [*Device* As Variant], [*Configuration* As Variant]) [= *NewValue* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* parameter. See tables below:

<i>Index</i> Parameter	Value	Description	Type
hsMatroxMeteor2dConfigurationFile	0	Name of the configuration file of the frame grabber.	String
hsMatroxMeteor2dGrabInputSource	1	Specifies which Data Input Source is used for grabbing images. Range: See table below.	hsMatroxMeteor2dParameterGrabInputSource
hsMatroxMeteor2dOutputSource	2	Specifies which Data Input Source is used to output the acquired image. Range: See table below.	hsMatroxMeteor2dParameterOutputSource
hsMatroxMeteor2dExtTrigger	3	Specifies if the frame grabber uses the trigger detection setting defined in the dcf file or if the trigger detection is enabled or disabled. Range: See table below.	hsMatroxMeteor2dParameterExtTrigger
hsMatroxMeteor2dDirectionX	4	Horizontal Grab Direction used for grabbing images. Range: See table below.	hsMatroxMeteor2dGrabDirection
hsMatroxMeteor2dDirectionY	5	Vertical Grab Direction used for grabbing images. Range: See table below.	hsMatroxMeteor2dGrabDirection
hsMatroxMeteor2dOffsetX	8	Sets the Window Generator X origin. Range: Must be within grab range	Integer
hsMatroxMeteor2dOffsetY	9	Sets the Window Generator Y origin. Range: Must be within grab range.	Integer
hsMatroxMeteor2dSizeX	10	Sets the Window Generator width (pixels). Range: Greater than 0 and a multiple of 32. Must be within grab range.	Integer

Index Parameter	Value	Description	Type
hsMatroxMeteor2dSizeY	11	Sets the Window Generator height (pixels) Range: Greater than 1. Must be within grab range.	Integer

hsMatroxMeteor2dParameterGrabInputSource	Description
hsMatroxMeteor2dGrabInputSource0	Data Input Source 0, D[7:0]
hsMatroxMeteor2dGrabInputSource1	Data Input Source 1, D[15:8]
hsMatroxMeteor2dGrabInputSource2	Data Input Source 2, D[23:16]
hsMatroxMeteor2dGrabInputSource3	Data Input Source 3, D[32:24]
hsMatroxMeteor2dGrabInputSourceRGB	Data Input Source RGB, D[23:0]
hsMatroxMeteor2dGrabInputSourceNone	No grab is performed. The output image will be the last grabbed image on the selected output source.

hsMatroxMeteor2dParameterOutputSource	Description
hsMatroxMeteor2dOutputSource0	Image output through the Data Input Source 0.
hsMatroxMeteor2dOutputSource1	Image output through the Data Input Source 1.
hsMatroxMeteor2dOutputSource2	Image output through the Data Input Source 2.
hsMatroxMeteor2dOutputSource3	Image output through the Data Input Source 3.

hsMatroxMeteor2dParameterExtTrigger	Description
hsMatroxMeteor2dExtTriggerDefault	The trigger setting defined in the dcf file is used.
hsMatroxMeteor2dExtTriggerEnable	The frame grabber uses the trigger detection setting defined in the dcf file and enables it.
hsMatroxMeteor2dExtTriggerDisable	Trigger detection is disabled.

hsMatroxMeteor2dGrabDirection	Description
hsMatroxMeteor2dDirectionDefault	Grab normally for specified axis.
hsMatroxMeteor2dDirectionForward	Grab normally for specified axis.
hsMatroxMeteor2dDirectionReverse	Flip the grabbed image for specified axis.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsMatroxMeteor2dParameterIndex	Enum, see table above in the Range section.
<i>Device</i>	Index number or name of the frame grabber device. Optional	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Parameter	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
Dim InputSource As Integer
InputSource = _
    AcqDevice1.GrabberMatroxMeteor2dParameter( _
        hsMatroxMeteor2dGrabInputSource, _
        "METEOR2DIG 0", _
        "Configuration0")
AcqDevice1.GrabberMatroxMeteor2dParameter( _
    hsMatroxMeteor2dExtTrigger) = _ hsMatroxMeteor2dExtTriggerDisable
```

GrabberMatroxMeteor2IoUserBit

Description

This property is used to read or write the specified User Bit on a Matrox Meteor-II/Standard frame grabber. By using the User Bits, it is possible to interface with an external device. If the specified device is not a Meteor-II/Standard frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxMeteor2IoUserBit**(*Device* As Variant, *BitId* As hsMatroxMeteor2IoUserBit) [= *NewValue* As Integer]

Type

Integer

Range

0 or 1

Parameter(s)

Parameter	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>BitId</i>	Bit selected for reading or writing	hsMatroxMeteor2IoUserBit	Enum, see table below

hsMatroxMeteor2IoUserBit	Value	Description
hsMatroxMeteor2IoUser1In	1	bit User1In
hsMatroxMeteor2IoUser2In	2	bit User2In

hsMatroxMeteor2IoUserBit	Value	Description
hsMatroxMeteor2IoUser1Out	3	bit User1Out
hsMatroxMeteor2IoUser2Out	4	bit User2Out

Remark(s)

One acquisition must be performed before the **first** use of this property.

Example

```
Dim BitValue As Integer
BitValue = _
    AcqDevice1.GrabberMatroxMeteor2IoUserBit _
        ("METEOR2 0", hsMatroxMeteor2IoUser1In)
AcqDevice1.GrabberMatroxMeteor2IoUserBit _
    ("METEOR2 0", hsMatroxMeteor2IoUser2Out) = 0
```

GrabberMatroxMeteor2mcIoUserBit

Description

This property is used to read or write the specified User Bit on a Matrox Meteor-II/Multi-Channel frame grabber. By using the User Bits, it is possible to interface with an external device. If the specified device is not a Meteor-II/Multi-Channel frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxMeteor2mcIoUserBit**(*Device* As Variant, *BitId* As Integer) [= *NewValue* As Integer]

Type

Integer

Range

0 or 1

Parameter(s)

Parameter	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>BitId</i>	Bit selected for reading or writing	hsMatroxMeteor2mcIoUserBit	Enum, see table below.

hsMatroxMeteor2mcIoUserBit	Value	Description
hsMatroxMeteor2mcIoUser1In	1	bit User1In
hsMatroxMeteor2mcIoUser2In	2	bit User2In
hsMatroxMeteor2mcIoUser1Out	3	bit User1Out
hsMatroxMeteor2mcIoUser2Out	4	bit User2Out
hsMatroxMeteor2mcIoUser1In422	5	bit User1In422
hsMatroxMeteor2mcIoUser2In422	6	bit User2In422

hsMatroxMeteor2mcIoUserBit	Value	Description
hsMatroxMeteor2mcIoUser1Out422	7	bit User1Out422
hsMatroxMeteor2mcIoUser2Out422	8	bit User2Out 422

Example

```
Dim BitValue As Integer
BitValue = _
    AcqDevice1.GrabberMatroxMeteor2mcIoUserBit _
    ("METEOR2MC 0", hsMatroxMeteor2mcIoUser1In)
AcqDevice1.GrabberMatroxMeteor2mcIoUserBit _
    ("METEOR2MC 0", hsMatroxMeteor2mcIoUser2Out) = 0
```

GrabberMatroxMeteor2Parameter

Description

Indexed property used to specify the parameters of a Matrox Meteor-II/Standard frame grabber. If the specified device is not a Meteor-II/Standard frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxMeteor2Parameter**(*Index* As hsMatroxMeteor2ParameterIndex , [*Device* As Variant] , [*Configuration* As Variant]) [= *NewValue* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* parameter. See tables below.

<i>Index</i> Parameter	Value	Description	Type
hsMatroxMeteor2ConfigurationFile	0	Name of the configuration file of the frame grabber. It can also be used to specify a keyword that informs the frame grabber to load a predefined configuration. see table below for the list of keywords.	String
hsMatroxMeteor2GrabChannel	1	Specifies which Analog Video Input channel is used for grabbing images. See table below.	Integer
hsMatroxMeteor2ExtTrigger	2	Specifies if the frame grabber uses the trigger detection setting defined in the dcf file or if the trigger detection is enabled or disabled. See table below.	Integer
hsMatroxMeteor2Timeout	3	Timeout value expressed in milliseconds.	Integer
hsMatroxMeteor2Contrast	4	Sets the contrast for the acquisition. Minimum: 0 Maximum: 255	Integer
hsMatroxMeteor2Brightness	5	Sets the brightness for the acquisition. Minimum: 0 Maximum: 255	Integer
hsMatroxMeteor2AutomaticInputGain	6	Specifies if the input gain is set automatically. Enabled=True, Disabled=False	Boolean

<i>Index</i> Parameter	Value	Description	Type
hsMatroxMeteor2InputGain	7	Sets the gain for the acquisition. Minimum: 0 Maximum: 255	Integer
hsMatroxMeteor2OffsetX	8	Sets the Window Generator X origin. Range: Must be within grab range.	Integer
hsMatroxMeteor2OffsetY	9	Sets the Window Generator Y origin. Range: Must be within grab range.	Integer
hsMatroxMeteor2SizeX	10	Sets the Window Generator width (pixels). Range: Greater than 0 and a multiple of 32. Must be within grab range.	Integer
hsMatroxMeteor2SizeY	11	Sets the Window Generator height (pixels) Range: Greater than 1 and must be within grab range.	Integer

hsMatroxMeteor2ParameterGrabChannel	Value	Description
hsMatroxMeteor2GrabChannel1	0	Analog Video Input 1
hsMatroxMeteor2GrabChannel2	1	Analog Video Input 2
hsMatroxMeteor2GrabChannel3	2	Analog Video Input 3
hsMatroxMeteor2GrabChannel4	3	Analog Video Input 4
hsMatroxMeteor2GrabChannel5	4	Analog Video Input 5
hsMatroxMeteor2GrabChannel6	5	Analog Video Input 6
hsMatroxMeteor2GrabChannel7	6	Analog Video Input 7

hsMatroxMeteor2ParameterExtTrigger	Value	Description
hsMatroxMeteor2ExtTriggerDefault	0	The trigger setting defined in the dcf file is used.
hsMatroxMeteor2ExtTriggerEnable	1	The frame grabber uses the trigger detection setting defined in the dcf file and enables it.
hsMatroxMeteor2ExtTriggerDisable	2	Trigger detection is disabled.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsMatroxMeteor2Parameter Index	Enum, see table above, in the Range section.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current grabbing channel
Dim Channel As Integer
Channel = AcqDevice1.GrabberMatroxMeteor2Parameter _
    (hsMatroxMeteor2GrabChannel, "METEOR2 0", 0)
' Sets the timeout value
AcqDevice1.GrabberMatroxMeteor2Parameter _
    (hsMatroxMeteor2Timeout, "METEOR2 0") = 500
```

GrabberMatroxMeteor2mcParameter

Description

Indexed property used to specify the parameters of a Matrox Meteor-II/Multi-Channel frame grabber. If the specified device is not a Meteor-II/Multi-Channel frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxMeteor2mcParameter**(*Index* As hsMatroxMeteor2mcParameterIndex , [*Device* As Variant] , [*Configuration* As Variant]) [= *NewValue* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* parameter. See tables below:

<i>Index</i> Parameter	Value	Description	Property Type
hsMatroxMeteor2mcConfigurationFile	0	Name of the configuration file of the frame grabber. It can also be used to specify a keyword that informs the frame grabber to load a predefined configuration. see table below for the list of keywords.	String
hsMatroxMeteor2mcGrabChannel	1	Specifies which Analog Video Input Channel is used for grabbing images.	Integer
hsMatroxMeteor2mcGrabInputSource	2	Specifies which Analog Video Input Source is used for grabbing images.	Integer
hsMatroxMeteor2mcSyncSource	3	Specifies which Analog Video Input Source is used for synchronization.	Integer
hsMatroxMeteor2mcOutputSource	4	Specifies which Analog Video Input Source is used to output the acquired image.	Integer
hsMatroxMeteor2mcExtTrigger	5	Specifies if the frame grabber uses the trigger detection setting defined in the dcf file or if the trigger detection is enabled or disabled.	Integer

Index Parameter	Value	Description	Property Type
hsMatroxMeteor2mcTimeout	6	Timeout value in milliseconds.	Integer
hsMatroxMeteor2mcOffsetX	8	Sets the Window Generator X origin. Range: Must be within grab range.	Integer
hsMatroxMeteor2mcOffsetY	9	Sets the Window Generator Y origin. Range: Must be within grab range.	Integer
hsMatroxMeteor2mcSizeX	10	Sets the Window Generator width (pixels). Range: Greater than 0 and a multiple of 32. Must be within grab range.	Integer
hsMatroxmeteor2mcSizeY	11	Sets the Window Generator height (pixels) Range: Greater than 1. Must be within grab range.	Integer

Configuration keywords	Standard Configuration
M_DEFAULT	NTSC, 640x480, 3x8 bits, 12.5MHz, composite (Color)
M_RS170	RS-170, 640x480, 8 bits, 12.5MHz, analog (Monochrome)
M_CCIR	CCIR, 768x576, 8bits, 14.8MHz, analog (Monochrome)
M_NTSC	NTSC, 640x480, 3x8 bits, 12.5MHz, composite (Color)
M_NTSC_YC	RS-170 Y/C (SVHS), 640x480, 3x8 bits, 12.5MHz (Color)
M_PAL	PAL I, 768x576, 3x8 bits, 14.8MHz, composite (Color)
M_PAL_YC	PAL Y/C, 768x576, 3x8 bits, 14.8MHz (Color)

hsMatroxMeteor2mcParameterGrabChannel	Value	Description
hsMatroxMeteor2mcGrabChannel1	0	Analog Video Input Channel 1
hsMatroxMeteor2mcGrabChannel1	1	Analog Video Input Channel 2

hsMatroxMeteor2mcParameterGrabInputSource	Value	Description
hsMatroxMeteor2mcGrabInputSource1	0	Analog Video Input Source 1 (Red).
hsMatroxMeteor2mcGrabInputSource2	1	Analog Video Input Source 2 (Green).
hsMatroxMeteor2mcGrabInputSource3	2	Analog Video Input Source 3 (Blue).
hsMatroxMeteor2mcGrabInputSourceRGB	3	Analog Video Input Source 1, 2 and 3 are grabbed simultaneously.
hsMatroxMeteor2mcGrabInputSourceNone	4	No grab is performed. The output image will be the last grabbed image on the selected output source.

hsMatroxMeteor2mcParameterSyncSource	Value	Description
hsMatroxMeteor2mcSyncSource1	0	Synchronization from the Analog Video Input Source 1

hsMatroxMeteor2mcParameterSyncSource	Value	Description
hsMatroxMeteor2mcSyncSource2	1	Synchronization from the Analog Video Input Source 2
hsMatroxMeteor2mcSyncSource3	2	Synchronization from the Analog Video Input Source 3
hsMatroxMeteor2mcSyncSourceSYNC	3	Synchronization from the Analog Video Input Source SYNC

hsMatroxMeteor2mcParameterOutputSource	Value	Description
hsMatroxMeteor2mcOutputSource1	0	Image output through the Analog Video Input Source 1
hsMatroxMeteor2mcOutputSource2	1	Image output through the Analog Video Input Source 2
hsMatroxMeteor2mcOutputSource3	2	Image output through the Analog Video Input Source 3

hsMatroxMeteor2mcParameterExtTrigger	Value	Description
hsMatroxMeteor2mcExtTriggerDefault	0	The trigger setting defined in the dcf file will be used.
hsMatroxMeteor2mcExtTriggerEnable	1	The frame grabber uses the trigger detection setting defined in the dcf file and enables it.
hsMatroxMeteor2mcExtTriggerDisable	2	Trigger detection is disabled.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsMatroxMeteor2mcParameterIndex	Enum, see table above in the Range section.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current grabbing channel
Dim Channel As Integer
Channel = AcqDevice1.GrabberMatroxMeteor2mcParameter _
    (hsMatroxMeteor2mcGrabChannel, "Meteor2MC 0",0)
' Sets the timeout value
```

```
AcqDevice1.GrabberMatroxMeteor2mcParameter _
    (hsMatroxMeteor2mcTimeout, "METEOR2MC 0") = 500
```

GrabberMatroxOrionIoUserBit

Description

This property is used to read or write the specified User Bit, as defined in the dcf file, on a Matrox Orion frame grabber. By using the User Bits, it is possible to interface with an external device. If the specified device is not a Orion frame grabber, an error is fired and no internal actions are taken.

Syntax

```
control.GrabberMatroxOrionIoUserBit(Device As Variant, BitId As
hsMatroxMeteor2IoUserBit) [= NewValue As Integer])
```

Type

Integer

Range

0 or 1

Parameter(s)

Parameter	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>BitId</i>	Bit selected for reading or writing.	hsMatroxOrionIoUserBit	Enum, see table below.

hsMatroxOrionIoUserBit	Value	Description
hsMatroxOrionIoUser1In	1	bit User1In
hsMatroxOrionIoUser2In	2	bit User2In
hsMatroxOrionIoUser1Out	3	bit User1Out
hsMatroxOrionIoUser2Out	4	bit User2Out

Remark(s)

One acquisition must be performed before the **first** use of this property.

Example

```
Dim BitValue As Integer
BitValue = _
    AcqDevice1.GrabberMatroxOrionIoUserBit _
        ("ORION 0", hsMatroxOrionIoUser1In)
AcqDevice1.GrabberMatroxOrionIoUserBit _
    ("ORION 0", hsMatroxOrionIoUser2Out) = 0
```

GrabberMatroxOrionParameter

Description

Indexed property used to specify the parameters of a Matrox Orion frame grabber. If the specified device is not a Orion frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberMatroxOrionParameter**(*Index* As hsMatroxOrionParameterIndex , [*Device* As Variant] , [*Configuration* As Variant]) [= *NewValue* As Variant]

Type

Variant

Range

The range of this property depends on the *Index* parameter. See tables below:

<i>Index</i> Parameter	Value	Description	Property type
hsMatroxOrionConfigurationFile	0	Name of the configuration file of the frame grabber. It can also be used to specify a keyword that informs the frame grabber to load a predefined configuration. see table below for the list of keywords.	String
hsMatroxOrionGrabChannel	1	Specifies which Analog Video Input channel is used for grabbing images. See table below.	Integer
hsMatroxOrionExtTrigger	2	Specifies if the frame grabber uses the trigger detection setting defined in the dcf file or if the trigger detection is enabled or disabled. See table below.	Integer
hsMatroxOrionTimeout	3	Timeout value expressed in milliseconds.	Integer
hsMatroxOrionContrast	4	Sets the contrast for the acquisition. Minimum: 0 Maximum: 255	Integer
hsMatroxOrionBrightness	5	Sets the brightness for the acquisition. Minimum: 0 Maximum: 255	Integer
hsMatroxOrionAutomaticInputGain	6	Specifies if the input gain is set automatically. Enabled=True, Disabled=False	Boolean
hsMatroxOrionInputGain	7	Sets the gain for the acquisition. Minimum: 0 Maximum: 255	Integer
hsMatroxOrionOffsetX	8	Sets the Window Generator X origin. Range: Must be within grab range.	Integer
hsMatroxOrionOffsetY	9	Sets the Window Generator Y origin. Range: Must be within grab range.	Integer
hsMatroxOrionSizeX	10	Sets the Window Generator width (pixels). Range: Greater than 0 and a multiple of 32. Must be within grab range.	Integer
hsMatroxOrionSizeY	11	Sets the Window Generator height (pixels) Range: Greater than 1 and must be within grab range.	Integer

Configuration keywords	Standard Configuration
M_DEFAULT	NTSC, 640x480, 3x8 bits, 12.5MHz, composite (Color)
M_RS170	RS-170, 640x480, 8 bits, 12.5MHz, analog (Monochrome)
M_CCIR	CCIR, 768x576, 8bits, 14.8MHz, analog (Monochrome)
M_NTSC	NTSC, 640x480, 3x8 bits, 12.5MHz, composite (Color)
M_NTSC_YC	RS-170 Y/C (SVHS), 640x480, 3x8 bits, 12.5MHz (Color)
M_PAL	PAL I, 768x576, 3x8 bits, 14.8MHz, composite (Color)
M_PAL_YC	PAL Y/C, 768x576, 3x8 bits, 14.8MHz (Color)

hsMatroxOrionParameterGrabChannel	Value	Description
hsMatroxOrionGrabChannel1	0	Analog Video Input 1
hsMatroxOrionGrabChannel2	1	Analog Video Input 2
hsMatroxOrionGrabChannel3	2	Analog Video Input 3
hsMatroxOrionGrabChannel4	3	Analog Video Input 4
hsMatroxOrionGrabChannel5	4	Analog Video Input 5
hsMatroxOrionGrabChannel6	5	Analog Video Input 6
hsMatroxOrionGrabChannel7	6	Analog Video Input 7
hsMatroxOrionGrabChannel8	7	Analog Video Input 8

hsMatroxOrionParameterExtTrigger	Value	Description
hsMatroxOrionExtTriggerDefault	0	The trigger setting defined in the dcf file is used.
hsMatroxOrionExtTriggerEnable	1	The frame grabber uses the trigger detection setting defined in the dcf file and enables it.
hsMatroxOrionExtTriggerDisable	2	Trigger detection is disabled.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsMatroxOrionParameter Index	Enum, see table above, in the Range section.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current grabbing channel
Dim Channel As Integer
Channel = AcqDevice1.GrabberMatroxOrionParameter _
    (hsMatroxOrionGrabChannel, "Orion 0", 0)

' Sets the timeout value
AcqDevice1.GrabberMatroxOrionParameter _
    (hsMatroxOrionTimeout, "Orion 0") = 500
```

GrabberModel

Description

Frame grabber model. Read only.

Syntax

control.**GrabberModel**(*Device* As variant)

Type

hsGrabberModel

Range

The valid range for this property is as follows:

hsGrabberModel	Value	Description
hsInvalidModel	-1	Invalid selection. Device is not a frame grabber.
hsGeneric	0	Generic frame grabber.
hsDT3152	1	Data Translation DT3152 frame grabber.
hsDT3155	2	Data Translation DT3155 frame grabber.
hsDT3157	3	Data Translation DT3157 frame grabber.
hsICASYNC	4	Coreco Imaging IC-ASYNC frame grabber.
hsPCVISION	6	Coreco Imaging PCVision frame grabber
hsMETEOR2	7	Matrox Meteor-II/Standard frame grabber
hsMETEOR2MC	8	Matrox Meteor-II/Multi-Channel frame grabber
hsMETEOR2DIG	9	Matrox Meteor-II/Digital frame grabber
hsDEPICT4X1	10	Opteon DepictMaster 4x1 controller board
hsPX610A	11	Imagenation PXC610A frame grabber.
hsPXC200	12	Imagenation PXC200 frame grabber.
hsPCDIG	13	Coreco Imaging PC-DIG frame grabber.
hsORION	14	Matrox Orion frame grabber
hsDIRECTSHOWDEVICE	15	DirectShow device
hsDEPICTUSB	16	Depict USB camera
hsDEPICTETHERNET	17	Depict ethernet camera
hsDEPICTSSERIE	18	Depict S-Series device
hsMETEOR2CL	19	Matrox Meteor II/Camera Link frame grabber
hsGocator	22	LMI Gocator device

hsGrabberModel	Value	Description
hsGIGE	23	GigE Vision Camera
hsICam	24	Sony Intelligent Camera
hsDCAM1394	25	DCAM 1394 Camera

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```
Dim GrabberModel As hsGrabberModel
GrabberModel = AcqDevice1.GrabberModel(2)
```

GrabberGigEParameter

Description

Indexed parameter of a device for a given configuration.

Syntax

```
control.GrabberGigEParameter(Index As Integer, [Device As Variant]  
, [Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The range depends on the selected parameter. See **GrabberGigEParameterIndex** for additional parameters.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterCategory

Description

Indexed parameter of a device for a given configuration, to retrieve the category of a parameter.

Syntax

```
control.GetGrabberGigEParameterCategory(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

String as Variant

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to

the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterFlags

Description

Indexed parameter of a device for a given configuration, to retrieve the flags of a parameter.

Syntax

```
control.GetGrabberGigEParameterFlags(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

Integer

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterImpact

Description

Indexed parameter of a device that gives the impacted parameters when changed.

Syntax

```
control.GetGrabberGigEParameterImpact(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

String as Variant

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterIndex

Description

This property is used to retrieve the index of a given parameter's name for a device.

Syntax

```
control.GrabberGigEParameterIndex(Name As Variant, [Device As Variant], [Configuration  
As Variant])
```

Type

Integer

Range

0 or more
-1 if not found

Parameter(s)

The GigE Vision standard enforces many of these *Names*, but some of them vary according to the manufacturer/firmware of the device. HexSight adds the following *Names* to the attributes list for extended functionalities.

Name	Description	Type	Range
<i>HSAcquisitionTimeout</i>	Specifies the time (ms) before the acquisition is aborted.	long	Minimum: 0 Maximum: 1000000 Default: 5000
<i>HSContinuousExecution</i>	If True, the device will continuously stream data. Only the last image (triggered by Start or Execute) will be acquired. Notes: Increases throughput of images by reducing the latency between acquisitions, while keeping CPU usage to a minimum. Also works with triggering.	boolean	True, False
<i>HSStopDelay</i>	If the device is in streaming mode instead of single frame mode, it specifies the time (ms) before the acquisition is stopped. Notes: Adjust according to required throughput; most GigE Vision devices require a non-zero value. Not applicable if <i>HSContinuousExecution</i> is set to True.	long	Minimum: 0 Maximum: 1000000 Default: 200

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterMax

Description

Indexed parameter of a device to retrieve the maximum value of a setting.

Syntax

```
control.GetGrabberGigEParameterMax(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

Variant

Range

The range depends on the selected parameter.

Remark(s)

In case of an enumerated property, the function will return the list of possible values.
This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterMin

Description

Indexed parameter of a device to retrieve the minimum value of a setting.

Syntax

control.**GetGrabberGigEParameterMin**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

Variant

Range

The range depends on the selected parameter.

Remark(s)

In case of an enumerated property, the function will return the list of possible values. This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterName

Description

Indexed property to retrieve the name of the parameter of a device.

Syntax

control.**GrabberGigEParameterName**(*Index* As Integer, [*Device* As Variant]
, [*Configuration* As Variant])

Type

String as Variant

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParametersCount

Description

This property is used to get the number of available parameters for a device.

Syntax

control.**ParametersCount**([*Device* As Variant], [*Configuration* As Variant])

Type

Integer

Range

0 or more.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberGigEParameterType

Description

Indexed parameter of a device for a given configuration, to retrieve the type of a parameter.

Syntax

```
control.GetGrabberGigEParameterType(Index As Integer, [Device As Variant]  
, [Configuration As Variant])
```

Type

Integer

Range

The range depends on the device.

Remark(s)

This property applies to the default configuration if the Configuration parameter is not provided. See the ConfigurationDefault property for more details. This property applies to the configuration's currently selected input device if the Device parameter is not provided.

GrabberPx610aParameter

Description

Indexed parameter of an Imagenation PX610A frame grabber for a given configuration.

Syntax

```
control.GrabberPx610aParameter(Index As hsPx610aParameterIndex , [Device As Variant]  
, [Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The range depends on the selected parameter.

hsPx610aParameterIndex	Description	Type	Range
hsPx610aCameraInput	Camera selection among the four multiplexed inputs.	Integer	Minimum: 0 Maximum: 3
hsPx610aTimeout	Timeout value in millisecond.	Integer	Minimum: 0 Maximum: Unlimited
hsPx610aTriggerType	Sets trigger type for acquisitions.	hsPx610aTriggerTypes	Enum, see table below.
hsPx610aTriggerDebounce	When activated, new triggers are locked out until trigger input has returned to untriggered state during at least one vertical blank.	Boolean	True, False
hsPx610aTriggerDelay	Delay applied between the detection of a trigger and the beginning of the acquisition (Number of video fields to skip).	Integer	Minimum: 0 Maximum: Unlimited
hsPx610aSyncMode	Synchronization mode.	hsPx610aSync Modes	Enum, see table below.

hsPx610aParameterIndex	Description	Type	Range
hsPx610aHorizontalResolution	Sets the number of pixels to be digitized during each horizontal period.	hsPx610aHorizontalResolutions	Enum, see table below.
hsPx610aSingleField	When activated, the board treats every field of video received As if it were field 0. Typically used in non-interlaced video.	Boolean	True, False
hsPx610aFieldLength	Sets number of active lines in a single video field.	Integer	Min: 1 Maximum: 32767
hsPx610aBlankLength	Sets number lines in the vertical blank period of the video signal.	Integer	User or Internal sync: Min: 14, Maximum: 256 WEN sync: Min: 1, Maximum: 256
hsPx610aScaleX	Sets the horizontal subsampling factor. A ScaleX setting of 1 will sample all the pixels in the incoming video line. A value of 2 will sample 1 pixel for every 2 pixels in the video line, and so forth.	Integer	Min: 1 Maximum: 17
hsPx610aScaleY	Sets the vertical subsampling factor. A ScaleY setting of 1 will sample all the lines in the incoming video line. A value of 2 will sample 1 line for every 2 lines in the video signal, and so forth.	Integer	Min: 1 Maximum: 17
hsPx610aX0	X0 defines the first column that will be digitized in the subsampled video line.	Integer	Minimum: 0 Maximum: (hsPx610aHorizontalResolution/ hsPx610aScaleX) - 1
hsPx610aDX	DX defines the number of columns that will be digitized in the subsampled video line, starting at column X0.	Integer	Min: 4 Maximum: (hsPx610aHorizontalResolution/ hsPx610aScaleX) - hsPx610aX0
hsPx610aY0	Y0 defines the first line that will be digitized in the subsampled video signal.	Integer	Minimum: 0 Maximum: min[1023, (hsPx610aFieldLength)-1]

hsPx610aParameterIndex	Description	Type	Range
hsPx610aDY	DY defines the number of lines that will be digitized in the subsampled video signal, starting at line Y0.	Integer	Min: 1 Maximum: min[1024, (hsPx610aFieldLength)-1]
hsPx610aHorizontalSyncPolarity	Sets polarity of the horizontal synchronization signal.	hsPx610aSyncLinePolarity	Enum, see table below.
hsPx610aVerticalSyncPolarity	Sets polarity of the vertical synchronization signal.	hsPx610aSyncLinePolarity	Enum, see table below.
hsPx610aSyncEnabled	Enables/Disables synchronization signals.	Boolean	True, False
hsPx610aVideoOffset	Offset applied to the digitized signal.	Integer	Min: -128 Maximum: +127
hsPx610aVideoGain	Gain applied to the digitized signal.	Single	Minimum: 0.500 Maximum: 8.000
hsPx610aStrobeMode	Sets mode of action for strobe lines.	hsPx610StrobeModes	Enum, see table below.
hsPx610aStrobe0Polarity	Sets polarity for strobe line 0.	hsPx610aStrobePolarity	Enum, see table below.
hsPx610aStrobe1Polarity	Sets polarity for strobe line 1.	hsPx610aStrobePolarity	Enum, see table below.
hsPx610aStrobe0Length	Sets the length of the pulse generated on strobe line 0 (Number of video lines).	Integer	Min: 1 Maximum: 65535
hsPx610aStrobe1Length	Sets the length of the pulse generated on strobe line 1 (Number of video lines).	Integer	Min: 1 Maximum: 65535
hsPx610aStrobeGapLength	Sets the length of the gap between the two strobe pulses (Number of video lines).	Integer	Min: 1 Maximum: 65535

hsPx610aTriggerTypes	Description
hsPx610aTriggerDisabled	No use of triggers.
hsPx610aTriggerHighLevel	Triggering occurs when the input signal is HIGH.
hsPx610aTriggerLowLevel	Triggering occurs when the input signal is LOW.
hsPx610aTriggerRisingEdge	Triggering occurs when a low-to-high transition is detected on the input signal.
hsPx610aTriggerFallingEdge	Triggering occurs when a high-to-low transition is detected on the input signal.

hsPx610aSyncModes	Description
hsPx610aSyncModeAutomatic	The PX610A automatically detects the type of signal (NSTC or CCIR/PAL) and synchronizes to the incoming video.
hsPx610aSyncModeInternal	The board generates its own video timing and ignores sync information in the incoming video signal.
hsPx610aSyncModeUser	The video timing is entirely specified by the user.
hsPx610aSyncModeWEN	The board synchronizes vertically to the WEN signal received on the trigger input. The board still uses the horizontal sync in the incoming video signal.

hsPx610aHorizontalResolutions	Description
hsPx610aHorizontalResolutionRS170	RS-170 sets the number of pixels per video line to 640.
hsPx610aHorizontalResolutionCCIR	CCIR sets the number of pixels per video line to 768.

hsPx610aSyncLinePolarity	Description
hsPx610aSyncLineActiveHigh	Active high synchronization signal.
hsPx610aSyncLineActiveLow	Active low synchronization signal.

hsPx610aStrobeModes	Description
hsPx610aStrobeModeDisabled	No strobing.
hsPx610aStrobeModeNormal	The GrabberPx610aFireStrobe method is used to fire strobes.
hsPx610aStrobeModeTriggered	A complete strobe sequence (strobe 0, gap, strobe 1) is initiated each time a trigger is detected.

hsPx610aStrobePolarity	Description
hsPx610aPolarityActiveHigh	Active high strobe signal.
hsPx610aPolarityActiveLow	Active low strobe signal.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsPx610aParameter Index	Enum, see table below.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger type
Dim Type As hsPx610aTriggerType
Type = AcqDevice1.GrabberPx610aParameter _
    (hsPx610aTriggerType,"PX610A 0")
' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberPx610aParameter(hsPx610aTimeout) = 5000
```

GrabberPxc200InputPort

Description

Access to the lines of the input port of an Imagenation PXC200 frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a PXC200 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberPxc200InputPort**(Device As Variant).

Type

Integer

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Range

Minimum: 0x0
Maximum: 0xF

Remark(s)

This property is read only As it is not possible to change the status of an input-only port.

Example

```
Dim PortValue As Integer
PortValue = AcqDevice1.GrabberPxc200InputPort("PXC200 0")
```

GrabberPxc200OutputPort

Description

Access to the lines of the output port of an Imagenation PXC200 frame grabber. This parameter is not indexed by a configuration, but by a device. If the specified device is not a PXC200 frame grabber, an error is fired and no internal actions are taken.

Syntax

control.**GrabberPxc200OutputPort**(Device As Variant).

Type

Integer

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Range

Minimum: 0x0

Maximum: 0xF

Remark(s)

Reading this property does not read the output port of an Imagenation PXC200 frame grabber (it is impossible to read an output-only port), but it returns the last value that was written to the port.

Example

```
AcqDevice1.GrabberPxc200OutputPort("PXC200 0") = 4
```

```
Dim LastOutput As Integer
```

```
LastOutput = AcqDevice1.GrabberPxc200OutputPort("PXC200 0")
```

GrabberPxc200Parameter

Description

Indexed parameter of an Imagenation PXC200 frame grabber for a given configuration.

Syntax

```
control.GrabberPxc200Parameter(Index As hsPxc200ParameterIndex , [Device As Variant] ,  
[Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The range depends on the selected parameter.

<i>Index</i> Parameter	Description	Type	Range
hsPxc200CameraInput	Camera selection among the four multiplexed inputs.	Integer	Minimum: 0 Maximum: 3
hsPxc200Timeout	Timeout value in milliseconds.	Integer	Minimum: 0 Maximum: Unlimited
hsPxc200TriggerUse	Configures how the acquisition request will be triggered.	hsPxc200TriggerUses	Enum, see table below.
hsPxc200Trigger0State	State of the trigger input 0 required to start an acquisition.	hsPxc200TriggerStates	Enum, see table below.
hsPxc200Trigger1State	State of the trigger input 1 required to start an acquisition.	hsPxc200TriggerStates	Enum, see table below.
hsPxc200Trigger2State	State of the trigger input 2 required to start an acquisition.	hsPxc200TriggerStates	Enum, see table below.

Index Parameter	Description	Type	Range
hsPxc200Trigger3State	State of the trigger input 3 required to start an acquisition.	hsPxc200TriggerStates	Enum, see table below.
hsPxc200VideoType	Sets the type of the video signal.	Pxc200VideoTypes	Enum, see table below.
hsPxc200Contrast	Sets the monochrome gain of the video digitizing circuit.	Single	Minimum: 0.0 Maximum: 2.0 1.0: doesn't modify signal.
hsPxc200Brightness	Sets the monochrome offset of the video digitizing circuit.	Single	Min: -0.5 Maximum: -0.5 0.0: doesn't modify signal.
hsPxc200VideoLevel	Sets the voltage difference between the bottom of video sync and bright white.	Single	Minimum: 0.0 Maximum: 2.5 Default: 1.3
hsPxc200LowPassFilter	Low-pass filter that reduces high-frequency information in the video signal. For optimum accuracy, should be <i>disabled</i> .	hsPxc200LowPassFilterValues	Enum, see table below.
hsPxc200CoreFunction	Causes all video below a specified level to be digitized to black. Can improve the apparent contrast of some dark images. For optimum accuracy, should be <i>disabled</i> .	hsPxc200CoreFunctionValues	Enum, see table below.
hsPxc200PeakFilter	Amplifies high-frequency information in the video signal. Can sharpen edges in a blurry image. For optimum accuracy, should be <i>disabled</i> .	hsPxc200PeakFilterValues	Enum, see table below.
hsPxc200GammaCorrection	Activates a gamma correction filter: NTSC: 2.2 PAL: 2.8 For optimum accuracy, should be <i>disabled</i> .	Boolean	True: enabled False: disabled
hsPxc200LumaComb Filter	Comb filter that reduces artifacts in the monochrome signal caused by crosstalk from the color signal. For optimum accuracy, should be <i>disabled</i> .	Boolean	True: enabled False: disabled
hsPxc200SVideo	Video signal is an S-Video signal, with separate color and monochrome channels.	Boolean	True: S-Video False: disabled

Index Parameter	Description	Type	Range
hsPxc200NotchFilter	This filter removes the color burst signal from the video signal before it is digitized. For optimum accuracy, should be <i>disabled</i> .	Boolean	True: enabled False: disabled
hsPxc200AutoGain Control	Activates automatic gain control (AGC) for color saturation to compensate for non-standard color signals.	Boolean	True: enabled False: disabled
hsPxc200BwDetect	Sets the color signal to zero when the board detects a missing or weak color burst signal.	Boolean	True: enabled False: disabled
hsPxc200ChromaCombFilter	This comb filter reduces color artifacts. For optimum accuracy, should be <i>disabled</i> .	Boolean	True: enabled False: disabled
hsPxc200InputLine0 Event	The event which will cause the input line 0 state to be set.	hsPx200InputLineEvents	Enum, see table below.
hsPxc200InputLine1 Event	The event which will cause the input line 1 state to be set.	hsPx200InputLineEvents	Enum, see table below.
hsPxc200InputLine2 Event	The event which will cause the input line 2 state to be set.	hsPx200InputLineEvents	Enum, see table below.
hsPxc200InputLine3 Event	The event which will cause the input line 3 state to be set.	hsPx200InputLineEvents	Enum, see table below.
hsPxc200InputLine0Debounce Long	<i>Enabled</i> : a delay of at least two vertical blanks is required before another transition can be detected on Line 0. <i>Disabled</i> : the minimum delay is one vertical blank. No effect when trigger Event for input is Level .	Boolean	True: enabled False: disabled
hsPxc200InputLine1Debounce Long	<i>Enabled</i> : a delay of at least two vertical blanks is required before another transition can be detected on Line 1. <i>Disabled</i> : the minimum delay is one vertical blank. No effect when trigger Event for input is Level .	Boolean	True: enabled False: disabled
hsPxc200InputLine2Debounce Long	<i>Enabled</i> : a delay of at least two vertical blanks is required before another transition can be detected on Line 2. <i>Disabled</i> : the minimum delay is one vertical blank. No effect when trigger Event for input is Level .	Boolean	True: enabled False: disabled

Index Parameter	Description	Type	Range
hsPxc200InputLine3Debounce Long	<i>Enabled:</i> a delay of at least two vertical blanks is required before another transition can be detected on Line 3. <i>Disabled:</i> the minimum delay is one vertical blank. No effect when trigger Event for input is Level .	Boolean	True: enabled False: disabled
hsPxc200InputLine0Debounce Both	<i>Enabled:</i> delay is applied to both the latched low-to-high or high-to-low transition and the returning transition on Line 0. <i>Disabled:</i> only the latched low-to-high or high-to-low transition is debounced, not the returning transition.	Boolean	True: enabled False: disabled
hsPxc200InputLine1Debounce Both	<i>Enabled:</i> delay is applied to both the latched low-to-high or high-to-low transition and the returning transition on Line 1. <i>Disabled:</i> only the latched low-to-high or high-to-low transition is debounced, not the returning transition.	Boolean	True: enabled False: disabled
hsPxc200InputLine2Debounce Both	<i>Enabled:</i> delay is applied to both the latched low-to-high or high-to-low transition and the returning transition on Line 2. <i>Disabled:</i> only the latched low-to-high or high-to-low transition is debounced, not the returning transition.	Boolean	True: enabled False: disabled
hsPxc200InputLine3Debounce Both	<i>Enabled:</i> delay is applied to both the latched low-to-high or high-to-low transition and the returning transition on Line 2. <i>Disabled:</i> only the latched low-to-high or high-to-low transition is debounced, not the returning transition.	Boolean	True: enabled False: disabled
hsPxc200OutputLine4Strobe Type	Sets the strobe type, if any, for output line 4.	hsPxc200OutputLineStrobeTypes	Enum, see table below.
hsPxc200OutputLine5Strobe Type	Sets the strobe type, if any, for output line 5.	hsPxc200OutputLineStrobeTypes	Enum, see table below.
hsPxc200OutputLine6Strobe Type	Sets the strobe type, if any, for output line 6.	hsPxc200OutputLineStrobeTypes	Enum, see table below.
hsPxc200OutputLine7Strobe Type	Sets the strobe type, if any, for output line 7.	hsPxc200OutputLineStrobeTypes	Enum, see table below.

Index Parameter	Description	Type	Range
hsPxc200OutputLine4StrobeStartLine	Starting point (line number) in the video field for the strobe pulse on output line 4.	Integer	Min: 1 Maximum: 512
hsPxc200OutputLine5StrobeStartLine	Starting point (line number) in the video field for the strobe pulse on output line 5.	Integer	Min: 1 Maximum: 512
hsPxc200OutputLine6StrobeStartLine	Starting point (line number) in the video field for the strobe pulse on output line 6.	Integer	Min: 1 Maximum: 512
hsPxc200OutputLine7StrobeStartLine	Starting point (line number) in the video field for the strobe pulse on output line 7.	Integer	Min: 1 Maximum: 512
hsPxc200OutputLine4StrobePeriod	Period of the strobe pulse on output line 4 (in ms).	Single	Min: 64 us Maximum: 4.2 s
hsPxc200OutputLine5StrobePeriod	Period of the strobe pulse on output line 5 (in ms).	Single	Min: 64 us Maximum: 4.2 s
hsPxc200OutputLine6StrobePeriod	Period of the strobe pulse on output line 6 (in ms).	Single	Min: 64 us Maximum: 4.2 s
hsPxc200OutputLine7StrobePeriod	Period of the strobe pulse on output line 7 (in ms).	Single	Min: 64 us Maximum: 4.2 s
hsPxc200OutputLinesTrigger	Specifies which input line is used as the trigger source when triggered strobes are used.	hsPxc200OutputLinesTriggers	Enum, see table below.
hsPxc200OutputLinesHoldoffStart0	Defines the starting point (line number) of holdoff region in field 0 where no strobes will be fired.	Integer	Min: 1 Maximum: Image field height
hsPxc200OutputLinesHoldoffStart1	Defines the starting point (line number) of holdoff region in field 1 where no strobes will be fired.	Integer	Min: 1 Maximum: Image field height
hsPxc200OutputLinesHoldoffWidth0	Defines the width (line number) of the holdoff region in field 0 where no strobes will be fired.	Integer	Min: 1 Maximum: Image field height
hsPxc200OutputLinesHoldoffWidth1	Defines the width (line number) of the holdoff region in field 1 where no strobes will be fired.	Integer	Min: 1 Maximum: Image field height
hsPxc200OutputLine4HoldoffEnabled	Specifies if the holdoff period is applied to the strobe on output line 4.	Boolean	True, False
hsPxc200OutputLine5HoldoffEnabled	Specifies if the holdoff period is applied to the strobe on output line 5.	Boolean	True, False
hsPxc200OutputLine6HoldoffEnabled	Specifies if the holdoff period is applied to the strobe on output line 6.	Boolean	True, False

Index Parameter	Description	Type	Range
hsPxc200OutputLine7HoldoffEnable	Specifies if the holdoff period is applied to the strobe on output line 7.	Boolean	True, False
hsPxc200OutputLinesGap	When set to a positive value (in ms) enables double strobing on line 7 and sets the time between the two strobe pulses. Double strobing on line 7 is disabled otherwise.	Single	Min: -1 Maximum: Unlimited

hsPxc200TriggerUses	Description
hsPxc200TriggerUseAll	Acquisition is triggered if <u>all</u> of the conditions configured for Trigger 0, 1, 2 or 3 are met.
hsPxc200TriggerUseAny	Acquisition is triggered if <u>any</u> of the conditions configured for Trigger 0, 1, 2 or 3 are met.
hsPxc200TriggerUseNone	Does not synchronize the acquisition request to a trigger event

hsPxc200TriggerStates	Description
hsPxc200TriggerStateSet	Has to be set to be considered valid in the global trigger condition.
hsPxc200TriggerStateNotSet	Must not be set to be considered valid in the global trigger condition.
hsPxc200TriggerStateIgnore	Will be ignored in the global trigger condition.

hsPxc200VideoTypes	Description
hsPxc200VideoDetect	The PXC200 automatically detects the format of the incoming video signal.
hsPxc200VideoTypeNtsc	NTSC standard
hsPxc200VideoTypeNtscJ	NTSC with no pedestal voltage
hsPxc200VideoTypePal	PAL B, D, G, H or standards I
hsPxc200VideoTypePalM	PAL M standard
hsPxc200VideoTypePalN	PAL N standard
hsPxc200VideoTypeSecam	SECAM standard

hsPxc200LowPassFilterValues	Description
hsPxc200LowPassFilterDisabled	No filtering.
hsPxc200LowPassFilterAuto	Automatically sets the filtering based on the resolution.
hsPxc200LowPassFilter1	Sets low level of filtering.
hsPxc200LowPassFilter2	Sets medium level of filtering.
hsPxc200LowPassFilter3	Sets high level of filtering.

hsPxc200CoreFunctionValues	Description
hsPxc200CoreFunctionDisabled	Disables the coring function.
hsPxc200CoreFunction8	Will digitize As black all pixels with grey-level values below 8.
hsPxc200CoreFunction16	Will digitize As black all pixels with grey-level values below 16.
hsPxc200CoreFunction32	Will digitize As black all pixels with grey-level values below 32.

hsPxc200PeakFilterValues	Description
hsPxc200PeakFilterDisabled	Disables the peak filter
hsPxc200PeakFilter0	Highest gain filtering.
hsPxc200PeakFilter1	High gain filtering.
hsPxc200PeakFilter2	Low gain filtering.
hsPxc200PeakFilter3	Lowest gain filtering.

hsPxc200InputLineEvents	Description
hsPxc200InputLineLevel	Will set the trigger condition if a high TTL level is detected on the input.
hsPxc200InputLineLowToHigh	Will set the trigger condition if a low-to-high transition is detected on the input.
hsPxc200InputLineHighToLow	Will set the trigger condition if a high-to-low transition is detected on the input.

hsPxc200OutputLineStrobeTypes	Description
hsPxc200OutputLineStrobeDisabled	No strobe.
hsPxc200OutputLineStrobeSyncField0	A strobe pulse will be generated each time the line specified by Start Line is encountered in the video field 0.
hsPxc200OutputLineStrobeSyncField1	A strobe pulse will be generated each time the line specified by Start Line is encountered in the video field 1.
hsPxc200OutputLineStrobeSyncEither	A strobe pulse will be generated each time the line specified by Start Line is encountered in the video fields 0 and 1.
hsPxc200OutputLineStrobeTrigger	A strobe pulse will be generated each time a trigger condition is detected on the selected input line.

hsPxc200OutputLinesTriggers	Description
hsPxc300OutputLinesTriggerLine0	Input line 0 will be used to trigger the strobe on the triggered output lines.
hsPxc300OutputLinesTriggerLine1	Input line 1 will be used to trigger the strobe on the triggered output lines.
hsPxc300OutputLinesTriggerLine2	Input line 2 will be used to trigger the strobe on the triggered output lines.
hsPxc300OutputLinesTriggerLine3	Input line 3 will be used to trigger the strobe on the triggered output lines.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsPxc200ParameterIndex	Enum, see table above.
<i>Device</i>	Index number or name of the frame grabber device. Optional	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger use
Dim Use As hsPxc200TriggerUses
Use = AcqDevice1.GrabberPxc200Parameter _
    (hsPxc200TriggerUse, "PXC200 0")
' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberPxc200Parameter(hsPxc200Timeout) = 5000
```

GrabberSonylCamParameter**Description**

Indexed parameter of a Sony Intelligent Camera for a given configuration. Parameters marked with [RO] indicates that they are read-only.

Syntax

```
control.GrabberSonylCamParameter(Index As hsSonylCamParameterIndex , [Device As Variant] , [Configuration As Variant]) [= Value As Variant]
```

Type

Variant

Range

The range depends on the selected parameter.

hsSonylCamParameterIndex	Description	Type	Range
hsSonylCamAutoGain	0: off, 1: on	short	0/1, default: 0
hsSonylCamAutoGainCorrection	Auto gain correction value.	short	(-8)-8, default: 0
hsSonylCamAutoIris	true: on, false: off	bool	true/false, default: false
hsSonylCamAutoIrisTarget	Convergent value	short	(-8)-8, default: 0
hsSonylCamAutoIrisVoltageLower	Voltage magnitude lower limit	short	(-128)-127, default: 0
hsSonylCamAutoIrisVoltageStandard	Standard voltage setting	short	(-128)-127, default: 0

hsSonylCamParameterIndex	Description	Type	Range
hsSonylCamAutoIrisVoltageUpper	Voltage magnitude upper limit	short	(-128)-127, default: 0
hsSonylCamCameraXGain (X is R/G/B)	RGB gain value	short	0-32767, default: 4096
hsSonylCamCameraGain	Gain value, in units of 0.1 dB	short	0-180, default: 1
hsSonylCamCDSGain	CCD sensitivity shift adjustment	short	0-3, default: 0
hsSonylCamClampLevel	video black level	short	0-255, default: 0
hsSonylCamColorSupported [RO]	true: camera supports color, false: camera does not	bool	true/false
hsSonylCamCommonTriggerCount	Total common trigger count value	long	0-255
hsSonylCamCommonTriggerMode	See table below	long	0-0xFFFFFFFF, default: 0
hsSonylCamCommonTriggerReset	true: resets, false: does nothing	bool	true/false, default: false
hsSonylCamDIDLYSelect	DIO input delay output	short	0-31, default: 0
hsSonylCamDIDLYSRCSelect	DIO input signal. 0-3: digital input, 4: external trigger	short	0-4, default: 0
hsSonylCamDIDLYStep	DIO input delay time in μ s		0-65535, default: 0
hsSonylCamDOSelectN (N is 0-7)	Digital IO output signal	long	(-2147483648)-2147483647, default: 0
hsSonylCamFilter5x5XYZ (X is R/G/B, Y is 1-5, Z is 1-5)	5x5 filter values. X is the color channel, YZ are column, row	double	(-32)-32, default: 0
hsSonylCamFilterMode	true: on, false: off	bool	true/false, default: false
hsSonylCamGammaMode	See table below	short	0-4, default: 0
hsSonylCamGammaThreshold	Binning threshold	short	1-1023, default: 512
hsSonylCamGreyscaleAcquisition	true: acquire in grey-scale	bool	true/false, default: true
hsSonylCamGreyscalePattern	See table below	short	0-7, default: 0
hsSonylCamHeight [RO]	Maximum height of sensor	short	0-32767
hsSonylCamHorizontalBinning	0: off, 1: on	short	0/1, default: 0
hsSonylCamHorizontalPartialScan	true: on, false: off	bool	true/false, default: false
hsSonylCamHorizontalPartialScanStart	Start position	short	0-15, default: 0
hsSonylCamHorizontalPartialScanWidth	Scan width	short	0-15, default: 15
hsSonylCamLuminanceDetection	0: no detection, 1-3: uses 1, 2, or 4 frames	short	0-3, default: 1
hsSonylCamLuminanceDetectionHeightN (N is 1-4)	Detection frame height. VGA: 0-480, SXGA: 0-960	short	0-960
hsSonylCamLuminanceDetectionStartXN (N is 1-4)	Start point X-coordinate. VGA: 0-640, SXGA: 0-1280	short	0-1280
hsSonylCamLuminanceDetectionStartYN (N is 1-4)	Start point Y-coordinate. VGA: 0-480, SXGA: 0-960	short	0-960

hsSonylCamParameterIndex	Description	Type	Range
hsSonylCamLuminanceDetectionWidth N (N is 1-4)	Detection frame width. VGA: 0-640, SXGA: 0-1280	short	0-1280
hsSonylCamNonFilteredImage	true: transfers image before and after filter, false: only after	bool	true/false, default: false
hsSonylCamShutterDelay	Trigger delay in ms	short	0-4095, default: 0
hsSonylCamShutterMode	0: Monitoring mode, 1: Trigger mode	short	0/1, default: 0
hsSonylCamShutterPolarity	0: Positive, 1: Negative	short	0/1, default: 0
hsSonylCamShutterSpeed	See table below	short	0-15, default: 0
hsSonylCamStrobeDelay	Strobe delay in μ s	short	0-4095, default: 0
hsSonylCamTimeout	Timeout value in ms	long	1-2147483647, default: 5000
hsSonylCamTriggerCount	Trigger total count value	long	0-0xFFFFFFFF
hsSonylCamTriggerInhibit	0: Validates trigger interrupt, 1: Invalidated trigger interrupt	long	0/1, default: 0
hsSonylCamTriggerMode	0: off, 1: on	long	0/1, default: 0
hsSonylCamTriggerReset	true: resets trigger counter	bool	true/false, default: false
hsSonylCamVerticalBinning	0: off, 1: on	short	0/1, default: 0
hsSonylCamVerticalPartialScan	true: on, false: off	bool	true/false, default: false
hsSonylCamVerticalPartialScanHeight	Scan height	short	0-15, default: 15
hsSonylCamVerticalPartialScanStart	Start position	short	0-15, default: 0
hsSonylCamWhiteBalanceDetection	0: no detection, 1-3: uses 1, 2, or 4 frames	short	0-3, default: 1
hsSonylCamWhiteBalanceDetectionHeight N (N is 1-4, for frame number)	Detection frame height. VGA: 0-480, SXGA: 0-960	short	0-960
hsSonylCamWhiteBalanceDetectionStartX N (N is 1-4, for frame number)	Start point X-coordinate. VGA: 0-640, SXGA: 0-1280	short	0-1280
hsSonylCamWhiteBalanceDetectionStartY N (N is 1-4, for frame number)	Start point Y-coordinate. VGA: 0-480, SXGA: 0-960	short	0-960
hsSonylCamWhiteBalanceDetectionWidth N (N is 1-4, for frame number)	Detection frame width. VGA: 0-640, SXGA: 0-1280	short	0-1280
hsSonylCamWhiteBalanceMode	0: OFF 1: 3200K (preset 1) 2: 5600K (preset 2) 3: User preset (preset 3) 4: One-push white balance 5: Auto white balance	short	0-5
hsSonylCamWhiteBalanceTargetB	B side convergent value	short	(-128)-127, default: 0
hsSonylCamWhiteBalanceTargetR	R side convergent value	short	(-128)-127, default: 0
hsSonylCamWideDynamicRange	true: on, false: off	bool	true/false, default: false
hsSonylCamWideDynamicRangeEV	true: on, false: off	bool	true/false, default: false

hsSonylCamParameterIndex	Description	Type	Range
hsSonylCamWideDynamicRangeRelativeEV2	Relative value between first and second frame. Actual value = $18 - (<\text{this setting}> / 2)$	short	0-72, default: 0
hsSonylCamWideDynamicRangeSpeed1	See hsSonylCamShutterSpeed	short	0-15, default: 0
hsSonylCamWideDynamicRangeSpeed2	See hsSonylCamShutterSpeed	short	0-15, default: 0
hsSonylCamWideDynamicRangeSpeedEV2	Shutter speed of second frame. See table hsSonylCamWDRSpeedEV2	short	0-36, default: 0
hsSonylCamWidth [RO]	Maximum width of sensor	short	0-32767

hsSonylCamShutterSpeed	Description
0	1/90 for VGA and 1/30 for SXGA
1	2
2	1
3	1/15
4	1/30 for VGA and 1/60 for SXGA
5	1/100
6	1/120
7	1/250
8	1/500
9	1/1000
10	1/2000
11	1/4000
12	1/10000
13	1/25000
14	1/50000
15	1/100000

hsSonylCamGreyscalePattern	Description
0	CCD out
1	Grey scale for PS confirmation
2	Grey scale 1 for full-luminance output evaluation
3	Grey scale 2 for full-luminance output evaluation

hsSonylCamGreyscalePattern	Description
7	Color bars chart for color output evaluation

hsSonylCamCommonTriggerMode	Description
Bit 7 to 0	Counter interval
Bit 15 to 8	Valid count value (All triggers are valid in FFh)
Bit 23 to 16	Valid count value (All triggers are valid in FFh)
Bit 31 to 24	Valid count value (All triggers are valid in FFh)

hsSonylCamGammaMode	Description
0	Off
1	0.45
2	2.2
3	Reverse
4	Bin

hsSonylCamWDRSpeedEV2	Description
0	2.000000000
1	1.414213562
2	1.000000000
3	0.707106781
4	0.500000000
5	0.353553391
6	0.250000000
7	0.176776695
8	0.125000000
9	0.094280904
10	0.066666667
11	0.047140452
12	0.033333333
13	0.023570226
14	0.016666667
15	0.011785113
16	0.008333333
17	0.005656854
18	0.004000000
19	0.002828427
20	0.002000000
21	0.001414214
22	0.001000000

hsSonyICamWDRSpeedEV2	Description
23	0.000707107
24	0.000500000
25	0.000353553
26	0.000250000
27	0.000176777
28	0.000125000
29	0.000088388
30	0.000062500
31	0.000044194
32	0.000031250
33	0.000022097
34	0.000015625
35	0.000011049
36	0.000007813

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	hsSonyICamParameterIndex	Enum, see table below.
<i>Device</i>	Index number or name of the frame grabber device. Optional.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. This property applies to the configuration's currently selected input device if the *Device* parameter is not provided.

Example

```
' Reads the current trigger type
Dim Type As hsSonyICamTriggerType
Type = AcqDevice1.GrabberSonyICamParameter _
    (hsSonyICamTriggerType,"SONYICAM 0")
' Writes a new timeout value (5 seconds) for the current
' input device of the default configuration
AcqDevice1.GrabberSonyICamParameter(hsSonyICamTimeout) = 5000
```


ImageCoordinateSystemType

Description

Selects whether the coordinate system used to express pixel coordinates in the image is right-handed or left-handed. In a right-handed Image coordinate system (default), the origin is located at the bottom left corner of the image and the Y-axis is directed upward. In a left-handed Image coordinate system, the origin is located at the top left corner of the image and the Y-axis is directed downward.

Syntax

control.**ImageCoordinateSystemType** [= *Value* As hsCoordinateSystemType]

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

hsCoordinateSystem Type	Value	Description
hsRightHanded	0	The Image coordinate is right-handed (origin located at the bottom left corner and Y-axis directed upward).
hsLeftHanded	1	The Image coordinate is left-handed (origin located at the top left corner and Y-axis directed downward).

Remark(s)

A left-handed image as well as a right-handed image can be calibrated into either a left-handed or a right-handed World coordinate system. When the **ImageCoordinateSystemType** property of an image is changed, the image must be recalibrated to avoid any inconsistencies.

Example

```
' Acquire a new image into a left-handed uncalibrated
' image reference frame
AcqDevice1.ImageCoordinateSystemType = hsLeftHanded
AcqDevice1.Execute
```

InputDevice

Description

Index of the active input device that provides the image for a given configuration.

Syntax

control.**InputDevice**([*Configuration* As Variant]) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: **InputDeviceCount** - 1

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim InputDevice As Integer
InputDevice = AcqDevice1.InputDevice(1)
AcqDevice2.InputDevice("Config") = 1
```

InputDeviceAvailable

Description

Availability of a selected input device. Read only.

Syntax

control.**InputDeviceAvailable**(*Device* As Variant)

Type

Boolean

Range

True, False

Example

```
Dim DeviceAvailable As Boolean
DeviceAvailable = AcqDevice1.InputDeviceAvailable(3)
DeviceAvailable = AcqDevice1.InputDeviceAvailable ("METEOR2MC 0")
```

InputDeviceCount

Description

Number of input devices that from the input device list. Read only.

Syntax

control.**InputDeviceCount**

Type

Integer

Range

Minimum: 3
Maximum: 3 + number of frame grabber devices.

Parameter(s)

Not applicable.

Remark(s)

The first three input devices are always database selection, file emulation and generic frame grabber. To these three, add the number of frame grabber devices and you get the value of **InputDeviceCount**.

Example

```
Dim NInputDevices As Integer
```

`NInputDevices = Acquisition1.InputDeviceCount`

InputDeviceDescription

Description

Text description of a selected input device. Read only.

Syntax

control.**InputDeviceDescription**(*Device* As Variant)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the selected input device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```
Dim Description As String
Description = AcqDevice1.InputDeviceDescription(1)
Description = _
    AcqDevice2.InputDeviceDescription("Selection")
```

InputDeviceName

Description

Name of a selected input device. Read only.

Syntax

control.**InputDeviceName**(*Device* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the selected input device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```
Dim DeviceName As String
DeviceName = AcqDevice1.InputDeviceName(1)
```

InputDeviceType

Description

Type of a selected input device. Read only.

Syntax

control.**InputDeviceType**(*Device* As Variant)

Type

hsInputDeviceType

Range

The valid range for this property is as follows:

hsInputDeviceType	Value	Description
hsFrameGrabber	0	Hardware frame grabber.
hsFileEmulation	1	Image provided by a HexSight database file.
hsDatabaseSelection	2	Image provided by the runtime database.

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the selected input device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1

Example

```
Dim DeviceType As hsInputDeviceType
DeviceType = AcqDevice1.InputDeviceType(1)
DeviceType = AcqDevice1.InputDeviceType("Selection")
```

Name**Description**

Name used to access the tool through the Process Manager.

Syntax

control.**Name**

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ADName As String
ADName = AcqDevice1.Name
AcqDevice2.Name = "Acquisition"
```

OutputImage**OutputGreyScaleImage****Description**

Name of the output image provided by the selected input device for a given configuration. The input device can be an runtime database selection, a file emulation database or a frame grabber.

Syntax

control.**OutputImage**([*Configuration* As Variant]) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ImageName As String
ImageName = AcqDevice1.OutputImage ("Configuration0")
AcqDevice2.OutputImage(1) = "Image"
```

OutputView

Description

Name of the View where all the Images and the Scenes that can be generated by the Acquisition Device are output.

Syntax

control.**OutputView**([*Configuration* As Variant]) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ViewName As String
ViewName = AcqDevice1.OutputView("Configuration0")
AcqDevice2.OutputView(1) = "View"
```

PassThruRunning

Description

Specifies if the real time, pass through acquisition mode (**StartPassThru**) is active. Read only.

Syntax

control.**PassThruRunning**([Configuration]) [= *Value* As Boolean]

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Example

```
Dim PassThruRunning As Boolean
AcqDevice1.StopPassThru PictureBox1.hWnd
AcqDevice1.PassThruRunning = True
```

SelectionImage

SelectionGreyScaleImage

Description

Name of the Image in the runtime database that is used if **InputDevice** is set to Database Selection.

Syntax

control.**SelectionImage**([Configuration As Variant] [= *Value* As String])

Type

String

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ImageName As String
ImageName = AcqDevice1.SelectionImage(1)
AcqDevice2.SelectionImage("Configuration0") = "Image"
```

SelectionView

Description

Name of the View containing the input image in the runtime database for a given configuration. See **SelectionImage** for more information.

Syntax

control.**SelectionView**([Configuration As variant]) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Remark(s)

This property applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim ViewName As String
ViewName = AcqDevice1.SelectionView(1)
AcqDevice2.SelectionView("Configuration0") = "View"
```

TimeStampHigh

Description

Most significant 32 bits of the 64-bit time stamp of the image acquired by the last execution of the Acquisition Device. The time stamp is expressed as the number of packets of 100ns elapsed between January 1st 1601 and the time of the acquisition by the frame grabber.

TimeStampLow holds the other 32 bits of the time stamp. Read only.

Syntax

control.**TimeStampHigh**

Type

Long

Range

Unlimited

Parameter(s)

Not applicable.

Remark(s)

See also **TimeStampLow**.

Example

```
Dim High As Long
Dim Low As Long
High = AcqDevice1.TimeStampHigh
Low = AcqDevice1.TimeStampLow
```

TimeStampLow

Description

Least significant 32 bits of the 64-bit time stamp of the image acquired by the last execution of the Acquisition Device. The time stamp is expressed as the number of packets of 100ns elapsed between January 1st 1601 and the time of the acquisition by the frame grabber.

TimeStampHigh holds the other 32 bits of the time stamp. Read only.

Syntax*control.TimeStampLow***Type**

Long

Range

Unlimited

Parameter(s)

Not applicable.

Remark(s)See also **TimeStampHigh**.**Example**

```
Dim High As Long
Dim Low As Long
High = AcqDevice1.TimeStampHigh
Low = AcqDevice1.TimeStampLow
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax*control.TypeName***Type**

String

Range

Not applicable.

Remark(s)

This property is always "HSAcquisitionDevice".

Example

```
Dim TypeName As String
TypeName = AcqDevice1.TypeName
```

Version

Description

String containing the current HexSight version and build information. Read only.

Syntax*control.Version***Type**

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.3.000 R 3217 (release)".

Example

```
Dim Version As String
Version = AcquisitionDevice1.Version
```


Methods

AboutBox

Description

Displays the Acquisition Device tool About box.

Syntax

control.**AboutBox**

Parameter(s)

Not applicable.

Return

No return value.

Range

Not applicable.

Example

AcqDevice1.**AboutBox**

AppendImageToEmulationDatabase

Description

Acquires a new image using the current input source of a given configuration and appends it to its current emulation database.

Syntax

control.**AppendImageToEmulationDatabase**([*Configuration* As Variant], [*Name* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Name</i>	Name of the new image. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *Name* parameter is not provided, a default name is given to the new image.

Example

AcqDevice1.**AppendImageToEmulationDatabase** "Camera0", _
"Part_Rotated_10_Deg"

ControlPropertyExchange

Description

Reserved for internal use.

CreateConfiguration

Description

Adds a new configuration to the configuration list.

Syntax

control.**CreateConfiguration**(*Name* As String)

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the configuration to create.	String	Not applicable.

Return

Integer. Index of the newly created configuration.

Range

Minimum: 0

Maximum: unlimited

Example

AcqDevice1.**CreateConfiguration** "NewConfiguration"

DeleteConfiguration

Description

Removes the selected configuration from the configuration list.

Syntax

control.**DeleteConfiguration**(*Configuration* As Variant)

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

No return value.

Range

Not applicable.

Example

AcqDevice1.**DeleteConfiguration** 1

AcqDevice1.**DeleteConfiguration** "Configuration0"

EmulationDeleteImage

Description

Deletes the selected image from the emulation database of a given configuration.

Syntax

control.**EmulationDeleteImage**(*Index* As Integer , [*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the image to delete.	Integer	Minimum: 0 Maximum: EmulationImageCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

AcqDevice1.**EmulationDeleteImage** 2, "Configuration2"

EmulationDisableAll

Description

Disables all the images of an emulation database for a given configuration.

Syntax

control.**EmulationDisableAll**([*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

AcqDevice1.**EmulationDisableAll** 1
AcqDevice1.**EmulationDisableAll** "Configuration0"

EmulationEnableAll

Description

Enables all the images of an emulation database for a given configuration.

Syntax

control.**EmulationEnableAll**([*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

AcqDevice1.**EmulationEnableAll** 1

EmulationExport

Description

Saves the current image to a specified destination, as a HexSight image file (*.hig) or as a Windows Bitmap(*.bmp) file. An *.hig file conserves the image calibration as well as all other image properties.

Syntax

control.**EmulationExport**(*Index* As Integer, [*Configuration* As Variant], [*FileName* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the emulation image to save.	Integer	Minimum: 0 Maximum: EmulationImageCount - 1
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the image file to export. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the parameter *Configuration* is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file save dialog appears.

Example

Acquisition.**EmulationExport** 1, "RotatedParts", "Part_120_Deg.hig"

EmulationImport

Description

Imports a new image to the emulation database of a given configuration.

Syntax

control.**EmulationImport**([*Configuration* As Variant], [*FileName* As Variant],[*BMPCoordinateSystem* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the image file to import. Optional.	Variant	Not applicable.
<i>BMPCoordinateSystem</i>	Coordinate system to be used to express pixel coordinates in a bmp (Bitmap) image. Not applicable to hlg image format.	hsCoordinateSystem Type	Enum, see table below.

Constant	Value	Description
hsRightHanded	0	The Image coordinate is right-handed (origin located at the bottom left corner and Y-axis directed upward).
hsLeftHanded	1	The Image coordinate is left-handed (origin located at the top left corner and Y-axis directed downward).

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file open dialog appears. If the *BMPCoordinateSystem* is not provided the default *hsRightHanded* is applied

Example

```
AcqDevice1.EmulationImport "RotatedParts", "Part_120_Deg.bmp"
```

Execute

Description

Starts the execution of the Acquisition Device for a given configuration and waits for the oldest execution request in the queue to complete. Calling the **Execute** method is the same as making subsequent calls to the **Start** and **Wait** methods.

Syntax*control*.**Execute****Parameter(s)**

Not applicable.

Return

Single. Execution time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

Example

```
' Add one acquisition request to initiate the pipeline
' using the default configuration
AcqDevice1.Start
' Execution loop
Do
    ' Execute implicitly calls Start and Wait
    ' with the default configuration
    Time = AcqDevice1.Execute
    ' Update the last execution time text box
    TimeEdit.Text = Time
    DoEvents
Loop While Running = True
' Flush the pipeline
AcqDevice1.Wait
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GrabberDepict4x1GetBusyCount

Description

Returns the missed trigger count for the specified camera since the monitor was reset. This method is not indexed by a configuration, but by a device. If the specified device is not a DepictMaster 4x1 controller board or the specified camera isn't present, an error is fired and no internal actions are taken. Use the **GrabberDepict4x1ResetBusyCount** method to reset and to enable the busy count.

Syntax*control*.**GrabberDepict4x1CameraGetBusyCount**(*Device As Variant, Camera As Short*).

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the controller board device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Camera</i>	Camera port.	Short	Minimum: 0 Maximum: 3

Return

Long

Range

Minimum: 0

Maximum: Unlimited

Example

```
' Get the Camera Busy Count
Dim BusyCount As Long
BusyCount = AcqDevice1.GrabberDepict4x1GetBusyCount( _
    "DEPICT4X1 0", 0 )
```

GrabberDepict4x1ResetBusyCount

Description

Resets the counting of missed acquisition triggers for the specified camera and enables it. This method is not indexed by a configuration, but by a device. If the specified device is not a DepictMaster 4x1 controller board or the specified camera isn't present, an error is fired and no internal actions are taken. The busy, or missed trigger event count can be retrieved using the **GrabberDepict4x1GetBusyCount** method.

Syntax

control.**GrabberDepict4x1CameraResetBusyCount**(*Device* As Variant, *Camera* As Short).

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the controller board device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Camera</i>	Camera port.	Short	Minimum: 0 Maximum: 3

Return

No return value.

Range

Not applicable.

Example

```
' Reset and enable the Camera Busy Monitor
AcqDevice1.GrabberDepict4x1ResetBusyCount _
    "DEPICT4X1 0", 0
```

GrabberDirectShowDialog

Description

Opens the Stream Format and/or the Video Parameters dialog for the selected DirectShow device.

Syntax

```
control.DirectGrabberShowDialog( [Selection As hsDirectShowDialog], [Device As Variant], [Configuration As Variant]) As Boolean
```

Parameter(s)

Name	Description	Type	Range
<i>Selection</i>	Specifies the dialog to be opened.	hsDirectShowDialog	Not applicable.
<i>Device</i>	Index number or name of the DirectShow device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

<i>Selection</i> Parameter	Value	Description
hsDirectShowDialogStreamFormat	0	Opens the Stream Format dialog.
hsDirectShowDialogVideoParameters	1	Opens the Video Parameters dialog.

Return

Boolean

Range

Returns True if operation succeeds, otherwise returns False.

Example

```
AcquisitionDevice1.GrabberDirectShowDialog True, _  
    "Camera 1", 1
```

GrabberPx610aFireStrobe

Description

Generates a pulse on the two strobe lines of an Imagenation PX610A frame grabber.

Syntax

```
control.GrabberPx610aFireStrobe(Device As Variant, Command As  
hsPx610aStrobeCommands)
```

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Command</i>	Selects the action to be performed on the strobe lines.	hsPx610aStrobeCommands	Enum, see table below.

hsPx610aStrobeCommands	Description
hsPx610aStrobeCommandStop	Turns off all strobe activity immediately for both normal and triggered strobe modes. Does not affect the strobe mode.
hsPx610aStrobeCommandFullSequence	Fires a complete strobe sequence (strobe 0, gap, strobe 1).
hsPx610aStrobeCommandGap	Fires the gap, followed by strobe 1.
hsPx610aStrobeCommandStrobe1	Fires strobe 1 only.

Return

No return value.

Range

Not applicable.

Example

```
AcqDevice1.GrabberPc610aFireStrobe _
    "PX610A 1", hsPx610aStrobeCommandStrobe1
```

GrabberPxc200FireStrobe

Description

Generates a pulse on the specified output lines of an Imagenation PXC200 frame grabber.

Syntax

control.**GrabberPxc200FireStrobe**(*Device* As Variant, *Line4* As Boolean, *Line5* As Boolean, *Line6* As Boolean, *Line7* As Boolean)

Parameter(s)

Name	Description	Type	Range
<i>Device</i>	Index number or name of the frame grabber device.	Variant	Minimum: 0 Maximum: InputDeviceCount - 1
<i>Line4</i>	If true, a pulse will be generated on the output line 4 of the specified Imagenation PXC200 frame grabber.	Boolean	True, False
<i>Line5</i>	If true, a pulse will be generated on the output line 5 of the specified Imagenation PXC200 frame grabber.	Boolean	True, False
<i>Line6</i>	If true, a pulse will be generated on the output line 6 of the specified Imagenation PXC200 frame grabber.	Boolean	True, False
<i>Line7</i>	If true, a pulse will be generated on the output line 7 of the specified Imagenation PXC200 frame grabber.	Boolean	True, False

Return

No return value.

Range

Not applicable.

Example

```
AcqDevice1.GrabberPxc200FireStrobe _
```

"PXC200 0", True , False , False , False

HideCalibrationDialog

Description

Hides and terminates the active calibration dialog.

Syntax

control.**HideCalibrationDialog**

Parameter(s)

Parameter	Description	Type	Range
<i>ResultCode</i>	Specifies whether the calibration will be discarded or applied to the associated Acquisition Device Configuration.	hsCalibrationDialogResult	Enum. See table below.

hsCalibrationDialogResult	Value	Description
hsResultCancel	2	The calibration is discarded
hsResultOk	1	The calibration is applied to the associated Acquisition Device Configuration

Return

Type	Description	Range
<i>Boolean</i>	Specifies if the interface was displayed when the method was called.	True, False

Range

Not applicable.

Example

AcqDevice1.**HideCalibrationDialog** hsOK

HideColorCalibrationDialog

Description

Hides and terminates the active color calibration dialog.

Syntax

control.**HideColorCalibrationDialog**

Parameter(s)

Parameter	Description	Type	Range
<i>ResultCode</i>	Specifies whether the color calibration will be discarded or applied to the associated Acquisition Device Configuration.	hsCalibrationDialogResult	Enum. See table below.

hsCalibrationDialogResult	Value	Description
hsResultCancel	2	The color calibration is discarded
hsResultOk	1	The color calibration is applied to the associated Acquisition Device Configuration

Return

Type	Description	Range
<i>Boolean</i>	Specifies if the interface was displayed when the method was called.	True, False

Range

Not applicable.

Example

```
AcqDevice1.HideColorCalibrationDialog hsOK
```

HideProperties

Description

Hides the tool properties interface.

Syntax

```
control.HideProperties
```

Parameter(s)

None

Return

Type	Description	Range
<i>Boolean</i>	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = AcqDevice1.HideProperties
```

HideRealTimeDisplay

Description

Hides the tool real time display if visible.

Syntax

```
control.HideRealTimeDisplay
```

Parameter(s)

Not applicable.

Return

No return value.

Range

Not applicable.

Example

```
AcqDevice1.HideRealTimeDisplay
```

ImageToWorld

Description

This method takes two parameters (X and Y) that represent a point in the Image coordinate system. When the method is executed, the two entered parameters are modified to express the same point with respect to the World coordinate system.

Syntax

control.**ImageToWorld**(X As Single, Y As Single) As Boolean]

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point	Single	Any Single value.
Y	Y coordinate of the point	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 200
Y1 = 350
Succeed = AcqDevice1.ImageToWorld(X1,Y1)
```

LoadCalibration

Description

Loads a HexSight calibration file that replaces the current calibration for a given configuration. The calibration file is created with the **SaveCalibration** method.

Syntax

control.**LoadCalibration**([*Configuration* As Variant], [*FileName* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the calibration file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file open dialog appears.

Example

```
AcqDevice1.LoadCalibration "Configuration0", "Calib.cal"
```

LoadColorCalibration

Description

Loads a HexSight color calibration file that replaces the current color calibration for a given configuration. The color calibration file is created with the **SaveColorCalibration** method.

Syntax

```
control.LoadColorCalibration([Configuration As Variant], [FileName As Variant])
```

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the color calibration file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file open dialog appears.

Example

```
AcqDevice1.LoadColorCalibration "Configuration0", "ColorCalib.cal"
```

LoadConfiguration

Description

Reserved for internal use.

LoadEmulationDatabase

Description

Loads a new emulation database file into a given configuration.

Syntax

```
control.LoadEmulationDatabase([Configuration As Variant], [FileName As Variant])
```

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the database file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file open dialog appears.

Example

```
AcqDevice1.LoadEmulationDatabase "Configuration0", "Data.hdb"
```

LoadGrabberParameters

Description

Loads a set of frame grabber parameters.

Syntax

```
control.LoadGrabberParameters([Configuration As Variant], [FileName As Variant])
```

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the database file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file load dialog appears.

Example

```
Acqisition.LoadGrabberParameters "Configuration0", "grabber.fgp"
```

NewEmulationDatabase

Description

Clears the current emulation database of a given configuration.

Syntax

control.**NewEmulationDatabase**([*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

AcqDevice1.**NewEmulationDatabase** "Channel0"

SaveCalibration

Description

Saves the calibration information of a given configuration to a file. This file can later be imported into another application using the same camera setup by using the **LoadCalibration** method.

Syntax

control.**SaveCalibration**([*Configuration* As Variant], [*FileName* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the database file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file save dialog appears.

Example

AcqDevice1.**SaveCalibration** "Config", "CalibrationFile"

SaveColorCalibration

Description

Saves the color calibration information of a given configuration to a file. This file can later be imported into another application using the same camera setup by using the **LoadColorCalibration** method.

Syntax

control.**SaveColorCalibration**([*Configuration* As Variant], [*FileName* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the color calibration file to save. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file save dialog appears.

Example

AcqDevice1.**SaveColorCalibration** "Config", "ColorCalibrationFile"

SaveConfiguration

Description

Reserved for internal use.

SaveEmulationDatabase

Description

Saves the current emulation database of a given configuration.

Syntax

control.**SaveEmulationDatabase**([*Configuration* As Variant], [*FileName* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the database file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file save dialog appears.

Example

```
AcqDevice1.SaveEmulationDatabase "Camera1", "Camera1Emulation.hdb"
```

SaveGrabberParameters

Description

Saves the set of configuration parameters used by the current frame grabber. This file can later be imported and applied to a same-

frame grabber in another application by using the **LoadGrabberParameters** method.

Syntax

```
control.SaveGrabberParameters([Configuration As Variant], [ FileName As Variant])
```

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>FileName</i>	Name of the database file to load. Optional.	Variant	Not applicable.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *FileName* parameter is not provided, a file save dialog appears.

Example

```
Acquisition.SaveGrabberParameters "Config", _
    "GrabberFile"
```

SetDebugParameter

Description

Reserved for internal use.

ShowCalibDistortionLUTDialog

Description

This function opens a modeless dialog that is used to define the calibration parameters for a configuration. A reference to a **HSCalibrationDistortionInterface** object is returned. This object can be used to carry out the calibration procedure programmatically.

Syntax

control.**ShowCalibDistortionLUTDialog**(*Selection* as Boolean, [*Device* As Variant]
[*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Visible</i>	If TRUE, the dialog is displayed once it is created. If the dialog is not displayed at creation time, it can be displayed later using the object's Show method.	Boolean	False, True

Return

hsCalibrationDistortionInterface

Range

Not applicable.

Remark(s)

Applies to the default configuration property (**ConfigurationDefault**) if no *Configuration* parameter is specified.

Example

```
Dim CalibDialog As HSCalibrationDistortionInterface
CalibDialog = ShowCalibDistortionLUTDialog(False)
CalibDialog = ShowCalibDistortionLUTDialog(True, 1)
CalibDialog = ShowCalibDistortionLUTDialog (True, "Config 0")
```

ShowCalibDistortionModelDialog

Description

This function opens a modeless dialog that is used to define the calibration parameters for a configuration. A reference to a **HSCalibrationDistortionInterface** object is returned. This object can be used to carry out the calibration procedure programmatically.

Syntax

control.**ShowCalibDistortionModelDialog**(*Visible* As Boolean, [*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Visible</i>	If TRUE, the dialog is displayed once it is created. If the dialog is not displayed at creation time, it can be displayed later using the object's Show method.	Boolean	False, True

Return

hsCalibrationDistortionInterface

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim CalibDialog As HSCalibrationDistortionInterface
CalibDialog = ShowCalibDistortionModelDialog(False)
CalibDialog = ShowCalibDistortionModelDialog(True, 1)
CalibDialog = ShowCalibDistortionModelDialog(True, "Config 0")
```

ShowCalibPerspectiveDialog

Description

This function opens a modeless dialog that is used to define the calibration parameters for a configuration. A reference to a **HSCalibrationPerspectiveInterface** object is returned. This object can be used to carry out the calibration procedure programmatically.

Syntax

control.**ShowCalibPerspectiveDialog**(*Visible* As Boolean , [*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Visible</i>	If TRUE, the dialog is displayed once it is created. If the dialog is not displayed at creation time, it can be displayed later using the object's Show method.	Boolean	False, True

Return

hsCalibrationPerspectiveInterface

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim CalibDialog As HSCalibrationPerspectiveInterface
CalibDialog = ShowCalibPerspectiveDialog(False)
CalibDialog = ShowCalibPerspectiveDialog(True, 1)
CalibDialog = ShowCalibPerspectiveDialog (True, "Config 0")
```

ShowCalibXYScaleDialog

Description

This function opens a modeless dialog that is used to define the calibration parameters for a configuration. A reference to a HSCalibrationXYScaleInterface object is returned. This object can be used to carry out the calibration procedure programmatically.

Syntax

control.**ShowCalibXYScaleDialog**(*Visible* As Boolean , [*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Visible</i>	If TRUE, the dialog is displayed once it is created. If the dialog is not displayed at creation time, it can be displayed later using the object's Show method.	Boolean	False, True

Return

hsCalibrationXYScaleInterface

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
Dim CalibDialog As HSCalibrationXYScaleInterface
CalibDialog = ShowCalibXYScaleDialog(False)
CalibDialog = ShowCalibXYScaleDialog(True, 1)
CalibDialog = ShowCalibXYScaleDialog(True, "Config 0")
```

ShowColorCalibrationDialog

Description

This function opens a modeless dialog that is used to define the color calibration parameters for a configuration.

Syntax

control.**ShowColorCalibrationDialog**(*Visible* as Boolean, [*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Visible</i>	If TRUE, the dialog is displayed once it is created. If the dialog is not displayed at creation time, it can be displayed later using the object's Show method.	Boolean	False, True
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

hsColorCalibrationFCMInterface

Range

Not applicable.

Remark(s)

Applies to the default configuration property (**ConfigurationDefault**) if no *Configuration* parameter is specified.

Example

```
Dim CalibDialog As HSColorCalibrationFCMInterface
CalibDialog = ShowColorCalibrationDialog True, 1
```

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean , [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and what is the behavior of the control when the OK button is clicked. When this parameter is not provided all the elements of the bitfield are set except hsExecuteOnOK and hsExecuteRemaining.	Variant	The elements of the bitfield are displayed in the table below.

Option Field	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the bitfields hsExecuteOnOK and hsExecuteRemaining are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise returns False.

Example

```
AcqDevice1.ShowProperties True,hsShowOK + hsShowExecute
```

ShowRealTimeDisplay

Description

Shows the real time display for the selected input device of a given configuration. If the specified input device is not a frame grabber, if the frame grabber is not available or if the tool real time display is already visible, an error is fired.

Syntax

control.**ShowRealTimeDisplay**(*Modal* As Boolean , [*Configuration* As Variant], [*Zoom* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	Set if display window is modal or not.	Boolean	True, false
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Zoom</i>	Zoom factor applied to the image in the real time display.	Variant	100, 75, 50, 25.

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. If the *Zoom* parameter is not provided, a zoom factor of 100% is applied to the image in the real time display. The zoom factor can also be interactively modified using the window context menu.

Example

Acquisition.**ShowRealTimeDisplay** False, 1, 75

Start

Description

Adds a new processing request in the acquisition pipeline for a given configuration. The new request will start any hardware interaction as soon as the targeted hardware resource is available, even if the request is not at the head of the processing queue. Calls to **Start** are typically non-blocking. The processing request is handled asynchronously by independent threads. The **Wait** method is used to complete a request added to the processing queue by **Start**.

The blocking parameter is set by default to FALSE. You must set this parameter to TRUE if you want to ensure that the GRAB request is actually sent to the grabber before the call returns.

Syntax

control.**Start**([*Configuration* As Variant] , [*Blocking* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration. Optional.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Name	Description	Type	Range
<i>Blocking</i>	Blocks the start request until after the grabber request has been sent to the frame grabber.	Variant	True, False

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details. By default **Start** is non-blocking (*Blocking*= False)

Example

```
' Add one acquisition request to initiate the pipeline
' using the default configuration
AcqDevice1.Start
' Execution loop
Do
    ' Start a new acquisition request
    ' with the default configuration
    AcqDevice1.Start
    ' Wait for the oldest request to complete
    AcqDevice1.Wait
    ' Process the image
    ExecuteImageProcess
    DoEvents
Loop While Running = True
' Flush the pipeline
AcqDevice1.Wait
```

StartPassThru

Description

Starts the real time pass through acquisition mode. In this mode, images are grabbed continuously and transferred directly into video memory by the frame grabber. The pass through mode uses significantly less CPU time than the real time display. The downside of the pass through mode however is that unlike the real time display, the Acquisition Device cannot acquire images from the frame grabber while it is active. The pass through acquisition mode must be stopped with the **StopPassThru** method before calling the **Execute** or **Start** methods to acquire new images.

Syntax

control.**StartPassThru**(*WindowHandle* As OLE_HANDLE , [*Configuration* As Variant], [*Option* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>WindowHandle</i>	Handle of the window where the live images will be displayed.	OLE_HANDLE	0 to create a new modeless window or the handle to an existing window.
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1
<i>Option</i>	The method used to display the image.	hsDirectShowPassThru	Enum, see table below.

hsDirectShowParameterOutputSource	Value	Description
hsDirectShowPassThruStretch	0	Image is stretched to fit.
hsDirectShowPassThruCentered	1	Image is centered.
hsDirectShowPassThruNormal	2	Image displayed normally.

Return

No return value.

Range

Not applicable.

Remark(s)

If the *WindowHandle* parameter is set to 0, the pass through mode will create a modeless window in which the live images will be output. If the *WindowHandle* parameter is set to the handle of an existing window, the pass through mode will output the live images in that window. This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
' Start the pass through mode and display the live images
' in an existing Picture Box
AcqDevice1.StartPassThru PictureBox1.hWnd
```

StopPassThru

Description

Stops the real time pass through acquisition mode.

Syntax

```
control.StopPassThru([Configuration As Variant])
```

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

No return value.

Range

Not applicable.

Remark(s)

This method applies to the default configuration if the *Configuration* parameter is not provided. See the **ConfigurationDefault** property for more details.

Example

```
' Stop the pass through mode
AcqDevice1.StopPassThru
```

ValidateCalibration

Description

Validates the calibration used by a given configuration. Returns true if the calibration is valid. After a new **hsAuto** calibration (see **CalibrationMethod**) is created **ValidateCalibration** should be used to make sure the calibration works properly. After a new **hsAuto** calibration is created, the **Execute** method must be called once before calling **ValidateCalibration** in order to create the calibration internally.

Syntax

control.**ValidateCalibration**([*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

Boolean

Range

Returns True if the calibration is valid.

Example

```
' Create a new hsAuto calibration in the current
' Acquisition Device configuration
AcqDev.CalibrationMethod = hsAuto
' Number of pairs of calibration points
AcqDev.CalibrationParameter(0) = 4
' Calibrate into a left-handed coordinate system
AcqDev.CalibrationParameter(1) = 1
' Calibration point 1
AcqDev.CalibrationParameter(10) = 100.0
AcqDev.CalibrationParameter(11) = 50.0
AcqDev.CalibrationParameter(12) = -1500.0
AcqDev.CalibrationParameter(13) = 2300.0
' Calibration point 2
AcqDev.CalibrationParameter(14) = 500.0
AcqDev.CalibrationParameter(15) = 50.0
AcqDev.CalibrationParameter(16) = 1500.0
```

```

AcquDev.CalibrationParameter(17) = 2300.0
' Calibration point 3
AcquDev.CalibrationParameter(18) = 100.0
AcquDev.CalibrationParameter(19) = 400.0
AcquDev.CalibrationParameter(20) = -1500.0
AcquDev.CalibrationParameter(21) = 6100.0
' Calibration Point 4
AcquDev.CalibrationParameter(22) = 500.0
AcquDev.CalibrationParameter(23) = 400.0
AcquDev.CalibrationParameter(24) = 1500.0
AcquDev.CalibrationParameter(25) = 6100.0
' Execute the Acquisition Device process in order to
' create the calibration internally
AcquDev.Execute
' Validate the newly created calibration
Dim CalibrationOk As Boolean
CalibrationOk = AcquDev.ValidateCalibration

```

ValidateColorCalibration

Description

Validates the color calibration used by a given configuration. Returns true if the calibration is valid.

Syntax

control.**ValidateColorCalibration**([*Configuration* As Variant])

Parameter(s)

Name	Description	Type	Range
<i>Configuration</i>	Index or name of the selected configuration.	Variant	Minimum: 0 Maximum: ConfigurationCount - 1

Return

Boolean

Range

Returns True if the color calibration is valid.

Example

```

Dim CalibrationOk As Boolean
CalibrationOk = AcquDev.ValidateColorCalibration

```

Wait

Description

Completes the oldest processing request in the acquisition pipeline. The method will first wait for the hardware to finish the acquisition process before the necessary software processing is performed.

Syntax*control.Wait***Parameter(s)**

Not applicable.

Return

Integer. Index of the configuration used for the processing request.

Range

Minimum: 0

Maximum: **ConfigurationCount** - 1**Example**

```

' Add one acquisition request to initiate the pipeline
' using the default configuration
AcqDevice1.Start
' Execution loop
Do
    ' Start a new acquisition request
    ' with the default configuration
    AcqDevice1.Start
    ' Wait for the oldest request to complete
    AcqDevice1.Wait
    ' Process the image
    ExecuteImageProcess
    DoEvents
Loop While Running = True
' Flush the pipeline
AcqDevice1.Wait

```

WorldToImage

Description

This method takes two parameters (X and Y) that represent a point in the World coordinate system. When the method is executed, the two entered parameters are modified to express the same point with respect to the Image coordinate system.

Syntax*control.WorldToImage*(X As Single, Y As Single) As Boolean**Parameter(s)**

Parameter	Description	Type	Range
X	X coordinate of the point	Single	Any Single value.
Y	Y coordinate of the point	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```

Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 120.3
Y1 = 35.0
Succeed = AcqDevice1.WorldToImage(X1,Y1)

```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

ItlInportInterrupt

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

HSCalibrationDistortionInterface

Description

The HSCalibrationDistortionInterface control provides programmatic access to the Distortion Model and Distortion LUT calibration dialogs of the Acquisition Device. It can be used to open the dialog and configure target detection and dimension parameters to propose default values customized for a specific application. Target detection and calibration system solving can also be executed programmatically. The calibration dialog can also be hidden so that the entire calibration procedure can be carried out without having to show the interactive user interface.

Properties

AverageImageError

Description

Average error between the positions of the detected calibration and validation dots and their corresponding world coordinates projected in the image using the calibration.

AverageImageError is expressed in pixels. This property is available only after the calibration system is solved. Read only.

Syntax

control.AverageImageError

Type

Single

Range

Greater than zero.

Example

```
Dim AverageError As Single
Calibration.Solve
AverageError = Calibration.AverageImageError
```

AveragePixelHeight

Description

Average calibrated height of a pixel in the image, expressed in the calibrated units specified by the **UnitsLength** property. This property is available only after the calibration system is solved. Read only.

Syntax

control.AveragePixelHeight

Type

Single

Range

Greater than zero.

Example

```
Dim h As Single
```

```
Calibration.Solve
h = Calibration.AveragePixelHeight
```

AveragePixelWidth

Description

Average calibrated width of a pixel in the image, expressed in the calibrated units specified by the **UnitsLength** property. This property is available only after the calibration system is solved. Read only.

Syntax

```
control.AveragePixelWidth
```

Type

Single

Range

Greater than zero.

Example

```
Dim w As Single
Calibration.Solve
w = Calibration.AveragePixelWidth
```

AverageWorldError

Description

Average error between the actual world positions of the calibration and validation dots and their corresponding detected image coordinates projected in the world coordinate system using the calibration. **AverageWorldError** is expressed in the calibrated units specified by the **UnitsLength** property. This property is available only after the calibration system is solved. Read only.

Syntax

```
control.AverageWorldError
```

Type

Single

Range

Greater than zero.

Example

```
Dim AverageError As Single
Calibration.Solve
AverageError = Calibration.AverageWorldError
```

AxisXPositionX

Description

The approximate X position in the image of the first dot along the X-axis of the world coordinate system. Along with the **AxisXPositionY**, **AxisYPositionX**, **AxisYPositionY**, **OriginPositionX**, and **OriginPositionY** properties, this property is used to identify the approximate position of the 3 dots defining the world coordinate system before calling a target detection. Once the target is detected, the value of **AxisXPositionX** is automatically modified

to reflect the actual X position of the first dot detected along the X-axis of the world coordinate system. **AxisXPositionX** is expressed in pixels.

Syntax

control.**AxisXPositionX** [= Value As Single]

Type

Single

Range

Minimum: 0.

Maximum: width of the image.

Example

```
Calibration.OriginPositionX = 30
Calibration.OriginPositionY = 30
Calibration.AxisXPositionX = 60
Calibration.AxisXPositionY = 30
Calibration.AxisYPositionX = 30
Calibration.AxisYPositionY = 60
Calibration.Detect
```

AxisXPositionY

Description

The approximate Y position in the image of the first dot along the X-axis of the world coordinate system. Along with the **AxisXPositionX**, **AxisYPositionX**, **AxisYPositionY**, **OriginPositionX**, and **OriginPositionY** properties, this property is used to identify the approximate position of the 3 dots defining the world coordinate system before calling a target detection. Once the target is detected, the value of **AxisXPositionY** is automatically modified to reflect the actual X position of the first dot detected along the X-axis of the world coordinate system. **AxisXPositionY** is expressed in pixels.

Syntax

control.**AxisXPositionY** [= Value As Single]

Type

Single

Range

Minimum: 0.

Maximum: height of the image.

Example

```
Calibration.OriginPositionX = 30
Calibration.OriginPositionY = 30
Calibration.AxisXPositionX = 60
Calibration.AxisXPositionY = 30
Calibration.AxisYPositionX = 30
Calibration.AxisYPositionY = 60
Calibration.Detect
```

AxisYPositionX

Description

The approximate X position in the image of the first dot along the Y-axis of the world coordinate system. Along with the **AxisXPositionX**, **AxisXPositionY**, **AxisYPositionY**, **OriginPositionX**, and **OriginPositionY** properties, this property is used to identify the approximate position of the 3 dots defining the world coordinate system before calling a target detection. Once the target is detected, the value of **AxisYPositionX** is automatically modified to reflect the actual X position of the first dot detected along the X-axis of the world coordinate system. **AxisYPositionX** is expressed in pixels.

Syntax

control.**AxisYPositionX** [= Value As Single]

Type

Single

Range

Minimum: 0.

Maximum: width of the image.

Example

```
Calibration.OriginPositionX = 30
Calibration.OriginPositionY = 30
Calibration.AxisXPositionX = 60
Calibration.AxisXPositionY = 30
Calibration.AxisYPositionX = 30
Calibration.AxisYPositionY = 60
Calibration.Detect
```

AxisYPositionY

Description

The approximate Y position in the image of the first dot along the Y-axis of the world coordinate system. Along with the **AxisXPositionX**, **AxisXPositionY**, **AxisYPositionX**, **OriginPositionX**, and **OriginPositionY** properties, this property is used to identify the approximate position of the 3 dots defining the world coordinate system before calling a target detection. Once the target is detected, the value of **AxisYPositionY** is automatically modified to reflect the actual Y position of the first dot detected along the Y-axis of the world coordinate system. **AxisYPositionY** is expressed in pixels.

Syntax

control.**AxisYPositionY** [= Value As Single]

Type

Single

Range

Minimum: 0

Maximum: height of the image

Example

```
Calibration.OriginPositionX = 30
Calibration.OriginPositionY = 30
Calibration.AxisXPositionX = 60
```



```
Calibration.AxisXPositionY = 30
Calibration.AxisYPositionX = 30
Calibration.AxisYPositionY = 60
Calibration.Detect
```

CalibrationDotCount

Description
The number of calibration dots detected in the image. Calibration dots are used to compute the calibration parameters. Read only.

Syntax
control.**CalibrationDotCount**

Type
Integer

Range
Minimum: 0
Maximum: Unlimited

Example

```
Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.CalibrationDotCount - 1
    x = Calibration.CalibrationDotPositionXImage(d)
    y = Calibration.CalibrationDotPositionYImage(d)
Next d
```

CalibrationDotPositionXImage

Description
The X position in the image of the calibration dot selected with the **Index**.
CalibrationDotPositionXImage parameter, expressed in pixels. Read only.

Syntax
control.**CalibrationDotPositionXImage**(*Index* As Integer)

Type
Single

Range
Minimum: 0.
Maximum: width of the image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: CalibrationDotCount - 1

Example

```
Dim d As Integer
Dim x As Single
```

```

Dim y As Single
For d = 0 To Calibration.CalibrationDotCount - 1
    x = Calibration.CalibrationDotPositionXImage(d)
    y = Calibration.CalibrationDotPositionYImage(d)
Next d

```

CalibrationDotPositionXWorld

Description

The X position in the world coordinate system of the calibration dot selected with the **Index**. **CalibrationDotPositionXWorld** parameter, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**CalibrationDotPositionXWorld**(*Index* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: CalibrationDotCount - 1

Example

```

Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.CalibrationDotCount - 1
    x = Calibration.CalibrationDotPositionXWorld(d)
    y = Calibration.CalibrationDotPositionYWorld(d)
Next d

```

CalibrationDotPositionYImage

Description

The Y position in the image of the calibration dot selected with the **Index**. **CalibrationDotPositionYImage** parameter, expressed in pixels. Read only.

Syntax

control.**CalibrationDotPositionYImage**(*Index* As Integer)

Type

Single

Range

Minimum: 0
Maximum: height of the image

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: CalibrationDotCount - 1

Example

```

Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.CalibrationDotCount - 1
    x = Calibration.CalibrationDotPositionXImage(d)
    y = Calibration.CalibrationDotPositionYImage(d)
Next d

```

CalibrationDotPositionYWorld**Description**

The Y position in the world coordinate system of the calibration dot selected with the **Index**. **CalibrationDotPositionYWorld** parameter, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**CalibrationDotPositionYWorld**(*Index* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: CalibrationDotCount - 1

Example

```

Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.CalibrationDotCount - 1
    x = Calibration.CalibrationDotPositionXWorld(d)
    y = Calibration.CalibrationDotPositionYWorld(d)
Next d

```

DetectionCount

Description

Number of images that are used to locate the calibration and validation dots on the target. When a value greater than 1 is set, the dots are located for each image. Averaged positions are used to solve the calibration.

Syntax

control.**DetectionCount** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 100

Example

```
Dim i As Integer
Calibration.DetectionCount = 10
i = Calibration.DetectionCount
```

DisplayContourSceneEnabled

Description

When **DisplayContourSceneEnabled** is set to True, the contours of the detected calibration and validation dots are shown in the display when the **DisplayMode** property is set to **hsDots**.

Syntax

control.**DisplayContourSceneEnabled** [= *Value* As Boolean]

Type

Boolean

Range

True, False

Example

```
Calibration.DisplayContourSceneEnabled = True
```

DisplayImageEnabled

DisplayGreyScaleImageEnabled

Description

When **DisplayImageEnabled** is set to True, the image is shown in background in the display when the **DisplayMode** property is set to **hsImageError**.

Syntax

control.**DisplayImageEnabled** [= *Value* As Boolean]

Type

Boolean

Range

True, False

Example

```
Calibration.DisplayImageEnabled = True
```

DisplayMode

Description

DisplayMode controls which information is shown in the display.

Syntax

control.**DisplayMode** [= *Value* As hsCalibrationDisplayMode]

Type

hsCalibrationDisplayMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsDots	0	The detect calibration and validation dots are displayed.
hslmageError	1	The image error vectors are displayed.
hsWorldError	2	The world error vectors are displayed.

Example

```
Dim Mode As hsCalibrationDisplayMode
Calibration.DisplayMode = hsDots
Mode = Calibration.DisplayMode
```

DotConformity

Description

The required conformity for a detected calibration or validation dot to be considered valid. During the dot detection process, an average form factor is computed, along with the standard deviation. With a **DotConformity** setting of 1.0, only dots with form factor values within the standard deviation from the average form factor will be accepted. Decreasing **DotConformity** will extend the range of acceptable form factors to up to 6 standard deviations when set to the minimum value of 0.5.

Syntax

control.**DotConformity** [= *Value* As Single]

Type

Single

Range

Minimum: 0.5
Maximum: 1.0

Example

```
Calibration.DotConformity = 0.75
Calibration.Detect
```

DotPitch

Description

The actual horizontal and vertical spacing of the dots in the calibration grid, in the world coordinate system. The same dot pitch value is used for the spacing of the dots in the validation grid. **DotPitch** is expressed in the calibrated units specified by the **UnitsLength** property.

Syntax

control.**DotPitch** [= *Value* As Single]

Type

Single

Range

Greater than zero.

Example

```
Calibration.UnitsLength = hsInch
Calibration.DotPitch = 0.1
Calibration.Solve
```

EdgeSensitivity

Description

The sensitivity of the edge detection algorithm used to locate the dots. Setting **EdgeSensitivity** to its minimum value of 0 will only accept dots with very high contrast. On the other hand, with the maximum setting of 99, dots with very weak contrast will be accepted. The default setting of 60 usually gives the best trade-off between sensitivity to medium contrast dots and noise rejection. **EdgeSensitivity** can be slightly increased when some dots are not detected properly.

Syntax

control.**EdgeSensitivity** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0
Maximum: 99

Example

```
Calibration.EdgeSensitivity = 70
Calibration.Detect
```

MaximumDotRadius

Description

The maximum radius for a detected calibration or validation dot to be considered valid. **MaximumDotRadius** is expressed in pixels.

Syntax

control.**MaximumDotRadius** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1
Maximum: 100

Example

```
Calibration.MinimumDotRadius = 30
Calibration.MaximumDotRadius = 30
Calibration.Detect
```

MaximumImageError

Description

Maximum error between the positions of the detected calibration and validation dots and their corresponding world coordinates projected in the image using the calibration.

MaximumImageError is expressed in pixels. This property is available only after the calibration system is solved. Read only.

Syntax

control.**MaximumImageError**

Type

Single

Range

Greater than zero.

Example

```
Dim MaximumError As Single
Calibration.Solve
MaximumError = Calibration.MaximumImageError
```

MaximumWorldError

Description

Maximum error between the actual world positions of the calibration and validation dots and their corresponding detected image coordinates projected in the world coordinate system using the calibration. **MaximumWorldError** is expressed in the calibrated units specified by the **UnitsLength** property. This property is available only after the calibration system is solved. Read only.

Syntax

control.**MaximumWorldError**

Type

Single

Range

Greater than zero.

Example

```
Dim MaximumError As Single
Calibration.Solve
MaximumError = Calibration.MaximumWorldError
```

MinimumDotRadius

Description

The minimum radius for a detected calibration or validation dot to be considered valid.

MinimumDotRadius is expressed in pixels.

Syntax

control.**MinimumDotRadius** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 100

Example

```
Calibration.MinimumDotRadius = 30
Calibration.MaximumDotRadius = 30
Calibration.Detect
```

OriginPositionX

Description

The approximate X position in the image of the origin of the world coordinate system. Along with the **AxisXPositionX**, **AxisXPositionY**, **AxisYPositionX**, **AxisYPositionY**, and **OriginPositionY** properties, this property is used to identify the approximate position of the 3 dots defining the world coordinate system before calling a target detection. Once the target is detected, the value of **OriginPositionX** is automatically modified to reflect the actual X position of the first dot detected along the X-axis of the world coordinate system.

OriginPositionX is expressed in pixels.

Syntax

control.OriginPositionX [= Value As Single]

Type

Single

Range

Minimum: 0.

Maximum: width of the image.

Example

```
Calibration.OriginPositionX = 30
Calibration.OriginPositionY = 30
Calibration.AxisXPositionX = 60
Calibration.AxisXPositionY = 30
Calibration.AxisYPositionX = 30
Calibration.AxisYPositionY = 60
Calibration.Detect
```

OriginPositionXWorld

Description

The X offset between the origin of the world coordinate system used to detect the target and the actual origin of the world coordinate system. Along with the **OriginPositionYWorld** property, this property is used to set the actual position of the origin of the X-Y coordinate system used to determine the world location of each dot in the target detection process.

OriginPositionXWorld and **OriginPositionYWorld** are especially useful when cameras looking at different sections of the same target must be calibrated in the same coordinate system.

Syntax

control.OriginPositionXWorld [= Value As Single]

Type

Single

Range

Not applicable.

Example

```
Calibration.OriginPositionXWorld = 10
Calibration.OriginPositionYWorld = -10
Calibration.Solve
```

OriginPositionY

Description

The approximate Y position in the image of the origin of the world coordinate system. Along with the **AxisXPositionX**, **AxisXPositionY**, **AxisYPositionX**, **AxisYPositionY**, and **OriginPositionX** properties, this property is used to identify the approximate position of the 3 dots defining the world coordinate system before calling a target detection. Once the target is detected, the value of **OriginPositionY** is automatically modified to reflect the actual X position of the first dot detected along the X-axis of the world coordinate system.

OriginPositionY is expressed in pixels.

Syntax

control.OriginPositionY [= Value As Single]

Type

Single

Range

Minimum: 0.

Maximum: height of the image.

Example

```
Calibration.OriginPositionX = 30
Calibration.OriginPositionY = 30
Calibration.AxisXPositionX = 60
Calibration.AxisXPositionY = 30
Calibration.AxisYPositionX = 30
Calibration.AxisYPositionY = 60
Calibration.Detect
```

OriginPositionYWorld

Description

The X offset between the origin of the world coordinate system used to detect the target and the actual origin of the world coordinate system. Along with the **OriginPositionXWorld** property, this property is used to set the actual position of the origin of the X-Y coordinate system used to determine the world location of each dot in the target detection process.

OriginPositionXWorld and **OriginPositionYWorld** are especially useful when cameras looking at different sections of the same target must be calibrated in the same coordinate system.

Syntax

control.OriginPositionYWorld [= Value As Single]

Type

Single

Range

Not applicable.

Example

```
Calibration.OriginPositionXWorld = 10
Calibration.OriginPositionYWorld = -10
Calibration.Solve
```

UnitsLength

Description

Length unit used to express target dimensions.

Syntax

control.**UnitsLength** [= Value As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Calibration.UnitsLength = hsInch
Calibration.PositionXWorld(hsTopLeft) = 1.5
Calibration.PositionYWorld(hsTopLeft) = 12
Calibration.Solve
```

ValidationDotCount

Description

The number of validation dots detected in the image. Validation dots are used to validate the calibration parameters. Read only.

Syntax

control.**ValidationDotCount**

Type

Integer

Range

Minimum: 0

Maximum: Unlimited

Example

```
Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.ValidationDotCount - 1
    x = Calibration.ValidationDotPositionXImage(d)
    y = Calibration.ValidationDotPositionYImage(d)
Next d
```

ValidationDotPositionXImage

Description

The X position in the image of the validation dot selected with the **Index**.
ValidationDotPositionXImage parameter, expressed in pixels. Read only.

Syntax

control.**ValidationDotPositionXImage**(*Index* As Integer)

Type

Single

Range

Minimum: 0
Maximum: Width of the image.

Parameter(s)

Example

```
Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.ValidationDotCount - 1
    x = Calibration.ValidationDotPositionXImage(d)
    y = Calibration.ValidationDotPositionYImage(d)
Next d
```

ValidationDotPositionXWorld

Description

The X position in the world coordinate system of the validation dot selected with the **Index**.
ValidationDotPositionXWorld parameter, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**ValidationDotPositionXWorld**(*Index* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: ValidationDotCount - 1

Example

```

Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.ValidationDotCount - 1
    x = Calibration.ValidationDotPositionXWorld(d)
    y = Calibration.ValidationDotPositionYWorld(d)
Next d

```

ValidationDotPositionYImage**Description**

The Y position in the image of the validation dot selected with the **Index**. **ValidationDotPositionYImage** parameter, expressed in pixels. Read only.

Syntax

control.**ValidationDotPositionYImage**(*Index* As Integer)

Type

Single

Range

Minimum: 0

Maximum: height of the image

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: ValidationDotCount - 1

Example

```

Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.ValidationDotCount - 1
    x = Calibration.ValidationDotPositionXImage(d)
    y = Calibration.ValidationDotPositionYImage(d)
Next d

```

ValidationDotPositionYWorld

Description

The Y position in the world coordinate system of the validation dot selected with the **Index**. **ValidationDotPositionYWorld** parameter, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**ValidationDotPositionYWorld**(*Index* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the dot.	Integer	Minimum: 0 Maximum: ValidationDotCount - 1

Example

```
Dim d As Integer
Dim x As Single
Dim y As Single
For d = 0 To Calibration.ValidationDotCount - 1
    x = Calibration.ValidationDotPositionXWorld(d)
    y = Calibration.ValidationDotPositionYWorld(d)
Next d
```

WorldCoordinateSystemType

Description

Type of World coordinate system used to express coordinates in calibrated units. Can either be right-handed (default) or left-handed. In a right-handed World coordinate system, the Y-axis is 90 degrees counter clockwise from the X-axis. In a left-handed World coordinate system, the Y-axis is 90 degrees clockwise from the X-axis.

Syntax

control.**WorldCoordinateSystemType** [= *Value* As hsCoordinateSystemType]

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRightHanded	0	World coordinate is right-handed (Y-axis 90 degrees counter clockwise from the X-axis).
hsLeftHanded	1	World coordinate is left-handed (Y-axis 90 degrees clockwise from the X-axis).

Example

```
' Calibrate the system into a left-handed world
' reference frame
CalibrationInterface.WorldCoordinateSystemType = hsLeftHanded
```

Methods

Detect

Description

Acquires a new image and detects the target in this image. Solving the calibration does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax

control.**Detect**

Parameter(s)

None

Return

No return value.

Example

```
Calibration.Detect
```

EndDialog

Description

Terminates the calibration dialog. Depending on the value of the **Result** parameter, the calibration is either discarded or applied to the associated Acquisition Device Configuration.

Syntax

control.**EndDialog**(*Result* As hsCalibrationDialogResult)

Parameter(s)

Parameter	Description	Type	Range
<i>Result</i>	Specifies whether the calibration will be discarded or applied to the associated Acquisition Device Configuration.	hsCalibrationDialogResult	Enum. See table below.

hsCalibrationDialogResult	Value	Description
hsResultCancel	2	The calibration is discarded
hsResultOk	1	The calibration is applied to the associated Acquisition Device Configuration

Return

No return value.

Example

```
If DiscardCalibration = True Then
```

```

        Calibration.EndDialog hsResultCancel
    Else
        Calibration.EndDialog hsResultOk
    End If

```

GetDispatchPointer

Description

Reserved for internal use.

Hide

Description

Hides the calibration dialog. Unlike **EndDialog**, calling **Hide** hides the dialog but leaves it active without applying the calibration to the associated Acquisition Device Configuration. To terminate the dialog, **EndDialog** must be called.

Syntax

control.**Hide**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Hide**

Show

Description

Shows the calibration dialog. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameters settings can then be changed before making the dialog visible by calling **Show**.

Syntax

control.**Show**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Show**

Solve

Description

Solves the calibration. Calling **Solve** does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax

control.**Solve**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Solve**

Events

This object has no events.

HSCalibrationPerspectiveInterface

The HSCalibrationPerspectiveInterface control provides programmatic access to the Perspective calibration dialog of the Acquisition Device. It can be used to open the dialog and configure target detection and dimension parameters to propose default values customized for a specific application. Target detection and calibration system solving can also be executed programmatically. The calibration dialog can also be hidden so that the entire calibration procedure can be carried out without having to show the interactive user interface.

Properties

AveragePixelHeight

Description

Average calibrated height of a pixel in the image, expressed in the calibrated units specified by the property **UnitsLength**. This property is available only after the calibration system is solved. Read only.

Syntax

control.AveragePixelHeight

Type

Single

Range

Greater than zero.

Example

```
Dim h As Single
Calibration.Solve
h = Calibration.AveragePixelHeight
```

AveragePixelWidth

Description

Average calibrated width of a pixel in the image, expressed in the calibrated units specified by the property **UnitsLength**. This property is available only after the calibration system is solved. Read only.

Syntax

control.AveragePixelWidth

Type

Single

Range

Greater than zero.

Example

```
Dim w As Single
Calibration.Solve
```

w = Calibration.AveragePixelWidth

CornerApproximatePositionX

Description

The horizontal portion where the actual corner is expected to be located in the rectangle that bounds the corner detection operator selected with the **Index.CornerApproximatePositionX** parameter, expressed as a percentage of the width of the corner detector's bounding rectangle. It defines the location of a vertical line that bounds the expected location of the corner. When **CornerApproximatePositionX** is set to a value lower than 0.5, the actual corner is expected to be located in the left half of the rectangle. For example, with a value of 0.2, the corner is expected to be located in an area bounded by a rectangle which has a width of 20% of the total width of the corner detector and which has its left side aligned with the left side of the operator. When **CornerApproximatePositionX** is set to a value higher than 0.5 the actual corner is expected to be located in the right half of the rectangle. For example, with a value of 0.8, the corner is expected to be located in an area bounded by a rectangle which has a width of 20% of the total width of the corner detector and which has its right side aligned with the right side of the operator.

Syntax

control.CornerApproximatePositionX(Index As hsCalibrationCorner) [= Value As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

Calibration.CornerApproximatePositionX(hsTopLeft) = 0.2

Calibration.CornerApproximatePositionX(hsTopRight) = 0.8

Calibration.Detect

CornerApproximatePositionY

Description

The vertical portion where the actual corner is expected to be located in the rectangle that bounds the corner detection operator selected with the **Index.CornerApproximatePositionY**

parameter, expressed as a percentage of the height of the corner detector's bounding rectangle. It defines the location of a horizontal line that bounds the expected location of the corner. When **CornerApproximatePositionY** is set to a value lower than 0.5, the actual corner is expected to be located in the bottom half of the rectangle. For example, with a value of 0.2, the corner is expected to be located in an area bounded by a rectangle which has a height of 20% of the total height of the corner detector and which has its bottom side aligned with the bottom side of the operator. When **CornerApproximatePositionY** is set to a value higher than 0.5 the actual corner is expected to be located in the top half of the rectangle. For example, with a value of 0.8, the corner is expected to be located in an area bounded by a rectangle which has a height of 20% of the total height of the corner detector and which has its top side aligned with the top side of the operator.

Syntax

control.**CornerApproximatePositionY**(*Index* As hsCalibrationCorner) [= *Value* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

Calibration.**CornerApproximatePositionY**(hsBottomLeft) = 0.2

Calibration.**CornerApproximatePositionY**(hsTopLeft) = 0.8

Calibration.Detect

CornerDetectorHeight

Description

The height of the rectangle that bounds the corner detection operator selected with the **Index.CornerDetectorHeight** parameter, expressed in pixels.

Syntax

control.**CornerDetectorHeight**(*Index* As hsCalibrationCorner) [= *Value* As Integer]

Type

Integer

Range

Minimum: 1.

Maximum: height of the image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```
Calibration.CornerDetectorHeight(hsTopLeft) = 10
Calibration.Detect
```

CornerDetectorPositionX**Description**

X position of the center of the rectangle that bounds the corner detection operator selected with the **Index.CornerDetectorPositionX** parameter, expressed in pixels.

Syntax

```
control.CornerDetectorPositionX(Index As hsCalibrationCorner) [= Value As Integer]
```

Type

Integer

Range

Minimum: 0.

Maximum: width of the image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```
Calibration.CornerDetectorPositionX(hsTopLeft) = 50
Calibration.Detect
```

CornerDetectorPositionY

Description

Y position of the center of the rectangle that bounds the corner detection operator selected with the **Index.CornerDetectorPositionY** parameter, expressed in pixels.

Syntax

control.**CornerDetectorPositionY**(*Index* As hsCalibrationCorner) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: height of the image

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

Calibration.**CornerDetectorPositionY**(hsTopLeft) = 430

Calibration.Detect

CornerDetectorWidth

Description

The width of the rectangle that bounds the corner detection operator selected with the **Index.CornerDetectorWidth** parameter, expressed in pixels.

Syntax

control.**CornerDetectorWidth**(*Index* As hsCalibrationCorner) [= *Value* As Integer]

Type

Integer

Range

Minimum: 1.

Maximum: width of the image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```
Calibration.CornerDetectorWidth(hsTopLeft) = 10
Calibration.Detect
```

DetectionCount

Description

Number of images that are used to locate the 4 points on the target. When a value greater than 1 is set, 4 corner shaped calibration points are located for each image. Averaged positions are used to solve the calibration.

Syntax

```
control.DetectionCount [= Value As Integer]
```

Type

Integer

Range

Minimum: 1

Maximum: 100

Example

```
Dim i As Integer
Calibration.DetectionCount = 10
i = Calibration.DetectionCount
```

DetectionMode

Description

The mode used to locate the calibration points. In the automatic mode, the 4 corner shaped calibration points are automatically located with corner detectors. In the manual mode, the position of the 4 points in the image must be provided by external means using the properties **PositionXImage** and **PositionYImage**.

Syntax

```
control.DetectionMode [= Value As hsCalibrationDetectionMode]
```

Type

hsCalibrationDetectionMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsAutomatic	0	The 4 calibration points are automatically located in the image using corner detection operators.

Constant	Value	Description
hsManual	1	The position of the 4 calibration points in the image are provided by external means.

Example

```
Dim Mode As hsCalibrationDetectionMode
Calibration.DetectionMode = hsAutomatic
Mode = Calibration.DetectionMode
```

LockOnCornersEnabled

Description

When **LockOnCornersEnabled** is set to True, the corner detectors are automatically moved to the actual position of the corners after they are detected in the image. This mode can be used to track the target as it is moved slightly in the image between successive call to **Detect**.

Syntax

control.**LockOnCornersEnabled** [= Value As Boolean]

Type

Boolean

Range

True, False

Example

```
Calibration.LockOnCornersEnabled = True
Do
    Calibration.Detect
Loop While TargetAdjustLoop = True
```

PositionXImage

Description

X position in the image of the corner selected with the **Index** parameter. When **DetectionMode** is set to **hsAutomatic**, **PositionXImage** is read only and returns the actual X position of the detected corner. When **DetectionMode** is set to **hsManual**, **PositionXImage** is read write and sets the X position of the calibration point in the image. **PositionXImage** is expressed in pixels.

Syntax

control.**PositionXImage**(*Index* As hsCalibrationCorner) [= Value As Single]

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```

Dim x As Single
Calibration.Detect
x = Calibration.PositionXImage(hsTopLeft)
If x < 100 Then
    MsgBox "Invalid X Position for Top Left Point"
End If

```

PositionXWorld**Description**

Actual X position in the world coordinate system of the corner selected with the **Index.PositionXWorld** parameter, expressed in the calibrated units set by the property **UnitsLength**.

Syntax

control.**PositionXWorld**(*Index* As hsCalibrationCorner) [= *Value* As Single]

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```

Calibration.UnitsLength = hsInch
Calibration.PositionXWorld(hsTopLeft) = 1.5
Calibration.PositionYWorld(hsTopLeft) = 12
Calibration.Solve

```


PositionYImage

Description

Y position in the image of the corner selected with the **Index** parameter. When **DetectionMode** is set to **hsAutomatic**, **PositionYImage** is read only and returns the actual X position of the detected corner. When **DetectionMode** is set to **hsManual**, **PositionYImage** is read write and sets the X position of the calibration point in the image. **PositionYImage** is expressed in pixels.

Syntax

control.**PositionYImage**(*Index* As hsCalibrationCorner) [= *Value* As Single]

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```
Dim y As Single
Calibration.Detect
y = Calibration.PositionYImage(hsTopLeft)
If y > 380 Then
    MsgBox "Invalid Y Position for Top Left Point"
End If
```

PositionYWorld

Description

Actual Y position in the world coordinate system of the corner selected with the **Index.PositionYWorld** parameter, expressed in the calibrated units set by the property **UnitsLength**.

Syntax

control.**PositionYWorld**(*Index* As hsCalibrationCorner) [= *Value* As Single]

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the corner.	hsCalibrationCorner	Enum, see table below.

hsCalibrationCorner	Value	Description
hsBottomLeft	0	The bottom left corner (point 0).
hsTopLeft	1	The top left corner (point 1).
hsTopRight	2	The top right corner (point 2).
hsBottomRight	3	The bottom right corner (point 3).

Example

```
Calibration.UnitsLength = hsInch
Calibration.PositionXWorld(hsTopLeft) = 1.5
Calibration.PositionYWorld(hsTopLeft) = 12
Calibration.Solve
```

UnitsLength**Description**

Length unit used to express target dimensions.

Syntax

control.UnitsLength [= Value As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Calibration.UnitsLength = hsInch
Calibration.PositionXWorld(hsTopLeft) = 1.5
Calibration.PositionYWorld(hsTopLeft) = 12
```

Calibration.Solve

WorldCoordinateSystemType

Description

Type of World coordinate system used to express coordinates in calibrated units. Can either be right-handed (default) or left-handed. In a right-handed World coordinate system, the Y-axis is 90 degrees counter clockwise from the X-axis. In a left-handed World coordinate system, the Y-axis is 90 degrees clockwise from the X-axis.

Syntax

control.**WorldCoordinateSystemType** [= *Value* As hsCoordinateSystemType]

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRightHanded	0	World coordinate is right-handed (Y-axis 90 degrees counter clockwise from the X-axis).
hsLeftHanded	1	World coordinate is left-handed (Y-axis 90 degrees clockwise from the X-axis).

Example

```
' Calibrate the system into a left-handed world
' reference frame
CalibrationInterface.WorldCoordinateSystemType = hsLeftHanded
```

Methods

Detect

Description

Acquires a new image and detects the target in this image. Solving the calibration does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax

control.**Detect**

Parameter(s)

None

Return

No return value.

Example

```
Calibration.Detect
```

EndDialog

Description

Terminates the calibration dialog. Depending on the value of the **Result** parameter, the calibration is either discarded or applied to the associated Acquisition Device Configuration.

Syntax

control.**EndDialog**(*Result* As hsCalibrationDialogResult)

Parameter(s)

Parameter	Description	Type	Range
<i>Result</i>	Specifies whether the calibration will be discarded or applied to the associated Acquisition Device Configuration.	hsCalibrationDialogResult	Enum. See table below.

hsCalibrationDialogResult	Value	Description
hsResultCancel	2	The calibration is discarded.
hsResultOk	1	The calibration is applied to the associated Acquisition Device Configuration.

Return

No return value.

Example

```
If DiscardCalibration = True Then
    Calibration.EndDialog hsResultCancel
Else
    Calibration.EndDialog hsResultOk
End If
```

GetDispatchPointer

Description

Reserved for internal use.

Hide

Description

Hides the calibration dialog. Unlike **EndDialog**, calling **Hide** hides the dialog but leaves it active without applying the calibration to the associated Acquisition Device Configuration. To terminate the dialog, **EndDialog** must be called.

Syntax

control.**Hide**

Parameter(s)

None

Return

No return value.

Example

```
Calibration.Hide
```

Show

Description

Shows the calibration dialog. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameter settings can then be changed before making the dialog visible by calling **Show**.

Syntax

control.**Show**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Show**

Solve

Description

Solves the calibration. Calling **Solve** does not implicitly call the target detection method.

Detect must therefore be called explicitly before solving the calibration.

Syntax

control.**Solve**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Solve**

Events

This object has no events.

HSCalibrationXYScaleInterface

Description

The HSCalibrationXYScaleInterface control provides programmatic access to the XY Scale calibration dialog of the Acquisition Device. It can be used to open the dialog and configure target detection and dimension parameters to propose default values customized for a specific application. Target detection and calibration system solving can also be executed programmatically. The calibration dialog can also be hidden so that the entire calibration procedure can be carried out without having to show the interactive user interface.

Properties

AveragePixelHeight

Description

Average calibrated height of a pixel in the image, expressed in the calibrated units specified by the **UnitsLength** property. This property is available only after the calibration system is solved. Read only.

Syntax

control.AveragePixelHeight

Type

Single

Range

Greater than zero.

Example

```
Dim h As Single
Calibration.Solve
h = Calibration.AveragePixelHeight
```

AveragePixelWidth

Description

Average calibrated width of a pixel in the image, expressed in the calibrated units specified by the **UnitsLength** property. This property is available only after the calibration system is solved. Read only.

Syntax

control.AveragePixelWidth

Type

Single

Range

Greater than zero.

Example

```
Dim w As Single
```

```
Calibration.Solve
w = Calibration.AveragePixelWidth
```

DetectionCount

Description

Number of images that are used to measure the target. When a value greater than 1 is set, target width and height are measured for each image. Averaged measurements are used to solve the calibration.

Syntax

```
control.DetectionCount [= Value As Integer]
```

Type

Integer

Range

Minimum: 1
Maximum: 100

Example

```
Dim i As Integer
Calibration.DetectionCount = 10
i = Calibration.DetectionCount
```

DetectionMode

Description

The mode used to measure the target. In the automatic mode, the target is automatically measured with horizontal and vertical calipers. In the manual mode, target width and height in the image must be provided by external means using the **WidthImage** and **HeightImage** properties.

Syntax

```
control.DetectionMode [= Value As hsCalibrationDetectionMode]
```

Type

hsCalibrationDetectionMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsAutomatic	0	Target height and width automatically measured in the image using horizontal and vertical calipers.
hsManual	1	Target height and width in the image provided by external means.

Example

```
Dim Mode As hsCalibrationDetectionMode
Calibration.DetectionMode = hsAutomatic
Mode = Calibration.DetectionMode
```

HeightImage

Description

Height of the target in the image. When **DetectionMode** is set to **hsAutomatic**, **HeightImage** is read only and returns the measurement of the vertical caliper. When **DetectionMode** is set to **hsManual**, **HeightImage** is read write and sets the height of the target in the image. **HeightImage** is expressed in pixels.

Syntax

control.HeightImage [= Value As Single]

Type

Single

Range

Greater than zero.

Example

```
Dim h As Single
Calibration.Detect
h = Calibration.HeightImage
If h > 100 Then
    MsgBox "Invalid Target Height"
End If
```

HeightWorld

Description

Actual height of the target in the world coordinate system, expressed in the calibrated units set by the **UnitsLength** property.

Syntax

control.HeightWorld [= Value As Single]

Type

Single

Range

Greater than zero.

Example

```
Calibration.UnitsLength = hsInch
Calibration.HeightWorld = 10
Calibration.Solve
```

HorizontalCaliperHeight

Description

Thickness of the horizontal caliper used to measure the width of the target. **HorizontalCaliperHeight** actually adjust the length of the projection of the caliper, that is, the number of pixels that are averaged to build the one-dimensional signal used to detect edges. **HorizontalCaliperHeight** is expressed in pixels.

Syntax

control.HorizontalCaliperHeight [= Value As Integer]

Type

Integer

Range

Minimum: 1.

Maximum: height of the image.

Example

```
Calibration.HorizontalCaliperHeight = 10
```

```
Calibration.Detect
```

HorizontalCaliperPositionX

Description

X position of the center of the horizontal caliper in the image. **HorizontalCaliperPositionX** is expressed in pixels.

Syntax

```
control.HorizontalCaliperPositionX [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: width of the image

Example

```
Calibration.HorizontalCaliperPositionX = 320
```

```
Calibration.Detect
```

HorizontalCaliperPositionY

Description

Y position of the center of the horizontal caliper in the image. **HorizontalCaliperPositionY** is expressed in pixels.

Syntax

```
control.HorizontalCaliperPositionY [= Value As Integer]
```

Type

Integer

Range

Minimum: 0.

Maximum: height of the image.

Example

```
Calibration.HorizontalCaliperPositionY = 240
```

```
Calibration.Detect
```

HorizontalCaliperWidth

Description

Length of the horizontal caliper used to measure the width of the target.

HorizontalCaliperWidth should be adjusted so that the horizontal caliper exceeds the boundaries of the target by at least 10 pixels on both sides. The two outermost edges that are located are used to compute the width of the target. Edge polarity is ignored.

HorizontalCaliperWidth is expressed in pixels.

Syntax

```
control.HorizontalCaliperWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: 1.

Maximum: width of the image.

Example

```
Calibration.HorizontalCaliperWidth = 400
```

```
Calibration.Detect
```

UnitsLength

Description

Length unit used to express target dimensions.

Syntax

```
control.UnitsLength [= Value As hsUnitsLength]
```

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Calibration.UnitsLength = hsInch
```

```
Calibration.HeightWorld = 15
```

```
Calibration.WidthWorld = 20
```

```
Calibration.Solve
```

VerticalCaliperHeight

Description

Thickness of the vertical caliper used to measure the height of the target.

VerticalCaliperHeight actually adjust the length of the projection of the caliper, that is, the number of pixels that are averaged to build the one-dimensional signal used to detect edges.

VerticalCaliperHeight is expressed in pixels.

Syntax

`control.VerticalCaliperHeight [= Value As Integer]`

Type

Integer

Range

Minimum: 1.

Maximum: width of the image.

Example

`Calibration.VerticalCaliperHeight = 10`

`Calibration.Detect`

VerticalCaliperPositionX

Description

X position of the center of the vertical caliper in the image. **VerticalCaliperPositionX** is expressed in pixels.

Syntax

`control.VerticalCaliperPositionX [= Value As Integer]`

Type

Integer

Range

Minimum: 0.

Maximum: width of the image.

Example

`Calibration.VerticalCaliperPositionX = 320`

`Calibration.Detect`

VerticalCaliperPositionY

Description

Y position of the center of the vertical caliper in the image. **VerticalCaliperPositionY** is expressed in pixels.

Syntax

`control.VerticalCaliperPositionY [= Value As Integer]`

Type

Integer

Range

Minimum: 0

Maximum: height of the image

Example

`Calibration.VerticalCaliperPositionY = 240`

`Calibration.Detect`

VerticalCaliperWidth

Description

Length of the vertical caliper used to measure the height of the target. **VerticalCaliperWidth** should be adjusted so that the vertical caliper exceeds the boundaries of the target by at least

10 pixels on both sides. The two outermost edges that are located are used to compute the height of the target. Edge polarity is ignored. **VerticalCaliperWidth** is expressed in pixels.

Syntax

control.**VerticalCaliperWidth** [= Value As Integer]

Type

Integer

Range

Minimum: 1

Maximum: height of the image.

Example

```
Calibration.VerticalCaliperWidth = 400
Calibration.Detect
```

WidthImage

Description

Width of the target in the image. When **DetectionMode** is set to **hsAutomatic**, **WidthImage** is read only and returns the measurement of the horizontal caliper. When **DetectionMode** is set to **hsManual**, **WidthImage** is read write and sets the width of the target in the image. **WidthImage** is expressed in pixels.

Syntax

control.**WidthImage** [= Value As Single]

Type

Single

Range

Greater than zero.

Example

```
Dim w As Single
Calibration.Detect
w = Calibration.WidthImage
If w > 100 Then
    MsgBox "Invalid Target Width"
End If
```

WidthWorld

Description

Actual width of the target in the world coordinate system, expressed in the calibrated units set by the **UnitsLength** property.

Syntax

control.**WidthWorld** [= Value As Single]

Type

Single

Range

Greater than zero.

Example

```
Calibration.UnitsLength = hsInch
```

```
Calibration.WidthWorld = 10
Calibration.Solve
```

WorldCoordinateSystemType

Description

Type of World coordinate system used to express coordinates in calibrated units. Can either be right-handed (default) or left-handed. In a right-handed World coordinate system, the Y-axis is 90 degrees counter clockwise from the X-axis. In a left-handed World coordinate system, the Y-axis is 90 degrees clockwise from the X-axis.

Syntax

```
control.WorldCoordinateSystemType [= Value As hsCoordinateSystemType]
```

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRightHanded	0	World coordinate is right-handed (Y-axis 90 degrees counter clockwise from the X-axis).
hsLeftHanded	1	World coordinate is left-handed (Y-axis 90 degrees clockwise from the X-axis).

Example

```
' Calibrate the system into a left-handed world
' reference frame
CalibrationInterface.WorldCoordinateSystemType = hsLeftHanded
```

Methods

Detect

Description

Acquires a new image and detects the target in this image. Solving the calibration does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax

```
control.Detect
```

Parameter(s)

None

Return

No return value.

Example

```
Calibration.Detect
```

EndDialog

Description

Terminates the calibration dialog. Depending on the value of the **Result** parameter, the calibration is either discarded or applied to the associated Acquisition Device Configuration.

Syntax

control.**EndDialog**(*Result* As hsCalibrationDialogResult)

Parameter(s)

Parameter	Description	Type	Range
<i>Result</i>	Specifies whether the calibration will be discarded or applied to the associated Acquisition Device Configuration.	hsCalibrationDialogResult	Enum. See table below.

hsCalibrationDialogResult	Value	Description
hsResultCancel	2	The calibration is discarded
hsResultOk	1	The calibration is applied to the associated Acquisition Device Configuration

Return

No return value.

Example

```
If DiscardCalibration = True Then
    Calibration.EndDialog hsResultCancel
Else
    Calibration.EndDialog hsResultOk
End If
```

GetDispatchPointer

Description

Reserved for internal use.

Hide

Description

Hides the calibration dialog. Unlike **EndDialog**, calling **Hide** hides the dialog but leaves it active without applying the calibration to the associated Acquisition Device Configuration. To terminate the dialog, **EndDialog** must be called.

Syntax

control.**Hide**

Parameter(s)

None

Return

No return value.

Example

```
Calibration.Hide
```

Show

Description

Shows the calibration dialog. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameters settings can then be changed before making the dialog visible by calling **Show**.

Syntax

control.**Show**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Show**

Solve

Description

Solves the calibration. Calling **Solve** does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax

control.**Solve**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Solve**

Events

This object has no events.

HSColorCalibrationFCMInterface

Description

The HSColorCalibrationFCMInterface control provides programmatic access to the color calibration dialog of the Acquisition Device. It can be used to open the dialog and execute target detection and calibration system solving. The calibration dialog can also be hidden so that the entire calibration procedure can be carried out without having to show the interactive user interface.

Properties

This object has no properties.

Methods

Detect

Description

Acquires a new image and detects the target in this image. Solving the calibration does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax

control.**Detect**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Detect**

EndDialog

Description

Terminates the calibration dialog. Depending on the value of the **Result** parameter, the calibration is either discarded or applied to the associated Acquisition Device Configuration.

Syntax

control.**EndDialog**(*Result* As hsCalibrationDialogResult)

Parameter(s)

Parameter	Description	Type	Range
<i>Result</i>	Specifies whether the calibration will be discarded or applied to the associated Acquisition Device Configuration.	hsCalibrationDialogResult	Enum. See table below.

hsCalibrationDialogResult	Value	Description
hsResultCancel	2	The calibration is discarded
hsResultOk	1	The calibration is applied to the associated Acquisition Device Configuration

Return

No return value.

Example

```
If DiscardCalibration = True Then
    Calibration.EndDialog hsResultCancel
Else
    Calibration.EndDialog hsResultOk
End If
```

GetDispatchPointer

Description

Reserved for internal use.

Hide

Description

Hides the calibration dialog. Unlike **EndDialog**, calling **Hide** hides the dialog but leaves it active without applying the calibration to the associated Acquisition Device Configuration. To terminate the dialog, **EndDialog** must be called.

Syntax

control.**Hide**

Parameter(s)

None

Return

No return value.

Example

```
Calibration.Hide
```

Show

Description

Shows the calibration dialog. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameters settings can then be changed before making the dialog visible by calling **Show**.

Syntax*control.Show***Parameter(s)**

None

Return

No return value.

ExampleCalibration.**Show**

ShowModal

Description

Shows the calibration dialog as modal. Modal behavior can only be achieved by creating the dialog as invisible and by calling **ShowModal** on the object.

Syntax*control.ShowModal***Parameter(s)**

None

Return

No return value.

ExampleCalibration.**ShowModal**

Solve

Description

Solves the calibration. Calling **Solve** does not implicitly call the target detection method. **Detect** must therefore be called explicitly before solving the calibration.

Syntax*control.Solve***Parameter(s)**

None

Return

No return value.

ExampleCalibration.**Solve**

Events

This object has no events.

Application

Description

The Application Control is the basis of a HexSight vision application. It manages data through the HexSight Explorer and Process Manager controls.

Properties

Database

Description

Reserved for internal use.

Description

Description

String that describes the standard usage of the control. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Application1.Description
```

DisplayMessageOnError

Description

Specifies if error messages are automatically displayed when an error occurs.

Syntax

control.**DisplayMessageOnError** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	Error messages are not displayed automatically when an error occurs.
True	Error messages are displayed automatically when an error occurs.

Example

```
Dim Displayed As Boolean
Displayed = Application1.DisplayMessageOnError
```

```
Application1.DisplayMessageOnError = False
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

```
control.ErrorObject
```

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

Explorer

Description

Reserved for internal use.

ExplorerTabVisible

Description

Specifies if the Explorer tab in the control interface is visible.

Syntax

```
control.ExplorerTabVisible [= Value As Boolean ]
```

Type

Boolean

Range

True: the Explorer tab is visible,
False: the Explorer tab is not visible.

Example

```
Dim Visible As Boolean  
Visible = Application1.ExplorerTabVisible  
Application1.ExplorerTabVisible = True
```

InterfaceVisible

Description

Specifies if the user interface of the control is displayed on its parent form.

Syntax

```
control.InterfaceVisible [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	The interface is displayed
False	The interface is not displayed

Example

```
Dim Visible As Boolean
Visible = Application1.InterfaceVisible
Application1.InterfaceVisible = False
```

Name**Description**

Name used to access the control.

Syntax*control*.**Name** [= *Value* As String]**Type**

String

Range

Not applicable.

Example

```
Dim CtrlName As String
CtrlName = Application1.Name = "ApplicA"
Application1.Name = "ApplicA"
```

ProcessManager**Description**

Reference to the HSProcessManager object contained in the control. Read only.

Syntax*control*.**ProcessManager****Type**

HSProcessManager

Range

Not applicable.

Example

```
Dim MyManager As HSProcessManager
Set MyManager = Application1.ProcessManager
```

ProcessManagerTabVisible**Description**

Specifies if the Process Manager tab in the control interface is visible.

Syntax*control*.**ProcessManagerTabVisible** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The Process Manager tab is visible.
False	The Process Manager tab is not visible.

Example

```
Dim Visible As Boolean
Visible = Application1.ProcessManagerTabVisible
Application1.ProcessManagerTabVisible = True
```

UpdateDisplayOnDone

Description

Specifies if the display created interactively using the Explorer control is automatically updated when the execution of a sequence of process is completed.

Syntax

*control.***UpdateDisplayOnDone** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Explorer displays are automatically updated when execution is completed.
False	Explorer displays are not automatically updated.

Example

```
Dim AutoUpdate As Boolean
AutoUpdate = Application1.UpdateDisplayOnDone
Application1.UpdateDisplayOnDone = False
```

UpdateExplorerOnDatabaseChange

Description

Specifies if the interface of the Explorer is automatically updated on database changes after the execution of a sequence of processes.

Syntax

*control.***UpdateExplorerOnDatabaseChange** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The HSExplorer control is automatically updated when a database change occurs.

Value	Description
False	The HSExplorer control is not automatically updated.

Example

```
Dim AutoUpdate As Boolean
AutoUpdate = Application1.UpdateExplorerOnDatabaseChange
Application1.UpdateExplorerOnDatabaseChange = True
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.1.000 R 2100(release)".

Example

```
Dim Version As String
Version = Application1.Version
```

Methods

AboutBox

Description

Displays the Caliper tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Example

```
Caliper1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetDebugParameter

Description

Reserved for internal use.

HideInterface

Description

Hides the user interface of the control.

Syntax

control.**HideInterface**() As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim HasSucceeded As Boolean
HasSucceeded = Application1.HideInterface
```

HideProperties

Description

Hides the properties interface of the control.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Application1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

Refresh

Description

Refreshes the user interface of the Application control.

Syntax

control.**Refresh**

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Example

Application1.**Refresh**

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowInterface

Description

Displays the user interface of the Application tool.

Syntax

control.**ShowInterface**(*Modal* As Boolean, [*DisplayTopMost* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	True, False
<i>DisplayTopMost</i>	When True the Application interface window is displayed topmost. Optional.	Variant	True, False

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Dim InterfaceDisplayed As Boolean
InterfaceDisplayed = Application1.ShowInterface (False)
```

ShowProperties

Description

Displays the control properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeeded As Boolean
Succeeded = Application1.ShowProperties(True)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Occurs when the runtime database has changed (a View, an Image and/or a Scene has been added, removed or modified by a process).

Syntax

Sub *control*.**DatabaseChange**(*ProcessName* As String)

Parameter(s)

Parameter	Description	Type	Range
<i>ProcessName</i>	Name of the process that has modified the database.	String	Not applicable.

Done

Description

Occurs when the execution of one control is completed.

Syntax

Sub *control*.**Done**(*ProcessName* As String)

Parameter(s)

Parameter	Description	Type	Range
<i>ProcessName</i>	Name of the process that notifies the Application control.	String	Not applicable.

Error**Description**

Reserved for internal use.

ExplorerSelectionChange**Description**

Reserved for internal use.

ProcessManagerModeChange**Description**

Reserved for internal use.

PropertiesChange**Description**

Occurs each time a property is changed interactively using the property page of a process.
Updates the user interface according to the new parameters.

Syntax

Sub *control*.**PropertiesChange**(*ProcessName* as String)

Parameter(s)

Parameter	Description	Type	Range
<i>ProcessName</i>	Name of the modified process.	String	Not applicable.

ArcCaliper

Description

The Arc Caliper measures distances between radial or annular edge pairs on arc-shaped objects. Users can select edge pairs of interest by setting Arc Caliper properties such as constraints, thresholds and polarities.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the input image. By default, bilinear interpolation is enabled because it ensures subpixel accuracy

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

The valid range for this property is as follows:

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
Result = ArcCaliper1.BilinearInterpolationEnabled
ArcCaliper2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = ArcCaliper1.CalibratedUnitsEnabled
ArcCaliper2.CalibratedUnitsEnabled = False
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim CurrentSystem As hsCoordinateSystem
```

```
' Get the current coordinate system
CurrentSystem = ArcCaliper1.CoordinateSystem
' Express the results in the Object coordinate system
ArcCaliper1.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = ArcCaliper1.Description
```

Edge1Constraints

Description

Bitwise field used to select the detection constraints for the first edge of the selected pair. Constraints can be set for position and/or magnitude and are used to score edges.

Syntax

control.**Edge1Constraints** (*Pair* As Variant) [= *Value* as hsConstraints]

Type

hsConstraints

Range

The valid range for this property is as follows:

Constant	Value	Description
hsNone	0	No constraint
hsPosition	1	Position constraint
hsMagnitude	2	Magnitude constraint
hsAllConstraints	3	Magnitude and position constraints

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair. Other types not supported.

Example

```
Dim EdgeCt As hsConstraints
EdgeCt = ArcCaliper1.Edge1Constraints("Pair0")
ArcCaliper2.Edge1Constraints(0) = hsPosition
```

Edge1Magnitude

Description

Magnitude of the first edge of the selected pair. Read only.

Syntax

control.**Edge1Magnitude**(Pair As Variant)

Type

Single

Range

Minimum: -255

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair. Other types not supported.

Example

```
Dim EdgeMag As Single
EdgeMag = ArcCaliper1.Edge1Magnitude(0)
EdgeMag = ArcCaliper2.Edge1Magnitude("Pair0")
```

Edge1MagnitudeConstraint

Description

Indexed property used to set the magnitude constraint function of the first edge of the selected pair. Two points are used: Base and Top.

Syntax

control.**Edge1MagnitudeConstraint** (*Pair* as Variant, *Index* As hsMagnitudeConstraintIndex) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the two points of the magnitude constraint function	hsMagnitudeConstraint Index	hsBase: 1 hsTop: 2

Example

```
Dim MagCt As Integer
MagCt = ArcCaliper1.Edge1MagnitudeConstraint(0,hsBase) = 9
ArcCaliper2.Edge1MagnitudeConstraint("Pair0",2) = 60
```

Edge1MagnitudeScore

Description

Magnitude score of the first edge of the selected pair. Read only.

Syntax*control*.**Edge1MagnitudeScore** (*Pair* As Variant)**Type**

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair. Other types not supported.

Example

```
Dim Score As Single
Score = ArcCaliper1.Edge1MagnitudeScore ("Pair0")
Score = ArcCaliper2.Edge1MagnitudeScore ( 0 )
```


Edge1Polarity

Description

Selection criterion of the first edge of the selected pair. The grey-scale transition of the edge must respect the polarity set by this property.

Syntax

control.**Edge1Polarity**(*Pair* As Variant) [= *Value* As hsPolarity]

Type

hsPolarity

Range

The valid range for this property is as follows:

Name	Value	Description
hsDarkToLight	0	The greylevel value must go from dark to light when crossing an edge.
hsLightToDark	1	The greylevel value must go from light to dark when crossing an edge.
hsEitherPolarity	2	The change in greylevel value change is not a criterion for locating an edge.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair. Other types not supported.

Example

```
Dim Pol As hsPolarity
Pol = ArcCaliper1.Edge1Polarity ("Pair0")
ArcCaliper2.Edge1Polarity ( 0 ) = hsEitherPolarity
```

Edge1PositionConstraint

Description

Indexed property used to set the position constraint function of the first edge of the selected pair. Four points are used: Base Left, Top Left, Top Right, Base Right.

Syntax

control.**Edge1PositionConstraint** (*Pair* As Variant, *Index* As hsPositionConstraintIndex) [= *Value* As Integer]

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the four points of the position constraint function	hsPositionConstraintIndex	hsBaseLeft: 1 hsTopLeft: 2 hsTopRight: 3 hsBaseRight: 4

Remark(s)

A position constraint value of 0.0 represents the left of the tool while 1.0 represents the right of the tool.

Example

```
Dim PosCt As Single
PosCt = ArcCaliper1.Edge1PositionConstraint(0,hsBaseLeft)
ArcCaliper2.Edge1PositionConstraint("Pair0",hsTopLeft) = 0.8
```

Edge1PositionScore

Description

Position score of the first edge of the selected pair. Read only.

Syntax

control.**Edge1PositionScore**(*Pair* as Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = ArcCaliper1.Edge1PositionScore(0)
Score = ArcCaliper1.Edge1PositionScore("Pair0")
```

Edge1PositionX

Description

X coordinate of the center of the first edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge1PositionX**(*Pair* As Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = ArcCaliper1.Edge1PositionX(0)
X_Pos = ArcCaliper1.Edge1PositionX("Pair0")
```

Edge1PositionY

Description

Y coordinate of the center of the first edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge1PositionY**(*Pair* As Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim Y_Pos As Single
Y_Pos = ArcCaliper1.Edge1PositionY(0)
Y_Pos = ArcCaliper1.Edge1PositionY("Pair0")
```

Edge1Radius

Description

Radius of the first edge of the selected pair, **ToolPositionX** and **ToolPositionY** being the center of the circular arc described by the said edge. Read only.

Syntax

control.**Edge1Radius**(*Pair* As Variant)

Type

Single

Range

Greater than zero.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The radius is expressed in units specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False	Pixel
CalibratedUnitsEnabled = True	UnitsLength property

Example

```
Dim CircDist As Single
CircDist = ArcCaliper1.Edge1Radius("Pair0")
```

Edge1Rotation

Description

Rotation of the first edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge1Rotation**(*Pair* as Variant)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	Edge1Rotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation As Single
Rotation = ArcCaliper1.Edge1Rotation(0)
Rotation = ArcCaliper1.Edge1Rotation("Pair0")
```

Edge1Score

Description

Score of the first edge of the selected pair. The score is computed according to the constraints set by the **Edge1Constraints** property. Read only.

Syntax

```
control.Edge1Score(Pair As Variant)
```

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = ArcCaliper1.Edge1Score(0)
Score = ArcCaliper1.Edge1Score("Pair0")
```

Edge1ScoreThreshold

Description

Minimum score to accept an edge as the first edge of the selected pair. The score of the first edge is returned by the **Edge1Score** property.

Syntax

control.**Edge1ScoreThreshold**(*Pair* As Variant) [= *Value* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim ScoreTh As Single
ScoreTh = ArcCaliper1.Edge1ScoreThreshold(1)
ArcCaliper.Edge1ScoreThreshold("Pair0") = 0.75
```

Edge2Constraints

Description

Bitwise field used to select the detection constraints for the second edge of the selected pair. Constraints can be set for position and/or magnitude and are used to score edges.

Syntax

control.**Edge2Constraints** (*Pair* As Variant) [= *Value* as hsConstraints]

Type

hsConstraints

Range

The valid range for this property is as follows:

Constant	Value	Description
hsNone	0	No constraint
hsPosition	1	Position constraint
hsMagnitude	2	Magnitude constraint

Constant	Value	Description
hsAllConstraints	3	Magnitude and position constraints

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim EdgeCt As hsConstraints
EdgeCt = ArcCaliper1.Edge2Constraints("Pair0")
ArcCaliper2.Edge2Constraints(0) = hsPosition
```

Edge2Magnitude

Description

Magnitude of the second edge of the selected pair. Read only.

Syntax

control.**Edge2Magnitude**(*Pair* As Variant)

Type

Single

Range

Minimum: -255

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim EdgeMag As Single
EdgeMag = ArcCaliper1.Edge2Magnitude(0)
EdgeMag = ArcCaliper2.Edge2Magnitude("Pair0")
```

Edge2MagnitudeConstraint

Description

Indexed property used to set the magnitude constraint function of the second edge of the selected pair. Two points are used: Base and Top.

Syntax

control.**Edge2MagnitudeConstraint** (*Pair* As Variant, *Index* As hsMagnitudeConstraintIndex) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the two points of the magnitude constraint function	hsMagnitudeConstraintIndex	hsBase: 1 hsTop: 2

Example

```
Dim MagCt As Integer
```

```
MagCt = ArcCaliper1.Edge2MagnitudeConstraint(0,hsBase) = 9
```

```
ArcCaliper2.Edge2MagnitudeConstraint("Pair0",2) = 60
```

Edge2MagnitudeScore

Description

Magnitude score of the second edge of the selected pair. Read only.

Syntax

control.**Edge2MagnitudeScore** (*Pair* As Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
```

```
Score = ArcCaliper1.Edge2MagnitudeScore("Pair0")
```


Score = ArcCaliper2.**Edge2MagnitudeScore**(0)

Edge2Polarity

Description

Validation criterion of the second edge of the selected pair. The grey-scale transition for the second edge must respect the polarity set by this property.

Syntax

control.**Edge2Polarity**(*Pair* As Variant) [= *Value* As hsPolarity]

Type

hsPolarity

Range

The valid range for this property is as follows:

Name	Value	Description
hsDarkToLight	0	The greylevel value must go from dark to light when crossing an edge.
hsLightToDark	1	The greylevel value must go from light to dark when crossing an edge.
hsEitherPolarity	2	The change in greylevel value change is not a criterion for locating an edge.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Pol As hsPolarity
Pol = ArcCaliper1.Edge2Polarity("Pair0")
ArcCaliper2.Edge2Polarity(0) = hsAllSlopes
```

Edge2PositionConstraint

Description

Indexed property used to set the position constraint function of the second edge of the selected pair. Four points are used: Base Left, Top Left, Top Right, Base Right.

Syntax

control.**Edge2PositionConstraint** (*Pair* As Variant, *Index* As hsPositionConstraintIndex) [= *Value* As Integer]

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the four points of the position constraint function.	hsPositionConstraintIndex	hsBaseLeft: 1 hsTopLeft: 2 hsTopright: 3 hsBaseRight: 4

Remark(s)

A position constraint value of 0.0 represents the left of the tool while 1.0 represents the right of the tool.

Example

```
Dim PosCt As Single
PosCt = ArcCaliper1.Edge2PositionConstraint(0,hsBaseLeft)
ArcCaliper2.Edge2PositionConstraint("Pair0",hsTopLeft) = 0.8
```

Edge2PositionScore**Description**

Position score of the second edge of the selected pair. Read only.

Syntax

control.**Edge2PositionScore**(*Pair* as Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = ArcCaliper1.Edge2PositionScore(0)
Score = ArcCaliper1.Edge2PositionScore("Pair0")
```

Edge2PositionX

Description

X coordinate of the center of the second edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge2PositionX**(*Pair* As Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = ArcCaliper1.Edge2PositionX(0)
X_Pos = ArcCaliper1.Edge2PositionX("Pair0")
```

Edge2PositionY

Description

Y coordinate of the center of the second edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge2PositionY**(*Pair* As Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim Y_Pos As Single
Y_Pos = ArcCaliper1.Edge2PositionY(0)
Y_Pos = ArcCaliper1.Edge2PositionY("Pair0")
```

Edge2Radius

Description

Radius of the second edge of the selected pair, **ToolPositionX** and **ToolPositionY** being the center of the circular arc described by the said edge. Read only.

Syntax

control.**Edge2Radius**(*Pair* As Variant)

Type

Single

Range

Greater than zero.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The radius is expressed in units specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False	Pixel
CalibratedUnitsEnabled = True	UnitsLength property

Example

```
Dim CircDist As Single
CircDist = ArcCaliper1.Edge2Radius("Pair0")
```

Edge2Rotation

Description

Rotation of the second edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge2Rotation**(*Pair* as Variant)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	Edge1Rotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation As Single
Rotation = ArcCaliper1.Edge2Rotation(0)
Rotation = ArcCaliper1.Edge2Rotation("Pair0")
```

Edge2Score

Description

Score of the second edge of the selected pair. The score is computed according to the constraints set by the **Edge2PositionConstraint** property. Read only.

Syntax

```
control.Edge2Score(Pair As Variant)
```

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = ArcCaliper1.Edge2Score(0)
Score = ArcCaliper1.Edge2Score("Pair0")
```

Edge2ScoreThreshold

Description

Minimum score to accept an edge as the second edge of the selected pair. The score of the second edge is returned by the **Edge2Score** property.

Syntax

control.**Edge2ScoreThreshold**(*Pair* As Variant) [= *Value* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim ScoreTh As Single
ScoreTh = ArcCaliper1.Edge2ScoreThreshold(1)
ArcCaliper.Edge2ScoreThreshold("Pair0") = 0.75
```

EdgeFilterHalfWidth

Description

Half-width of the convolution filter used to compute the edge magnitude curve from which actual edges are detected. The filter approximates the first derivative of the projection curve. The half-width of the filter should be set in order to match the width of the edge in the projection curve (the extent of the grey-scale transition, expressed in number of pixels).

Syntax

control.**EdgeFilterHalfWidth** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 25

Example

```
Dim HalfWidth As Integer
HalfWidth = ArcCaliper1.EdgeFilterHalfWidth
```

`ArcCaliper2.EdgeFilterHalfWidth = 4`

EdgeMagnitudeThreshold

Description

Magnitude threshold is used to find edges on the magnitude curve. A subpixel peak detection algorithm is applied on the region of every minimum or maximum of the curve that exceeds this threshold in order to locate edges.

Syntax

`control.EdgeMagnitudeThreshold [= Value As Single]`

Type

Single

Range

Minimum: 0

Maximum: 255

Example

```
Dim MagTh As Single
```

```
MagTh = ArcCaliper1.EdgeMagnitudeThreshold
```

```
ArcCaliper2.EdgeMagnitudeThreshold = 50.0
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

`control.ElapsedTime`

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
```

```
TimeSpent = ArcCaliper1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

`control.ErrorObject`

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

InputImage

InputGreyScaleImage

Description

Name of the Image on which the tool is applied.

Syntax

```
control.InputImage [= Name as String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = ArcCaliper1.InputImage
ArcCaliper2.InputImage = "image1"
```

InputView

InputGreyScaleView

Description

Name of the View from which the Image is retrieved. See **InputImage** for more details.

Syntax

```
control.InputView [= Name as String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = ArcCaliper1.InputView
ArcCaliper2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = ArcCaliper1.InputInstanceScene
ArcCaliper2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = ArcCaliper1.InputInstanceView
ArcCaliper2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = ArcCaliper1.InputModelScene
ArcCaliper2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View within which is retrieved the Model Scene. See **InputModelScene** for more details.

Syntax

control.InputModelView [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = ArcCaliper1.InputModelView
ArcCaliper2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

control.InstanceSelectionEnabled [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Enabled As Boolean
Enabled = ArcCaliper1.InstanceSelectionEnabled
ArcCaliper2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and the **InstanceSelectionEnabled** properties are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.InstanceSelectionIndex [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: number of instances found by the Locator, minus 1

Example

```
Dim Ndx As Integer
Ndx = ArcCaliper1.InstanceSelectionIndex
ArcCaliper2.InstanceSelectionIndex = 0
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.ModelTransformEnabled [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = ArcCaliper1.ModelTransformEnabled
ArcCaliper2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.Name [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = ArcCaliper1.Name
ArcCaliper2.Name = "ArcCaliperB"
```

PairCount

Description

Number of pairs defined in the tool. Read only.

Syntax*control*.**PairCount****Type**

Integer

Range

Minimum: 0

Maximum: unlimited

Example

Dim Count As Integer

Count = ArcCaliper2.**PairCount**

PairIndex

Description

Index of the selected pair. Read only. This property is used to retrieve the index of a pair from its name.

Syntax*control*.**PairIndex**(Name As String) As Integer**Parameter(s)**

Parameter	Description	Type	Range
<i>Name</i>	Name of a pair	String	Not applicable.

Return

Integer

Range

Minimum: 0

Maximum: **PairCount** - 1**Example**

Dim PairNdx As Integer

PairNdx = ArcCaliper1.**PairIndex**("pad_width")

PairName

Description

Name of the selected pair.

Syntax*control*.**PairName**(Pair As Variant) [= *NewName* As String]**Type**

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim PName As String
PName = ArcCaliper1.PairName(0)
ArcCaliper2.PairName("Pair0") = "FirstPair"
ArcCaliper3.PairName(1) = PName
```

PairPositionX**Description**

X coordinate of the selected pair in the currently selected coordinate system. The position of a pair is defined as the middle of the line segment drawn from the X-Y coordinates of the first and the second edge of that pair. Read only.

Syntax

control.**PairPositionX**(*Pair* As Variant)

Type

Single

Range

Bounded by the image. The minimum and maximum values depend on the **UnitsLength** property which sets the length units and the **CoordinateSystem** property which defines the coordinate system used by the tool to return its results.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = ArcCaliper1.PairPositionX("Pair0")
X_Pos = ArcCaliper2.PairPositionX(0)
```

PairPositionY

Description

Y coordinate of the selected pair in the currently selected coordinate system. The position of a pair is defined as the middle of the line segment drawn from the X-Y coordinates of the first and the second edge of that pair. Read only.

Syntax

control.**PairPositionY**(*Pair* As Variant)

Type

Single

Range

Bounded by the image. The minimum and maximum values depend on the **UnitsLength** property which sets the length units and the **CoordinateSystem** property which defines the coordinate system used by the tool to return its results.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim Y_Pos As Single
Y_Pos = ArcCaliper1.PairPositionY("Pair0")
Y_Pos = ArcCaliper2.PairPositionY(0)
```

PairRadius

Description

Mean value of **Edge1Radius** and **Edge2Radius** for the selected pair. Mathematically, this property corresponds to: $0.50 * (\text{Edge1Radius} + \text{Edge2Radius})$. Read only.

Syntax

Control.**PairRadius** (*Pair* As Variant)

Type

Single

Range

Greater than zero.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The radius is expressed in units specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False	Pixel
CalibratedUnitsEnabled = True	UnitsLength property

Example

```
Dim Value As Single
Value = ArcCaliper1.PairRadius ( 0 )
```

PairRotation

Description

Angle of rotation of the selected pair in the currently selected coordinate system. The rotation of a given pair is always the same as the rotation of its first and second edges. Read only.

Syntax

Control.**PairRotation** (*Pair* As Variant)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	Edge1Rotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Angle As Single
Angle = ArcCaliper1.PairRotation ( 0 )
Angle = ArcCaliper1.PairRotation ("Pair0")
```

PairScore

Description

Score of the selected pair. The score of the pair is equal to the mean score of the two edges (**Edge1Score** and **Edge2Score**) that comprise the pair. Read only.

Syntax

control.**PairScore**(*Pair* As Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim PScore As Single
PScore = ArcCaliper1.PairScore(0)
PScore = ArcCaliper1.PairScore("Pair0")
```

PairSize

Description

Distance between the two edges of a selected pair in the currently selected coordinate system. Read only.

Syntax

control.**PairSize**(*Pair* As Variant)

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input Image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim Distance As Single
Distance = ArcCaliper1.PairSize(0)
Distance = ArcCaliper1.PairSize("Pair0")
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be performed using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
```

```
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

ProjectionMode

Description

Projection mode used by the Arc Caliper tool.

Syntax

```
control.ProjectionMode [= Value As hsProjectionMode]
```

Type

hsProjectionMode

Range

Constant	Value	Description
hsProjectionAnnular	0	Annular projection is used to find edges that are aligned with the median annulus, such as arcs on concentric circles.
hsProjectionRadial	1	Radial projection is used to find edges aligned along radial projections, much like the spokes of a wheel.

Remark(s)

The default value for the **ProjectionMode** property is **hsProjectionRadial**.

Example

```
Dim ProjMode As hsProjectionMode
ProjMode = ArcCaliper1.ProjectionMode
ArcCaliper1.ProjectionMode = hsProjectionRadial
```

SamplingStep

Description

Step used in the tool's last execution to sample the rectangle from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

```
control.SamplingStep
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = ArcCaliper1.SamplingStep
```

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input Image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = ArcCaliper1.SamplingStepCustom
ArcCaliper2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Enabled As Boolean
Enabled = ArcCaliper1.SamplingStepCustomEnabled
ArcCaliper2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

control.**SamplingStepDefault**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input Image.

Example

Dim Sampling As Single

Sampling = ArcCaliper.**SamplingStepDefault**

ToolOpening

Description

Angle between the two bounding radii of the tool's sector.

Syntax

control.**ToolOpening** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolOpening property range
hsDegree	Minimum: -0 deg Maximum: 360 deg
hsRadian	Minimum: 0 rad Maximum: 2Pi rad

Example

Dim Value As Single

Value = ArcCaliper1.**ToolOpening**

ArcCaliper2.**ToolOpening** = 30.0

ToolPositionX

Description

X coordinate of the origin of the Tool.

Syntax

control.**ToolPositionX** [= Value as Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = ArcCaliper1.ToolPositionX
ArcCaliper2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the origin of the Tool.

Syntax

control.**ToolPositionY** [= *Value* as Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = ArcCaliper1.ToolPositionY
ArcCaliper2.ToolPositionY = 23.2
```

ToolRadius

Description

The radius of the tool corresponds to the radius of the median annulus of the tool's sector.

Syntax

control.**ToolRadius** [*Value* As Single]

Type

Single

Range

Greater than zero.

Remark(s)

The property is expressed in length units as specified in the following table:

Mode	Length unit
ModelTransformEnabled = False	Pixel
ModelTransformEnabled = True	UnitsLength property

Example

Dim Value As Single

Value = ArcCaliper1.**ToolRadius**

ArcCaliper2.**ToolRadius** = 25.0

ToolRotation

Description

Angle between the sector's bisector and the X-Axis of the coordinate system used for tool positioning.

Syntax

control.**ToolRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

Dim Angle As Single

Angle = ArcCaliper1.**ToolRotation**

ArcCaliper2.**ToolRotation** = -10.0

ToolThickness

Description

Distance between its two bounding annuli of the tool sector.

Syntax

control.**ToolThickness** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = ArcCaliper1.ToolThickness
ArcCaliper2.ToolThickness = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [= *Value* as Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = ArcCaliper1.TransformRotation
ArcCaliper2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value as Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = ArcCaliper1.TransformScale
ArcCaliper2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value as Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = ArcCaliper1.TransformTranslationX
ArcCaliper2.TransformTranslationX = 22.0
```


TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value as Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = ArcCaliper1.TransformTranslationY
ArcCaliper2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSArcCaliper".

Example

```
Dim ControlType As String
ControlType = ArcCaliper1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.UnitsLength [= Value As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Remark(s)

This property is only applicable if **ModelTransformEnabled** or **CalibratedUnitsEnabled** is set to True.

Example

```
Dim Unit As hsUnitsLength
Unit = ArcCaliper1.UnitsLength
ArcCaliper2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit

Syntax

```
control.UnitsOrientation [= Value As hsUnitsOrientation]
```

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = ArcCaliper1.UnitsOrientation
ArcCaliper2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax*control*.**Version****Type**

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = ArcCaliper1.Version
```

Methods

AboutBox

Description

Displays the ArcCaliper tool About box.

Syntax*control*.**AboutBox****Parameter(s)**

None

Return

No return value

ExampleArcCaliper1.**AboutBox**

AddPair

Description

Creates a new pair.

Syntax*control*.**AddPair**(*Name* As String) As Integer**Return**

Integer: index of the newly created pair.

Range

Minimum: 0

Maximum: unlimited

Parameter(s)

Name	Description	Type	Range
Name	Name of the ArcCaliper pair to be added	String	Not applicable.

Example

Dim NewIndex As Integer

```
NewIndex = ArcCaliper1.AddPair("ObjectWidth")
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the ArcCaliper process.

Syntax

```
control.Execute( )
```

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

Example

```
ArcCaliper1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

```
control.GetProcessingFormatEnabled(ProcessingFormat As hsProcessingFormat
```

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

```
control.HideProperties( )
```

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = ArcCaliper1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

RemovePair

Description

Removes a pair from the tool.

Syntax

```
control.RemovePair(Pair As Variant)
```

Return

No return value

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	ArcCaliper pair selected	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
ArcCaliper1.RemovePair("Pair0")
ArcCaliper1.RemovePair(0)
```

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and what is the behavior of the control when the OK button is clicked. When this parameter is not provided all the elements of the bitfield are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Value	Description
True	The operation succeeded.

Value	Description
False	The operation failed.

Example

```
Dim Success As Boolean
Success = ArcCaliper1.ShowProperties_
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Arc Edge Locator

Description

The Arc Edge Locator finds and locates radial or annular edges on arc-shaped objects according to user-defined criteria such as strength and polarity. The Arc Edge Locator also scores and sorts located edges.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the Image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* as Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result as Boolean
```

```
Result = ArcEdgeLocator1.BilinearInterpolationEnabled
```

```
ArcEdgeLocator2.BilinearInterpolationEnabled
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* as Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result as Boolean
Result = ArcEdgeLocator1.CalibratedUnitsEnabled
ArcEdgeLocator2.CalibratedUnitsEnabled
```

Constraints

Description

Bitwise field used to select the edge detection constraints. Constraints can be set for position and/or magnitude and are used to score edges.

Syntax

control.**Constraints** [= Value as hsConstraints]

Type

hsConstraints

Range

The valid range for this property is as follows:

Name	Value	Description
hsNone	0	No constraint
hsPosition	1	Position constraint
hsMagnitude	2	Magnitude constraint
hsAllConstraints	3	Magnitude and position constraints

Example

```
Dim EdgeConstraint as Integer
EdgeConstraint = ArcEdgeLocator1.Constraints
ArcEdgeLocator2.Constraints = hsMagnitude
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Name	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.

Name	Value	Description
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to Tool coordinate system.

Example

```
Dim Result as hsCoordinateSystem
Result = ArcEdgeLocator1.CoordinateSystem
ArcEdgeLocator1.CoordinateSystem = hs0b
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = ArcEdgeLocator1.Description
```

EdgeCount

Description

Number of edges detected by the tool. Read only.

Syntax

control.**EdgeCount**

Type

Integer

Range

Minimum: 0

Maximum: unlimited

Example

Dim Count as Integer

Count = ArcEdgeLocator1.**EdgeCount**

EdgeMagnitude

Description

Magnitude of the selected edge. Read only.

Syntax

control.**EdgeMagnitude** (*Index* as Integer)

Type

Single

Range

Minimum: -255

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Integer	Minimum: 0 Maximum: EdgeCount -1

Example

Dim Result as Single

Result = ArcEdgeLocator1.**EdgeMagnitude**(0)

EdgeMagnitudeScore

Description

Magnitude score of the selected edge. Read only.

Syntax

control.**EdgeMagnitudeScore** (*Index* As Integer)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount-1

Example

```
Dim Score as Single
Score = ArcEdgeLocator1.EdgeMagnitude(0)
```

EdgePositionScore

Description

Position score of the selected edge. Read only.

Syntax

control.**EdgePositionScore** (*Index* As Integer)

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount-1

Example

```
Dim Score as Single
Score = ArcEdgeLocator1.EdgePositionScore(0)
```

EdgePositionX

Description

X coordinate of the center of the selected edge in the currently selected coordinate system.
Read only.

Syntax

control.**EdgePositionX** (*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Integer	Minimum: 0 Maximum: EdgeCount-1

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim PosX as Single
PosX = ArcEdgeLocator1.EdgePositionX(0)
```

EdgePositionY

Description

Y coordinate of the center of the selected edge in the currently selected coordinate system.
Read only.

Syntax

control.**EdgePositionY** (*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Integer	Minimum: 0 Maximum: EdgeCount -1

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim PosX as Single
PosX = ArcEdgeLocator1.EdgePositionY(0)
```

EdgeRadius

Description

Radius of the selected edge, **ToolPositionX** and **ToolPositionY** being the center of the circular arc described by the selected edge. Read only.

Syntax

control.**Radius**(*Index* As Integer)

Type

Single

Range

Greater than zero.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Integer	Minimum: 0 Maximum: EdgeCount-1

Remark(s)

The radius is expressed in units specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False	Pixel
CalibratedUnitsEnabled = True	UnitsLength property

Example

```
Dim CircDist as Single
CircDist = ArcEdgeLocator1.EdgeRadius(0)
```

EdgeRotation

Description

Rotation of the selected edge with respect to the currently selected coordinate system. Read only.

Syntax

control.**EdgeRotation** (*Index* As Integer)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	EdgeRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Integer	Minimum: 0 Maximum: EdgeCount-1

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation as Single
Rotation = ArcEdgeLocator1.EdgeRotation(0)
```

EdgeScore

Description

Score of the selected edge. The score is computed according the constraints set by the **Constraints** property. Read only.

Syntax

control.**EdgeScore** (*Index* As Integer)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Integer	Minimum: 0 Maximum: EdgeCount -1

Example

```
Dim Score as Single
Score = ArcEdgeLocator1.EdgeScore ( 0 )
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent as Single
TimeSpent = ArcEdgeLocator1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

FilterHalfWidth

Description

Half-width of the convolution filter used to compute the edge magnitude curve from which actual edges are detected. The filter approximates the first derivative of the projection curve. The half width of the filter should be set in order to match the width of the edge in the projection curve (the extent of the grey-scale transition, expressed in number of pixels).

Syntax

```
control.FilterHalfWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: 1

Maximum: 25

Example

```
Dim HalfWidth as Integer
HalfWidth = ArcEdgeLocator1.FilterHalfWidth
EdgeLocator2.FilterHalfWidth = 2
```

InputImage

InputGreyScaleImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

```
control.InputImage [= Name as String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName as String
ImageName = ArcEdgeLocator1.InputImage
ArcEdgeLocator2.InputImage = "image"
```


InputView

InputGreyScaleView

Description

Name of the View within which is retrieved the image. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName as String
ViewName = ArcEdgeLocator1.InputView
EdgeLocator2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.InputInstanceScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName as String
SceneName = ArcEdgeLocator1.InputInstanceScene
ArcEdgeLocator2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View within which is retrieved the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.InputInstanceView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName as String
ViewName = ArcEdgeLocator1.InputInstanceView
EdgeLocator2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName as String
ModelSceneName = ArcEdgeLocator1.InputModelScene
ArcEdgeLocator2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View within which is retrieved the Model Scene. See **InputModelScene** for more details.

Syntax

control.**InputModelView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName as String
ModelViewName = ArcEdgeLocator1.InputModelView
ArcEdgeLocator2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

Specifies if the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property.

Syntax

control.**InstanceSelectionEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user controlled.

Remark(s)

This property is only significant if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Enabled as Boolean
Enabled = ArcEdgeLocator1._
    InstanceSelectionEnabled
ArcEdgeLocator2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **InstanceSelectionEnabled** property is set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.**InstanceSelectionIndex** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx as Integer
Ndx = ArcEdgeLocator1.InstanceSelectionIndex
ArcEdgeLocator2.InstanceSelectionIndex = 0
```

MagnitudeConstraint

Description

Indexed property used to set the magnitude constraint function. Two points are used: Base and Top.

Syntax

control.**MagnitudeConstraint** (*Index* As hsMagnitudeConstraintIndex) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	One of the two points of the magnitude constraint function.	hsMagnitudeConstraintIndex	hsBase: 1 hsTop: 2

Example

```
Dim MagCt as Integer
MagCt = ArcEdgeLocator1.MagnitudeConstraint(hsBase) = 9
ArcEdgeLocator2.MagnitudeConstraint(2) = 60
```

MagnitudeThreshold

Description

Magnitude threshold is used to find edges on the magnitude curve. A subpixel peak detection algorithm is applied on the region of every minimum or maximum of the curve that exceeds this threshold in order to locate edges.

Syntax

control.**MagnitudeThreshold** [= *Value* As Single]

Type

Single

Range

Single: 0

Maximum: 255

Example

```
Dim MagTh as Single
MagTh = ArcEdgeLocator1.MagnitudeThreshold
ArcEdgeLocator2.MagnitudeThreshold = 50.0
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model based.
False	The control is not model based.

Example

```
Dim Enabled as Boolean
Enabled = ArcEdgeLocator1.ModelTransformEnabled
ArcEdgeLocator2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName as String
ProcName = ArcEdgeLocator1.Name
ArcEdgeLocator2.Name = "ArcEdgeLocA"
```

Polarity

Description

Edge selection criterion. The grey-scale transition of the edge must respect the polarity set by this property.

Syntax

control.**Polarity** [= Value As hsPolarity]

Type

hsPolarity

Range

The valid range for this property is as follows:

Name	Value	Description
hsDarkToLight	0	The greylevel value must go from dark to light when crossing an edge.
hsLightToDark	1	The greylevel value must go from light to dark when crossing an edge.
hsEitherPolarity	2	The change in greylevel value change is not a criterion for locating an edge.

Example

```
Dim Pol as hsPolarity
Pol = ArcEdgeLocator1.Polarity
ArcEdgeLocator2.Polarity = hsLightToDark
```

PositionConstraint

Description

Indexed property used to set the position constraint function. Four points are used: Base Left, Top Left, Top Right, Base Right.

Syntax

control.**PositionConstraint** (*Index* As hsPositionConstraintIndex) [= *Value* As Integer]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	One of the four points of the position constraint function.	hsPositionConstraintIndex	hsBaseLeft: 1 hsTopLeft: 2 hsTopRight: 3 hsBaseRight: 4

Remark(s)

A position constraint value of 0.0 represents the left of the tool while 1.0 represents the right of the tool.

Example

```
Dim PosCt as Single
```

```
PosCt = ArcEdgeLocator1.PositionConstraint(hsBaseLeft)
```

```
ArcEdgeLocator2.PositionConstraint(hsTopLeft) = 0.8
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be perform using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.

Constant	Value	Description
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

ProjectionMode

Description

Projection mode used by the Arc Edge Locator tool.

Syntax

```
control.ProjectionMode [= Value As hsProjectionMode]
```

Type

hsProjectionMode

Range

Name	Value	Description
hsProjectionAnnular	0	Annular projection is used to find edges that are aligned with the median annulus, such as arcs on concentric circles.
hsProjectionRadial	1	Radial projection is used to find edges aligned along radial projections, much like the spokes of a wheel.

Remark(s)

The default value for the **ProjectionMode** property is **hsProjectionRadial**.

Example

```
Dim ProjMode as hsProjectionMode
ProjMode = ArcEdgeLocator1.ProjectionMode
ArcEdgeLocator1.ProjectionMode = hsProjectionRadial
```

SamplingStep

Description

Step used in the tool's last execution to sample the rectangle from the input image. When the tool is model-based, this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

```
control.SamplingStep
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step as Single  
Step = ArcEdgeLocator1.SamplingStep
```

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step

Syntax

control.**SamplingStepCustom** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step as Single  
Step = ArcEdgeLocator1.SamplingStepCustom  
AcrEdgeLocator2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default optimal sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Example

```
Dim Enabled as Boolean
Enabled = ArcEdgeLocator1.SamplingStepCustomEnabled
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

```
control.SamplingStepDefault
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Example

```
Dim Sampling as Single
Sampling = ArcEdgeLocator1.SamplingStepDefault
```

ScoreThreshold

Description

Minimum score to accept an edge. The score of an edge is returned by the **EdgeScore** property.

Syntax

```
control.ScoreThreshold [= Value As Single]
```

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Example

```
Dim Value as Single
Value = ArcEdgeLocator1.ScoreThreshold
ArcEdgeLocator2.ScoreThreshold = 0.65
```

SortResultsEnabled

Description

Property that specifies if edges are sorted in descending order of score values.

Syntax

```
control.SortResultsEnabled [= Value As Boolean ]
```

Type

Boolean

Range

Value	Description
False	The edges are not sorted.
True	The edges are sorted in descending order of score values.

Example

```
Dim SortEnabled as Boolean
SortEnabled = ArcEdgeLocator1.SortResultsEnabled
ArcEdgeLocator2.SortResultsEnabled = False
```

ToolOpening

Description

Angle between the two bounding radii of the tool's sector.

Syntax

```
control.ToolOpening [= Value As Single]
```

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolOpening property range
hsDegree	Minimum: 0 deg Maximum: 360 deg
hsRadian	Minimum: 0 rad Maximum: 2Pi rad

Example

```
Dim Value as Single
Value = ArcEdgeLocator1.ToolOpening
ArcEdgeLocator2.ToolOpening = 30.0
```

ToolPositionX

Description

X coordinate of the origin of the Tool.

Syntax

```
control.ToolPositionX [= Value as Single]
```

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos as Single
X_Pos = ArcEdgeLocator1.ToolPositionX
ArcEdgeLocator2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the origin of the Tool.

Syntax

control.**ToolPositionY** [= *Value* as Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos as Single
Y_Pos = ArcEdgeLocator1.ToolPositionY
ArcEdgeLocator2.ToolPositionY = 23.2
```

ToolRadius

Description

The radius of the tool corresponds to the radius of the median annulus of the tool's sector.

Syntax

control.**ToolRadius** [*Value* As Single]

Type

Single

Range

Greater than zero.

Remark(s)

The property is expressed in length units as specified in the following table:

Mode	Length unit
ModelTransformEnabled = False	Pixel
ModelTransformEnabled = True	UnitsLength property

Example

Dim Value as Single

Value = ArcEdgeLocator1.ToolRadius

ArcEdgeLocator2.**ToolRadius** = 25.0

ToolRotation

Description

Angle between the sector's bisector and the X-Axis of the coordinate system used for tool positioning.

Syntax

control.**ToolRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

Dim Angle as Single

Angle = ArcEdgeLocator1.**ToolRotation**

ArcEdgeLocator2.**ToolRotation** = -10.0

ToolThickness

Description

Distance between its two bounding annuli of the tool sector.

Syntax

control.**ToolThickness** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height as Single
Height = ArcEdgeLocator1.ToolThickness
ArcEdgeLocator2.ToolThickness = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [= *Value* as Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle as Single
Angle = ArcEdgeLocator1.TransformRotation
ArcEdgeLocator2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of a transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value as Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale as Single
Scale = ArcEdgeLocator1.TransformScale
ArcEdgeLocator2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value as Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in the length units specified by the **UnitsLength** property.

Example

```
Dim X_Trans as Single
X_Trans = ArcEdgeLocator1.TransformTranslationX
ArcEdgeLocator2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the control if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value as Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in the length units specified by the **UnitsLength** property.

Example

```
Dim Y_Trans as Single
Y_Trans = ArcEdgeLocator1.TransformTranslationY
ArcEdgeLocator2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSArcEdgeLocator".

Example

```
Dim ControlType as String
ControlType = ArcEdgeLocator1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.UnitsLength [= Value as hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Name	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit as hsUnitsLength
Unit = ArcEdgeLocator1.UnitsLength
ArcEdgeLocator2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* as hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Name	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit as hsUnitsOrientation
Unit = ArcEdgeLocator1.UnitsOrientation
ArcEdgeLocator2.UnitsOrientation= hsDegree
```

Version

Description

String containing the current HexSight version and build information. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version as String
Version = ArcEdgeLocator1.Version
```

Methods

AboutBox

Description

Displays the Arc Edge Locator tool About box.

Syntax*control*.**AboutBox****Parameter(s)**

None

Return

No return value.

ExampleArcEdgeLocator1.**AboutBox**

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the ArcEdgeLocator process.

Syntax*control*.**Execute**()**Parameter(s)**

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

ExampleArcEdgeLocator1.**Execute**

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

control.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed as Boolean
InterfaceWasDisplayed = ArcEdgeLocator1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* as Boolean, [*Variant Option*]) As Boolean

Parameters

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and what is the behavior of the control when the OK button is clicked. When this parameter is not provided all the elements of the bitfield are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Value	Description
True	The operation was successful.
False	The operation failed.

Example

```
Dim Success as Boolean
Success = ArcEdgeLocator1.ShowProperties _
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Arc Finder

Description

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.

Properties

ArcAverageContrast

Description

Average contrast between light and dark pixels on either side of the found arc, expressed in greylevel values. Read only.

Syntax

control.**ArcAverageContrast**

Type

Integer

Range

Minimum: greater than 0.

Maximum: 255

Example

```
Dim AverageContrast As Integer
ArcFinder1.Execute
AverageContrast = ArcFinder1.ArcAverageContrast
```

ArcCenterPointPositionX

Description

X coordinate of the center of the found arc, expressed in the currently selected coordinate system. Read only.

Syntax

control.**ArcCenterPointPositionX**

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim ArcCenterX as Single
```

ArcCenterX = ArcFinder1.ArcCenterPointPositionX

ArcCenterPointPositionY

Description

Y coordinate of the center of the found arc, expressed in the currently selected coordinate system. Read only.

Syntax

control.ArcCenterPointPositionY

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim ArcCenterY as Single
ArcCenterY = ArcFinder1.ArcCenterPointPositionY
```

ArcEndAngle

Description

Angle of the arc radius at the end point of the found arc, expressed in the currently selected coordinate system. Read only.

Syntax

control.ArcEndAngle

Type

Single

Range

-180 to 180 degrees

Example

```
Dim EndAngle as Single
EndAngle = ArcFinder1.ArcEndAngle
```

ArcEndPointPositionX

Description

X coordinate of the arc end point, expressed in the currently selected coordinate system. On an x-to-y-axis trajectory, the arc start point, defined by **ArcStartPointPositionX** and **ArcStartPointPositionY**, is the first point encountered; the end point, defined by **ArcEndPointPositionX** and **ArcEndPointPositionY**, is the last point encountered. Read only.

Syntax

control.ArcEndPointPositionX

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim ArcEndX as Single
ArcEndX = ArcFinder1.ArcEndPointPositionX
```

ArcEndPointPositionY

Description

Y coordinate of the arc end point, expressed in the currently selected coordinate system. On an x-to-y-axis trajectory, the arc start point, defined by **ArcStartPointPositionX** and **ArcStartPointPositionY**, is the first point encountered; the end point, defined by **ArcEndPointPositionX** and **ArcEndPointPositionY**, is the last point encountered. Read only.

Syntax

```
control.ArcEndPointPositionY
```

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim ArcEndY as Single
ArcEndY = ArcFinder1.ArcEndPointPositionY
```

ArcFitQuality

Description

Normalized average error between the calculated arc and the actual edges matched to the found arc. Fit quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to conformity tolerance. Read only.

Syntax

```
control.ArcFitQuality
```

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Example

```
Dim FitQuality As Single
FitQuality = ArcFinder1.ArcFitQuality
```

ArcFound

Description

When True, indicates that an arc was found. Read only.

Syntax

control.ArcFound

Type

Boolean

Range

Not applicable.

Example

```
Dim ArcFound as Boolean
ArcFound = ArcFinder1.ArcFound
```

ArcMatchQuality

Description

Percentage of edges actually matched to the found arc. Match Quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that edges were matched for every point along the found arc. Similarly, a value of 0.2 means edges were matched to 20% of the points along the found arc. Read only.

Syntax

control.ArcMatchQuality

Type

Single

Range

Minimum: greater than zero.
Maximum: 1.0

Example

```
Dim MatchQuality As Single
MatchQuality = ArcFinder1.ArcMatchQuality
```

ArcMustBeTotallyEnclosed

Description

When **ArcMustBeTotallyEnclosed** is True, the start and end points of the arc must be located on the radial bounding sides of the Search Area. When set to False, the found arc can enter and/or exit the Search Area at the inner or outer annular bounds of the Search Area.

Syntax

control.ArcMustBeTotallyEnclosed [= Value as Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim ArcEnclosed as Boolean
ArcEnclosed = ArcFinder1.ArcMustBeTotallyEnclosed
ArcFinder2.ArcMustBeTotallyEnclosed = True
```

ArcRadius

Description

Radius of the found arc, **ArcCenterPointPositionX** and **ArcCenterPointPositionY** being the center of the circular arc described by the found arc. Read only.

Syntax

control.**Radius**

Type

Single

Range

Greater than zero.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim Radius as Single
Radius = ArcFinder1.ArcRadius
```

ArcStartAngle

Description

Angle of the arc radius at the start point of the found arc, expressed in the currently selected coordinate system. Read only.

Syntax

control.**ArcStartAngle**

Type

Single

Range

Unbounded

Example

```
Dim StartAngle as Single
StartAngle = ArcFinder1.ArcStartAngle
```

ArcStartPointPositionX

Description

X coordinate of the arc start point, expressed in the currently selected coordinate system. On an x-to-y-axis trajectory, the arc start point, defined by **ArcStartPointPositionX** and **ArcStartPointPositionY**, is the first point encountered; the end point, defined by **ArcEndPointPositionX** and **ArcEndPointPositionY**, is the last point encountered. Read only.

Syntax*control.ArcStartPointPositionX***Type**

Single

Range

-180 to 180 degrees

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim ArcStartX as Single
ArcStartX = ArcFinder1.ArcStartPointPositionX
```

ArcStartPointPositionY

Description

Y coordinate of the arc start point, expressed in the currently selected coordinate system. On an x-to-y-axis trajectory, the arc start point, defined by **ArcStartPointPositionX** and **ArcStartPointPositionY**, is the first point encountered; the end point, defined by **ArcEndPointPositionX** and **ArcEndPointPositionY**, is the last point encountered. Read only.

Syntax*control.ArcStartPointPositionY***Type**

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim ArcStartY as Single
ArcStartY = ArcFinder1.ArcStartPointPositionY
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax*control.CalibratedUnitsEnabled* [= Value As Boolean]**Type**

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is Model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = ArcFinder1.CalibratedUnitsEnabled
ArcFinder2.CalibratedUnitsEnabled = False
```

ConformityTolerance

Description

Maximum local deviation between the expected edges of an arc and the edges actually detected in the input image. It corresponds to the maximum distance in calibrated units by which a matched edge can deviate from either side of its expected position in the arc. Expressed in the calibrated units specified by the **UnitsLength** property. This property can only be set when **UseDefaultConformityTolerance** is set to false. Read only otherwise.

Syntax

control.**ConformityTolerance**

Type

Single

Range

Minimum: **MinimumConformityTolerance**

Maximum: **MaximumConformityTolerance**

Example

```
Dim ConformityTolerance As Single
ConformityTolerance = ArcFinder1.ConformityTolerance
ArcFinder2.ConformityTolerance = 2.0
```

ContrastThreshold

Description

Defines the minimum contrast needed for an edge to be detected in the input image and used for arc computation. This threshold is expressed in terms of a step in greylevel values. The property is read only except when **ContrastThresholdMode** is set to **hsContrastThresholdFixedValue**.

Syntax

control.**ContrastThreshold** [= Value As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 255

Remark(s)

By default, the tool selects a **ContrastThresholdMode** based on image content to provide flexibility to variations in image lighting conditions and contrast. Adaptive threshold modes are generally recommended. A fixed-value contrast threshold should only be used when adaptive values do not provide satisfactory results.

Example

```
Dim CurrentThreshold As Integer
CurrentThreshold = ArcFinder1.ContrastThreshold
ArcFinder2.ContrastThresholdMode = hsContrastThresholdModeFixedValue
ArcFinder2.ContrastThreshold = 10
```

ContrastThresholdMode

Description

Selects the method used to compute the threshold used for detecting edges in the input image.

Syntax

control.**ContrastThresholdMode** [= *Value* As hsContrastThresholdMode]

Type

hsContrastThresholdMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContrastThresholdAdaptiveLowSensitivity	0	Uses a low sensitivity adaptive threshold for detecting edges. Adaptive Low Sensitivity reduces the amount of noisy edges but may also cause significant edges to be undetected.
hsContrastThresholdAdaptiveNormalSensitivity	1	Uses a normal sensitivity adaptive threshold for detecting edges.
hsContrastThresholdAdaptiveHighSensitivity	2	Uses a high sensitivity adaptive threshold for detecting edges. Adaptive High Sensitivity can help detect weak-contrast edges but also increases the amount of noisy edges.
hsContrastThresholdFixedValue	3	Uses a fixed value threshold for detecting edges.

Remark(s)

By default, the tool selects a **ContrastThresholdMode** based on image content to provide flexibility to variations in image lighting conditions and contrast. Adaptive threshold modes are generally recommended. A fixed-value contrast threshold should only be used when adaptive values do not provide satisfactory results.

Example

```
Dim CurrentMode As hsContrastThresholdMode
```

```
CurrentMode = ArcFinder2.ContrastThresholdMode
```

```
ArcFinder2.ContrastThresholdMode = hsContrastThresholdModeFixedValue
```

```
ArcFinder2.ContrastThresholdMode = 10
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

```
control.CoordinateSystem [= Value As hsCoordinateSystem]
```

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input Image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim Result As hsCoordinateSystem
```

```
Result = ArcFinder1.CoordinateSystem
```

```
ArcFinder2.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

DefaultConformityTolerance

Description

Default value for **ConformityTolerance** computed by the tool according to the calibration, the edge detection parameters, and the search parameters. See also **ConformityTolerance** and **UseDefaultConformityTolerance**. Read only.

Syntax

control.**DefaultConformityTolerance**

Type

Single

Range

Greater than zero.

Remark(s)

This default value is used for **ConformityTolerance** when **UseDefaultConformityTolerance** is set to True.

Example

```
Dim Default As Single
Default = ArcFinder1.DefaultConformityTolerance
```

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = ArcFinder1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0
Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = ArcFinder1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & ArcFinder.ErrorObject.Description
```

FitMode

Description

Specifies the mode used by the tool to calculate and return values for the found arc.

Syntax

control.**FitMode** [= *Value* as hsFitMode]

Type

hsFitMode

Range

The valid range for this property is as follows:

hsFitMode	Value	Description
hsComputeCenterAndRadius	0	The Arc Finder calculates and returns both the arc center and arc radius.
hsComputeRadius	1	The arc radius is calculated, the arc center returned is the value of the tool's center.
hsComputeCenter	2	The arc center is calculated; the radius returned is the value of the tool's radius.

Example

```
' Fit the arc according to a computed radius
ArcFinder1.FitMode = hsComputeRadius
ArcFinder1.Execute
```

InputImage

Description

Name of the Image within which the operation of the tool is carried out.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = ArcFinder1.InputImage
ArcFinder2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the Image within which the operation of the tool is carried out.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = ArcFinder1.InputImage
ArcFinder2.InputImage = "image1"
```

InputView

Description

Name of the View that provides the Image. See **InputImage** for more details.

Syntax

control.**InputView** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = ArcFinder1.InputView
ArcFinder2.InputView = "view0"
```


InputGreyScaleView

Description

Name of the View that provides the Image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = ArcFinder1.InputView
ArcFinder2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the **Locator** tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = ArcFinder1.InputInstanceScene
ArcFinder2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = ArcFinder1.InputInstanceView
```

```
ArcFinder2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

```
control.InputModelScene [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = ArcFinder1.InputModelScene
ArcFinder2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View that provides the Model Scene. See **InputModelScene** for more details.

Syntax

```
control.InputModelView [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = ArcFinder1.InputModelView
ArcFinder2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.
property

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = ArcFinder1.InstanceSelectionEnabled
ArcFinder2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = ArcFinder1.InstanceSelectionIndex
ArcFinder2.InstanceSelectionIndex = 0
```

MaximumAngleDeviation

Description

Maximum deviation in angle that will allow a detected edge to be used for generating an arc hypothesis. The deviation is calculated between the tangent angle of the arc at points where the edge is matched to the arc. Expressed in the orientation units specified by the UnitsOrientation property.

Syntax

```
control.MaximumAngleDeviation [= Value As Single]
```

Type

Single

Range

Minimum: 0 degrees

Maximum: 20 degrees

Example

```
Dim Value as Single
Value = ArcFinder1.MaximumAngleDeviation
ArcFinder2.MaximumAngleDeviation = 10
```

MaximumConformityTolerance

Description

Maximum value allowed for the **ConformityTolerance** property. Read only.

Syntax

control.**MaximumConformityTolerance**

Type

Single

Range

Greater than **MinimumConformityTolerance**.

Remark(s)

This property is computed by the Arc Finder by analyzing the calibration, edge detection parameters, and the search parameters. See also **ConformityTolerance** and **DefaultConformityTolerance**.

Example

```
Dim MaxConformityTolerance As Single
MaxConformityTolerance = ArcFinder1.MaximumConformityTolerance
```

MessageCount

Description

The number of messages issued during the last execution. The identification number and description of each message is retrieved using respectively the **MessageNumber** and **MessageDescription** properties. Read only.

Syntax

control.**MessageCount**

Type

Integer

Range

Minimum: 0

Maximum: Unlimited

Example

```
Dim m As Integer
For m = 0 To ArcFinder1.MessageCount - 1
    Description.Text = ArcFinder1.MessageDescription(m)
    Number.Text = Str(ArcFinder1.MessageNumber(m))
Next m
```

MessageDescription

Description

The description of a message issued during the last execution. The identification number of the message corresponding to this description is retrieved using the **MessageNumber** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageDescription**(*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```
Dim m As Integer
For m = 0 To ArcFinder1.MessageCount - 1
    Description.Text = ArcFinder1.MessageDescription(m)
    Number.Text = Str(ArcFinder1.MessageNumber(m))
Next m
```

MessageNumber

Description

The identification number of a message issued during the last execution. The description of the message corresponding to this identification number is retrieved using the **MessageDescription** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageNumber**(*Index* As Integer)

Type

Integer

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```
Dim m As Integer
For m = 0 To ArcFinder1.MessageCount - 1
    Description.Text = ArcFinder1.MessageDescription(m)
    Number.Text = Str(ArcFinder1.MessageNumber(m))
Next m
```

MinimumArcPercentage

Description

Minimum percentage of arc contours that need to be matched for an arc hypothesis to be considered as valid.

Syntax

control.**MinimumArcPercentage** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: 100.0

Example

```
Dim MinModelPercent As Single
MinModelPercent = ArcFinder1.MinimumModelPercentage
ArcFinder2.MinimumModelPercentage = 50.0
```

MinimumConformityTolerance

Description

Minimum value allowed for the **ConformityTolerance** property. Read only.

Syntax

control.**MinimumConformityTolerance**

Type

Single

Range

Minimum: greater than zero.

Maximum: lower than **MaximumConformityTolerance**.

Remark(s)

This property is computed by the Locator by analyzing the calibration, the edge detection parameters, and the search parameters. See also **ConformityTolerance** and **DefaultConformityTolerance**.

Example

```
Dim MinConformityTolerance As Single
MinConformityTolerance = ArcFinder1.MinimumConformityTolerance
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a Model Image and a transform is applied according to the object to be inspected in the input Image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.

Value	Description
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = ArcFinder1.ModelTransformEnabled
ArcFinder2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = ArcFinder1.Name
ArcFinder2.Name = "ArcFindA"
```

OutputEntity

Description

Name of the entity created by the tool. This entity is optionally output according to the value of the **OutputEntityEnabled** property.

Syntax

control.**OutputEntity** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim Arc As String
Arc = ArcFinder1.OutputEntity
ArcFinder1.OutputEntity = "Arc0"
```

OutputEntityEnabled

Description

Specifies if a found entity will be output to the runtime database.

Syntax

control.**OutputEntityEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The entity will not be output.
True	The entity will be output.

Example

```
Dim Enable As Boolean
Enable = ArcFinder1.OutputEntityEnabled
ArcFinder1.OutputEntityEnabled = False
```

OutputEntityView

Description

Name of the View to which the Output Entity is output. See **OutputEntity** for more details.

Syntax

```
control.OutputEntityView [= Name As String]
```

Type

String

Range

Not applicable.

Remark(s)**Example**

```
Dim ViewName As String
ViewName = ArcFinder1.OutputEntityView
ArcFinder2.OutputEntityView = "ArcView0"
```

PolarityMode

Description

Selects the type of polarity accepted for finding an arc. Polarity identifies the change in greylevel values from the tool center (inside) towards the outside.

Syntax

```
control.PolarityMode [= Value As hsPolarityMode]
```

Type

hsPolarityMode

Range

The valid range for this property is as follows:

hsPolarityMode	Value	Description
<i>hsDarkToLight</i>	0	The Arc Finder searches only for arc instances occurring at a dark to light transition in greylevel values.
<i>hsLightToDark</i>	1	The Arc Finder searches only for arc instances occurring at a light to dark transition in greylevel values.

hsPolarityMode	Value	Description
<i>hsEither</i>	2	The Arc Finder searches only for arc instances occurring either at a light to dark or dark to light transition in greylevel values.
<i>hsDontCare</i>	3	The Arc Finder searches only for arc instances occurring at any transition in greylevel values including reversals in contrast along the arc, for example on an unevenly colored background.

Remark(s)

Enabling **hsEither** can significantly increase the tool's execution time.

Example

```
' Search for an arc at a dark to light transition
ArcFinder1.PolarityMode = hsLightToDark
ArcFinder1.Execute
```

PositioningLevel

Description

Configurable effort level of the entity positioning process. A value of 0 will provide coarser positioning and lower execution time. Conversely, a value of 10 will provide high accuracy positioning of arc entities.

Syntax

control.**PositioningLevel** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 10

Example

```
Dim PositioningLevel As Integer
PositioningLevel = ArcFinder1.PositioningLevel
ArcFinder2.PositioningLevel = 5
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be performed using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SearchMode

Description

Specifies the method used by the tool to select a hypothesis.

Syntax

control.**SearchMode** [= Value as hsSearchMode]

Type

hsSearchMode

Range

The valid range for this property is as follows:

hsSearchMode	Value	Description
hsBestArc	0	Selects the best arc according to hypothesis strength.
hsArcClosestToGuideline	1	Selects the arc hypothesis closest to the Guideline.
hsArcClosestToInside	2	Selects the arc hypothesis closest to the inside of the tool Search Area. (closest to the tool center).
hsArcClosestToOutside	3	Selects the arc hypothesis closest to the outside of the tool Search Area. (furthest from the tool center)

Example

```
' Search for Arc closest to inside of tool
Dim Mode As hsSearchMode
ArcFinder2.SearchMode = hsArcClosestToInside
Mode = ArcFinder2.SearchMode
```

ShowEdges

Description

If true, edges in the Search Area are displayed in the Results display. Candidate edges for hypothesis generation are displayed in a different color (green) than other edges (blue).

Syntax

control.ShowEdges [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = ArcFinder1.ShowEdges
ArcFinder1.ShowEdges = False
```

ShowEntityFound

Description

When **ShowEntityFound** is set to true, the entity found by the tool is displayed in the Results display.

Syntax

control.ShowEntityFound [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = ArcFinder1.ShowEntityFound
ArcFinder1.ShowEntityFound = False
```

ShowEntityFoundAs

Description

Mode used to represent the entity in the Results display.

Syntax

control.ShowEntityFoundAs [= *Value* As hsEntityOutputMode]

Type

Integer

Range

The valid range for this property is as follows:

hsEntityOutputMode	Value	Description
hsFittedEntity	0	The arc is represented by the entire arc, fitted to the actual edges detected in the input image.
hsMatchedEntity	1	The arc is represented by the sections of the arc that have been matched to edges actually detected in the input image.

Example

```
Dim ShowAs as hsEntityOutputMode
```

```
Arcfinder.ShowEntityFoundAs = hsFittedEntity
ShowAs = ArcFinder2.ShowEntityFoundAs
```

ShowImage

Description

When **ShowImage** is set to True, the input image is shown in the Results display.

Syntax

```
control.ShowImage [= Value As Boolean]
```

Type

Boolean

Range

Not applicable.

Example

```
ArcFinder1.ShowImage = True
```

ShowGreyScaleImage

Description

When **ShowImage** is set to True, the input image is shown in the Results display.

Syntax

```
control.ShowImage [= Value As Boolean]
```

Type

Boolean

Range

Not applicable.

Example

```
ArcFinder1.ShowImage = True
```

ShowSearchArea

Description

Specifies if the Search Area is shown in the Results Viewer.

Syntax

```
control.ShowSearchArea [= Value As Boolean]
```

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = ArcFinder1.ShowSearchArea
ArcFinder1.ShowSearchArea = False
```

SubsamplingLevel

Description

Subsampling level used to detect edges that are used by the tool to generate hypotheses. High values provide a coarser search and lower execution time than lower values.

Syntax

control.**SubsamplingLevel** [= Value As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 8

Example

```
Dim SearchLevel As Single
SearchLevel = ArcFinder1.SubsamplingLevel
ArcFinder2.SubsamplingLevel = 4
```

ToolGuidelineOffset

Description

The radial offset of the Guideline marker from the center of the tool Search Area.

Syntax

control.**ToolGuidelineOffset** [= Value As Single]

Type

Single

Range

Maximum: $+0.5 \times \text{ToolThickness}$

Minimum: $-0.5 \times \text{ToolThickness}$

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
ArcFinder1.ToolThickness = 25
ArcFinder1.ToolGuidelineOffset = -5
ArcFinder1.Execute
```

ToolOpening

Description

Angle between the two bounding radii of the tool Search Area.

Syntax

control.**ToolOpening** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolOpening property range
hsDegree	Minimum: 0 deg Maximum: 360 deg
hsRadian	Minimum: 0 rad Maximum: 2Pi rad

Example

```
Dim Value As Single
Value = ArcFinder1.ToolOpening
ArcFinder2.ToolOpening = 30.0
```

ToolPositionX

Description

X coordinate of the position of the tool Search Area

Syntax

control.**ToolPositionX** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim X_Pos As Single
X_Pos = ArcFinder1.ToolPositionX
ArcFinder2.ToolPositionX = 23.2
```

ToolPositionY

Description

X coordinate of the position of the tool Search Area.

Syntax

control.**ToolPositionY** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = ArcFinder1.ToolPositionY
ArcFinder2.ToolPositionY = 23.2
```

ToolRadius

Description

The radius of the tool corresponds to the radius of the median annulus of the tool's Search Area.

Syntax

control.**ToolRadius** [= *Value* As Single]

Type

Single

Range

Greater than zero.

Remark(s)

The property is expressed in length units as specified in the following table:

Mode	Length unit
ModelTransformEnabled = False	Pixel
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim Value as Single
Value = ArcFinder1.ToolRadius
ArcFinder2.ToolRadius = 25.0
```

ToolRotation

Description

Angle of rotation of the tool Search Area.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = ArcFinder1.ToolRotation
ArcFinder2.ToolRotation = -10.0
```

ToolThickness

Description

Distance between the two bounding annuli of the tool Search Area.

SyntaxF

control.**ToolThickness** [= *Value* as Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: Less than 2***ToolRadius**.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Thickness as Single
Thickness = ArcFinder1.ToolThickness
ArcFinder2.ToolThickness = 23.4
```


TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = ArcFinder1.TransformRotation
ArcFinder2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = ArcFinder1.TransformScale
```

`ArcFinder2.TransformScale = 1.10`

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

`control.TransformTranslationX [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = ArcFinder1.TransformTranslationX
ArcFinder2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

`control.TransformTranslationY [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = ArcFinder1.TransformTranslationY
ArcFinder2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSArcFinder".

Example

```
Dim ControlType As String
ControlType = ArcFinder1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = ArcFinder1.UnitsLength
ArcFinder2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = ArcFinder1.UnitsOrientation
ArcFinder2.UnitsOrientation = hsDegree
```

UseDefaultConformityTolerance

Description

Specifies whether the default conformity tolerance returned by the **DefaultConformityTolerance** property is used instead of the user- defined conformity tolerance set by the **ConformityTolerance** property. When set to True, the default conformity tolerance is used.

Syntax

control.**UseDefaultConformityTolerance** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim UseDefaultCT As Boolean
UseDefaultCT = ArcFinder1.UseDefaultConformityTolerance
ArcFinder2.UseDefaultConformityTolerance = False
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = ArcFinder1.Version
```

Methods

AboutBox

Description

Displays the Arc Finder tool **About** box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

```
ArcFinder1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Arc Finder process.

Syntax

control.**Execute()**

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

```
ArcFinder1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

control.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = ArcFinder1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and what is the behavior of the control when the OK button is clicked. When this parameter is not provided all the elements of the bitfield are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Dim Success As Boolean
```

```
Success = ArcFinder1.ShowProperties(True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Barcode Reader

Description

The Barcode Reader reads and extracts information from bar code symbology.

Properties

BarcodeAngle

Description

Angle of rotation of the bar code instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the orientation units specified by the **UnitsOrientation** property. Read only.

Syntax

control.**BarcodeAngle**(*Index* As Integer)

Type

Single

Range

Minimum: -180

Maximum: 180

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```
Dim BarcodeAngle As Single
' Get the angle of the first barcode in the calibrated
' World coordinate system
BarcodeReader1.CoordinateSystem = hsCoordinateSystemWorld
BarcodeAngle = BarcodeReader1.BarcodeAngle(0)
```

BarcodeCornerX

Description

X coordinate of one of the 4 indexed corners of a given bar code instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the length units specified by the **UnitsLength** property when the World, the Object or the Tool coordinate system is selected. Expressed in pixels when the Image coordinate system is selected. In the exceptional case where the tool is not configured to work with calibrated units (**CalibratedUnitsEnabled** and **ModelTransformEnabled** properties are both set to False), values returned in the Tool coordinate system are expressed in pixels, and values cannot be returned in the World or Object coordinate systems. Read only.

Syntax

control.**BarcodeCornerX**(*Index* As Integer, *Corner* As Integer)

Type

Single

Range

Any value inside the boundaries of the selected coordinate system.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1
<i>Corner</i>	Index of the corner.	Integer	0,1,2,3

Example

```
' Draw a green box in the image around the first barcode
BarcodeReader1.CoordinateSystem = hsCoordinateSystemImage
DrawLine BarcodeReader1.BarcodeCornerX(0,0), _
        BarcodeReader1.BarcodeCornerY(0,0), _
        BarcodeReader1.BarcodeCornerX(0,1), _
        BarcodeReader1.BarcodeCornerY(0,1), _
        Green
DrawLine BarcodeReader1.BarcodeCornerX(0,1), _
        BarcodeReader1.BarcodeCornerY(0,1), _
        BarcodeReader1.BarcodeCornerX(0,2), _
        BarcodeReader1.BarcodeCornerY(0,2), _
        Green
DrawLine BarcodeReader1.BarcodeCornerX(0,2), _
        BarcodeReader1.BarcodeCornerY(0,2), _
        BarcodeReader1.BarcodeCornerX(0,3), _
        BarcodeReader1.BarcodeCornerY(0,3), _
        Green
DrawLine BarcodeReader1.BarcodeCornerX(0,3), _
        BarcodeReader1.BarcodeCornerY(0,3), _
        BarcodeReader1.BarcodeCornerX(0,0), _
        BarcodeReader1.BarcodeCornerY(0,0), _
        Green
```

BarcodeCornerY**Description**

Y coordinate of one of the 4 indexed corners of a given bar code instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the length units specified by the **UnitsLength** property when the World, the Object or the Tool coordinate system is selected. Expressed in pixels when the Image coordinate system is selected. In the exceptional case where the tool is not configured to work with calibrated units

(**CalibratedUnitsEnabled** and **ModelTransformEnabled** properties are both set to False), values returned in the Tool coordinate system are expressed in pixels, and values cannot be returned in the World or Object coordinate systems. Read only.

Syntax

control.**BarcodeCornerY**(*Index* As Integer, *Corner* As Integer)

Type

Single

Range

Any value inside the boundaries of the selected coordinate system.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1
<i>Corner</i>	Index of the corner.	Integer	0,1,2,3

Example

```
' Draw a green box in the image around the first barcode
BarcodeReader1.CoordinateSystem = hsCoordinateSystemImage
DrawLine BarcodeReader1.BarcodeCornerX(0,0), _
          BarcodeReader1.BarcodeCornerY(0,0), _
          BarcodeReader1.BarcodeCornerX(0,1), _
          BarcodeReader1.BarcodeCornerY(0,1), _
          Green
DrawLine BarcodeReader1.BarcodeCornerX(0,1), _
          BarcodeReader1.BarcodeCornerY(0,1), _
          BarcodeReader1.BarcodeCornerX(0,2), _
          BarcodeReader1.BarcodeCornerY(0,2), _
          Green
DrawLine BarcodeReader1.BarcodeCornerX(0,2), _
          BarcodeReader1.BarcodeCornerY(0,2), _
          BarcodeReader1.BarcodeCornerX(0,3), _
          BarcodeReader1.BarcodeCornerY(0,3), _
          Green
DrawLine BarcodeReader1.BarcodeCornerX(0,3), _
          BarcodeReader1.BarcodeCornerY(0,3), _
          BarcodeReader1.BarcodeCornerX(0,0), _
          BarcodeReader1.BarcodeCornerY(0,0), _
          Green
```

BarcodeDecodeTime

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Total time of the decoding process. Read only.

Syntax

control.**BarcodeDecodeTime**(*Index* As Integer)

Type

Single

Range

Greater than or equal to zero.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```
Dim DecodeTime As Single
' Get the time spent decoding the first barcode
DecodeTime = BarcodeReader1.BarcodeDecodeTime(0)
```

BarcodeFoundCount

Description

Number of bar codes actually found and decoded in the image. Read only.

Syntax

control.**BarcodeFoundCount**

Type

Integer

Range

Minimum: 0

Maximum: **BarcodeToDecode** - 1

Example

```
Dim BarcodesToDecode As Integer
Dim BarcodesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim ErrorString As Long
Dim i As Integer
' Configure the tool to read 4 barcodes in the image
BarcodeReader1.BarcodeToDecode = 4
BarcodeReader1.Execute
...
' Get the results
BarcodesToDecode = BarcodeReader1.BarcodeToDecode
BarcodesFound = BarcodeReader1.BarcodeFoundCount
' Get the strings of the decoded barcodes
For i = 0 To BarcodesFound - 1
```

```

        ResultString = BarcodeReader1.BarcodeString(i)
    Next i
    ' Get the error code of the undecoded barcodes
    For i = BarcodesFound To BarcodesToDecode - 1
        ErrorCode = BarcodeReader1.BarcodeStatusCode(i)
        ErrorString = BarcodeReader1.BarcodeStatusCodeString(i)
    Next i

```

BarcodeStatusCode

Description

Status code which reports on the success or failure of the decoding process on a given bar code instance. Read only.

Syntax

control.**BarcodeStatusCode**(*Index* As Integer)

Type

Long

Range

The valid range for this property is as follows:

Value	Description
0000	Success
1000	No Execute Performed
3048	Symbology Decode Failed

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```

Dim BarcodesToDecode As Integer
Dim BarcodesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim ErrorString As Long
Dim i As Integer
' Configure the tool to read 4 barcodes in the image
BarcodeReader1.BarcodeToDecode = 4
BarcodeReader1.Execute
...
' Get the results
BarcodesToDecode = BarcodeReader1.BarcodeToDecode
BarcodesFound = BarcodeReader1.BarcodeFoundCount

```

```

' Get the strings of the decoded barcodes
For i = 0 To BarcodesFound - 1
    ResultString = BarcodeReader1.BarcodeString(i)
Next i
' Get the error code of the undecoded barcodes
For i = BarcodesFound To BarcodesToDecode - 1
    ErrorCode = BarcodeReader1.BarcodeStatusCode(i)
    ErrorString = BarcodeReader1.BarcodeStatusCodeString(i)
Next i

```

BarcodeStatusCodeString

Description

Status code message associated to the status code returned by the **BarcodeStatusCode** property. Read only.

Syntax

control.**BarcodeStatusCodeString**(*Index* As Integer)

Type

String

Range

Not applicable

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```

Dim BarcodesToDecode As Integer
Dim BarcodesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim ErrorString As Long
Dim i As Integer
' Configure the tool to read 4 barcodes in the image
BarcodeReader1.BarcodeToDecode = 4
BarcodeReader1.Execute
...
' Get the results
BarcodesToDecode = BarcodeReader1.BarcodeToDecode
BarcodesFound = BarcodeReader1.BarcodeFoundCount
' Get the strings of the decoded barcodes
For i = 0 To BarcodesFound - 1
    ResultString = BarcodeReader1.BarcodeString(i)

```

```

Next i
' Get the error code of the undecoded barcodes
For i = BarcodesFound To BarcodesToDecode - 1
    ErrorCode = BarcodeReader1.BarcodeStatusCode(i)
    ErrorString = BarcodeReader1.BarcodeStatusCodeString(i)
Next i

```

BarcodeString

Description

Data string decoded from a given bar code instance. Read only.

Syntax

control.**BarcodeString**(*Index* As Integer)

Type

String

Range

Not applicable

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```

Dim BarcodesToDecode As Integer
Dim BarcodesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim ErrorString As Long
Dim i As Integer
' Configure the tool to read 4 barcodes in the image
BarcodeReader1.BarcodeToDecode = 4
BarcodeReader1.Execute
...
' Get the results
BarcodesToDecode = BarcodeReader1.BarcodeToDecode
BarcodesFound = BarcodeReader1.BarcodeFoundCount
' Get the strings of the decoded barcodes
For i = 0 To BarcodesFound - 1
    ResultString = BarcodeReader1.BarcodeString(i)
Next i
' Get the error code of the undecoded barcodes
For i = BarcodesFound To BarcodesToDecode - 1
    ErrorCode = BarcodeReader1.BarcodeStatusCode(i)

```

```
ErrorString = BarcodeReader1.BarcodeStatusCodeString(i)
Next i
```

BarcodeToDecode

Description

Number of bar code instances to decode in the image.

Syntax

control.**BarcodeToDecode** [= *Value* as Integer]

Type

Integer

Range

Minimum:1

Maximum:5

Example

```
Dim BarcodesToDecode As Integer
Dim BarcodesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim ErrorString As Long
Dim i As Integer
' Configure the tool to read 4 barcodes in the image
BarcodeReader1.BarcodeToDecode = 4
BarcodeReader1.Execute
...
' Get the results
BarcodesToDecode = BarcodeReader1.BarcodeToDecode
BarcodesFound = BarcodeReader1.BarcodeFoundCount
' Get the strings of the decoded barcodes
For i = 0 To BarcodesFound - 1
    ResultString = BarcodeReader1.BarcodeString(i)
Next i
' Get the error code of the undecoded barcodes
For i = BarcodesFound To BarcodesToDecode - 1
    ErrorCode = BarcodeReader1.BarcodeStatusCode(i)
    ErrorString = BarcodeReader1.BarcodeStatusCodeString(i)
Next i
```

BarcodeType

Description

Bar code type of a given bar code instance. Read only.

Syntax

control.**BarcodeType**(*Index* As Integer)

Type

hsBarcodeType

Range

The valid range for this property is as follows:

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology
hsBarcodeISBN10	12	ISBN-10 symbology
hsBarcodeISBN13	13	ISBN-13 symbology
hsBarcodeUPCA	14	UPC-A symbology
hsBarcodeQR	15	QR Code symbology
hsBarcodePartial	16	Partial decoding only

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```

Dim CodeType As Integer
CodeType = BarcodeReader1.BarcodeType(0)
' Return an error if not a Code 39 barcode
If CodeType <> hsBarcodeCode39 Then
    CodeString = "Invalid Code Type"
Else
    CodeString = BarcodeReader1.BarcodeString(0)
Endif

```

BarcodeTypeEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **BarcodeTypeEnabled** is set to True for a given bar code type, the tool will attempt locate/decode bar codes of this type. Most types can be auto-detected and enabled by a successful **Learn** however BC412, Postnet, or Pharmacode symbologies must be enabled *before* performing a **Learn**. More than one bar code type can be enabled simultaneously, but in that case, the following properties must be set broadly enough to take into account all the enabled types: **HorizontalProbeSpacing**, **VerticalProbeSpacing**, **MaximumBarWidth**, **MinimumBarWidth**, **MaximumCodeHeight**, **MinimumCodeHeight**, **MaximumCodeLength**, **MinimumCodeLength**, **MaximumNumberOfBars**, and **MinimumEdgeStrength**.

Syntax

control.BarcodeTypeEnabled(*BarcodeType* As hsBarcodeType) [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The tool is configured to decode bar codes of the type specified by the <i>BarcodeType</i> parameter.
False	The tool is configured to ignore bar codes of the type specified by the <i>BarcodeType</i> parameter.

Parameter(s)

Name	Description	Type	Range
<i>BarcodeType</i>	Type of bar code to configure.	hsBarcodeType	Enum. See table below.

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology

Example

```
' Search for a Code 39 barcode in the image
BarcodeReader1.BarcodeTypeEnabled(hsBarcodeCode39) = True
BarcodeReader1.Execute
```

BarcodeTypeString

Description

Name of the bar code type of a given bar code instance. Read only.

Syntax

control.**BarcodeTypeString** (*Index* As Integer)

Type

String

Range

Not applicable

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

Example

```
Dim CodeType As Integer
Dim CodeTypeString As String
Dim CodeString As String
' Get the type and contents of the first barcode
CodeType = BarcodeReader1.BarcodeType(0)
CodeTypeString = BarcodeReader1.BarcodeTypeString(0)
CodeString = BarcodeReader1.BarcodeString(0)
```

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
Result = BarcodeReader1.BilinearInterpolationEnabled
BarcodeReader2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = BarcodeReader1.CalibratedUnitsEnabled
BarcodeReader2.CalibratedUnitsEnabled = False
```

ChecksumEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **ChecksumEnabled** is set to True, a BC412 bar code must contain a checksum character to be correctly decoded. When set to False, BC412 bar codes without a checksum character can also be decoded. This property is specific to BC412 bar codes; setting the **ChecksumEnabled** property for a bar code type other than BC412 has no effect.

Syntax

control.**ChecksumEnabled**(*BarcodeType* As hsBarcodeType) [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The bar code must contain a checksum character to be correctly decoded.
False	Bar codes without a checksum character can also be decoded.

Parameter(s)

Name	Description	Type	Range
<i>BarcodeType</i>	Type of bar code to configure.	hsBarcodeType	Enum. See table below.

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology

Example

```
' Search for a BC412 barcode without a checksum character
BarcodeReader1.BarcodeTypeEnabled(hsBarcodeBC412) = True
BarcodeReader1.BarcodeChecksumEnabled(hsBarcodeBC412) = False
BarcodeReader1.Execute
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= *Value* as hsCoordinateSystem]

Type

hsCoordinateSystem

Range

hsCoordinateSystem	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.

hsCoordinateSystem	Value	Description
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```

Dim CurrentSystem As hsCoordinateSystem
' Get the current coordinate system
CurrentSystem = BarcodeReader1.CoordinateSystem
' Express the results in the Object coordinate system
BarcodeReader1.CoordinateSystem = hsCoordinateSystemObject

```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

DecodeLeftToRightEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **DecodeLeftToRightEnabled** is set to True, a BC412 bar code is decoded from left to right. When set to False, a BC412 bar code is decoded from right to left. This mode is only useful when **StartStopEnabled** is set to False. This mode should not be used for SEMI BC412 bar codes that contain both Start/Stop and Checksum. This property is specific to BC412 bar codes; setting the **DecodeLeftToRightEnabled** property for a bar code type other than BC412 has no effect.

Syntax

control.**DecodeLeftToRightEnabled**(BarcodeType As hsBarcodeType) [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The bar code is decoded from left to right.
False	The bar code is decoded from right to left.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the bar code instance.	Integer	Minimum: 0 Maximum: BarcodeToDecode - 1

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology

Example

```
' Decode a BC412 barcode from right to left
BarcodeReader1.BarcodeTypeEnabled(hsBarcodeBC412) = True
BarcodeReader1.DecodeLeftToRightEnabled(hsBarcodeBC412) = False
BarcodeReader1.Execute
```

Description**Description**

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Info As String
Info = BarcodeReader1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: Unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = BarcodeReader1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & BarcodeReader1.ErrorObject.Description
```

HorizontalProbeSpacing

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Sets the horizontal distance between bar code scanning probes, in pixels. This property is auto-configured after a successful **Learn**.

Syntax

control.**HorizontalProbeSpacing** [= *Value* As Integer]

Type

Integer

Range

Minimum: 8

Maximum: 256

Example

```
Dim HorizontalProbe As Integer
' Get the current probe spacing
HorizontalProbe = BarcodeReader1.HorizontalProbeSpacing
```

InputImage

Description

Name of the Image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = BarcodeReader1.InputImage
BarcodeReader2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the Image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = BarcodeReader1.InputImage
BarcodeReader2.InputImage = "image1"
```

InputView

Description

Name of the View from which the Image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = BarcodeReader1.InputView
BarcodeReader2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View from which the Image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = BarcodeReader1.InputView
BarcodeReader2.InputView = "view0"
```

InputInstanceScene

Description

Name of the scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.InputInstanceScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = BarcodeReader1.InputInstanceScene
```

```
BarcodeReader2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

```
control.InputInstanceView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = BarcodeReader1.InputInstanceView
BarcodeReader2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

```
control.InputModelScene [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = BarcodeReader1.InputModelScene
BarcodeReader2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details.

Syntax

```
control.InputModelView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = BarcodeReader1.InputModelView
```

```
BarcodeReader2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = BarcodeReader1.InstanceSelectionEnabled
BarcodeReader2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = BarcodeReader1.InstanceSelectionIndex
BarcodeReader2.InstanceSelectionIndex = 0
```

MaximumBarWidth

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The maximum width, in pixels, of the bars in the bar code to be decoded. This property is auto-configured after a successful **Learn**.

Syntax

control.**MaximumBarWidth** [= *Value* As Integer]

Type

Integer

Range

Minimum: Greater than or equal to **MinimumBarWidth**.

Maximum: 100

Example

```
Dim MinBarWidth As Integer
Dim MaxBarWidth As Integer
' Get the current bar width range
MinBarWidth = BarcodeReader1.MinimumBarWidth
MaxBarWidth = BarcodeReader1.MaximumBarWidth
' Set the bar width range
BarcodeReader1.MinimumBarWidth = 3
BarcodeReader1.MaximumBarWidth = 25
```

MaximumCodeHeight

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The maximum height, in pixels, of the bar code to be decoded. This property is auto-configured after a successful **Learn**.

Syntax

control.**MaximumCodeHeight** [= *Value* As Integer]

Type

Integer

Range

Minimum: Greater than or equal to **MinimumCodeHeight**.

Maximum: 1024

Example

```
Dim MinCodeHeight As Integer
Dim MaxCodeHeight As Integer
' Get the current code height range
MinCodeHeight = BarcodeReader1.MinimumCodeHeight
MaxCodeHeight = BarcodeReader1.MaximumCodeHeight
' Set the code height range
BarcodeReader1.MinimumCodeHeight = 3
BarcodeReader1.MaximumCodeHeight = 25
```

MaximumCodeLength

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The maximum length, in pixels, of the bar code to be decoded. This property is auto-configured after a successful **Learn**.

Syntax

control.**MaximumCodeLength** [= *Value* As Integer]

Type

Integer

Range

Minimum: Greater than or equal to **MinimumCodeLength**.

Maximum: 1024

Example

```
Dim MinCodeLength As Integer
Dim MaxCodeLength As Integer
' Get the current code length range
MinCodeHeight = BarcodeReader1.MinimumCodeLength
MaxCodeHeight = BarcodeReader1.MaximumCodeLength
' Set the code length range
BarcodeReader1.MinimumCodeLength = 3
BarcodeReader1.MaximumCodeLength = 25
```

MaximumNumberOfBars

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The maximum number of bars in the bar code to be decoded. This property is auto-configured after a successful **Learn**.

Syntax

control.**MaximumNumberOfBars** [= *Value* As Integer]

Type

Integer

Range

Minimum: Greater than or equal to **MinimumNumberOfBars**.

Maximum: 256

Example

```
Dim MinBarCount As Integer
Dim MaxBarCount As Integer
' Get the current bar count range
MinBarCount = BarcodeReader1.MinimumNumberOfBars
MaxBarCount = BarcodeReader1.MaximumNumberOfBars
```

```
' Set the bar count range
BarcodeReader1.MinimumNumberOfBars = 3
BarcodeReader1.MaximumNumberOfBars = 25
```

MinimumBarWidth

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The minimum width, in pixels, of the bars in the bar code to be decoded. This property is auto-configured after a successful **Learn**.

Syntax

```
control.MinimumBarWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: 1

Maximum: Lower than or equal to **MaximumBarWidth**.

Example

```
Dim MinBarWidth As Integer
Dim MaxBarWidth As Integer
' Get the current bar width range
MinBarWidth = BarcodeReader1.MinimumBarWidth
MaxBarWidth = BarcodeReader1.MaximumBarWidth
' Set the bar width range
BarcodeReader1.MinimumBarWidth = 3
BarcodeReader1.MaximumBarWidth = 25
```

MinimumCodeHeight

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The maximum height, in pixels, of the bar code to be decoded. This property is auto-configured after a succesful **Learn**.

Syntax

```
control.MinimumCodeHeight [= Value As Integer]
```

Type

Integer

Range

Minimum: 4

Maximum: Lower than or equal to **MaximumCodeHeight**.

Example

```
Dim MinCodeHeight As Integer
```

```

Dim MaxCodeHeight As Integer
' Get the current code height range
MinCodeHeight = BarcodeReader1.MinimumCodeHeight
MaxCodeHeight = BarcodeReader1.MaximumCodeHeight
' Set the code height range
BarcodeReader1.MinimumCodeHeight = 3
BarcodeReader1.MaximumCodeHeight = 25

```

MinimumCodeLength

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The minimum length, in pixels, of the bar code to be decoded. This property is auto-configured after a succesful **Learn**.

Syntax

control.**MinimumCodeLength** [= *Value* As Integer]

Type

Integer

Range

Minimum: 20

Maximum: Lower than or equal to **MaximumCodeLength**.

Example

```

Dim MinCodeLength As Integer
Dim MaxCodeLength As Integer
' Get the current code length range
MinCodeHeight = BarcodeReader1.MinimumCodeLength
MaxCodeHeight = BarcodeReader1.MaximumCodeLength
' Set the code length range
BarcodeReader1.MinimumCodeLength = 3
BarcodeReader1.MaximumCodeLength = 25

```

MinimumEdgeStrength

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The minimum contrast strength required for a bar edge to be detected in the image.

Syntax

control.**MinimumEdgeStrength** [= *Value* As Integer]

Type

Integer

Range

Minimum: 5

Maximum: 100

Example

```
Dim MinEdgeStrength As Integer
' Get the current minimum edge strength
MinEdgeStrength = BarcodeReader1.MinimumEdgeStrength
' Set the minimum edge strength
BarcodeReader1.MinimumEdgeStrength = 20
```

MinimumNumberOfBars

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The minimum number of bars in the bar code to be decoded. This property is auto-configured after a succesful [Learn](#).

Syntax

control.**MinimumNumberOfBars** [= *Value* As Integer]

Type

Integer

Range

Minimum: 2

Maximum: Lower than or equal to **MaximumNumberOfBars**.

Example

```
Dim MinBarCount As Integer
Dim MaxBarCount As Integer
' Get the current bar count range
MinBarCount = BarcodeReader1.MinimumNumberOfBars
MaxBarCount = BarcodeReader1.MaximumNumberOfBars
' Set the bar count range
BarcodeReader1.MinimumNumberOfBars = 3
BarcodeReader1.MaximumNumberOfBars = 25
```

MinimumQuietZone

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Minimum size, in pixels, of the bar code quiet zone. This quiet zone is a clear space surrounding the bar code, without text or clutter of any kind.

Syntax

control.**MinimumQuietZone** [= *Value* As Integer]

Type

Integer

Range

Minimum: 5

Maximum: 100

Example

```
Dim MinQuietZone As Integer
' Get the current minimum quiet zone
MinQuietZone = BarcodeReader1.MinimumQuietZone
' Set the minimum quiet zone
BarcodeReader1.MinimumQuietZone = 20
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = BarcodeReader1.ModelTransformEnabled
BarcodeReader2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = BarcodeReader1.Name
BarcodeReader2.Name = "BcReaderA"
```

Polarity

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Sets the polarity for a BC412 bar code. This property is specific to BC412 bar codes; setting the **Polarity** property for a bar code type other than BC412 has no effect.

Syntax

control.**Polarity**(BarcodeType As hsBarcodeType) [= Value As hsBarcodePolarity]

Type

hsBarcodePolarity

Range

hsBarcodePolarity	Value	Description
hsBarcodePolarityLightOnDark	0	Light barcode on a dark background
hsBarcodePolarityDarkOnLight	1	Dark barcode on a light background

Parameter(s)

Name	Description	Type	Range
<i>BarcodeType</i>	Type of bar code to configure.	hsBarcodeType	Enum. See table below.

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology

Example

```
' Search for a BC412 light barcode on a dark background
BarcodeReader1.BarcodeTypeEnabled(hsBarcodeBC412) = True
BarcodeReader1.PolarityEnabled(hsBarcodeBC412) = True
BarcodeReader1.Polarity(hsBarcodeBC412) = _
```

hsBarcodePolarityLightOnDark
BarcodeReader1.Execute

PolarityEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **PolarityEnabled** is set to True, the tool only considers BC412 bar codes of the polarity is set by the **Polarity** property. When polarity is not known, **PolarityEnabled** should be set to False. This property is specific to BC412 bar codes; setting the **Polarity** property for a bar code type other than BC412 has no effect.

Syntax

control.PolarityEnabled(BarcodeType As hsBarcodeType) [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The tool only considers BC412 bar codes of the polarity set by the Polarity property
False	The tool ignores the Polarity property.

Parameter(s)

Name	Description	Type	Range
<i>BarcodeType</i>	Type of bar code to configure.	hsBarcodeType	Enum. See table below.

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology

Example

```
' Search for a BC412 light barcode on a dark background
BarcodeReader1.BarcodeTypeEnabled(hsBarcodeBC412) = True
BarcodeReader1.PolarityEnabled(hsBarcodeBC412) = True
BarcodeReader1.Polarity(hsBarcodeBC412) =
    hsBarcodePolarityLightOnDark
BarcodeReader1.Execute
```

ProcessingAlgorithm

Description

Activates one of the optional enhancement preprocessing algorithms that can be applied to the input image prior to the decoding process. The selected preprocessing algorithm is also applied prior to a **Learn**.

Syntax

```
control.ProcessingAlgorithm [= Value As hsBarcodeProcessing]
```

Type

hsBarcodeProcessing

Range

The valid range for this property is as follows:

hsBarcodeProcessing	Value	Description
hsBarcodeProcessingNone	0	No pre-processing is applied.
hsBarcodeProcessingInvert	1	Reverses the polarity of the image.
hsBarcodeProcessingErode	2	Erodes light objects.
hsBarcodeProcessingDilate	3	Dilates light objects.
hsBarcodeProcessingOpen	4	Removes small light defects.
hsBarcodeProcessingClose	5	Removes small dark defects.
hsBarcodeProcessingGaussian	6	Smooths the edges in the image.
hsBarcodeProcessingSharpen	7	Enhances the edges in the image.
hsBarcodeProcessingSharpenLow	8	Enhances the edges in the image after a low-pass filter is applied.
hsBarcodeProcessingMedian	9	Removes noise with a median filter.
hsBarcodeProcessingHistogramStretching	10	Stretches the histogram to increase contrast.

Example

```
' Filter out high frequency noise in the image prior to
' reading the code
BarcodeReader1.ProcessingAlgorithm = hsBarcodeProcessingMedian
BarcodeReader1.ProcessingSize = 5
BarcodeReader1.Execute
```

ProcessingSize

Description

Sets the **Size** parameter that is used by the preprocessing algorithm selected by the **ProcessingAlgorithm** property.

Syntax

control.**ProcessingSize** [= Value As Integer]

Type

Integer

Range

The range depends on the value of **ProcessingAlgorithm**:

hsBarcodeProcessing	Range	Description
hsBarcodeProcessingNone	Not applicable.	Not applicable.
hsBarcodeProcessingInvert	Not applicable.	Not applicable.
hsBarcodeProcessingErode	Greater than 0	Size of neighborhood
hsBarcodeProcessingDilate	Greater than 0	Size of neighborhood
hsBarcodeProcessingOpen	Greater than 0	Size of neighborhood
hsBarcodeProcessingClose	Greater than 0	Size of neighborhood
hsBarcodeProcessingGaussian	3,5,7	Kernel size
hsBarcodeProcessingSharpen	3,5,7	Kernel size
hsBarcodeProcessingSharpenLow	3,5,7	Kernel size
hsBarcodeProcessingMedian	3,5,7	Kernel size
hsBarcodeProcessingHistogramStretching	Not applicable.	Not applicable.

Example

```
' Filter out high frequency noise in the image prior to
' reading the code
BarcodeReader1.ProcessingAlgorithm = hsBarcodeProcessingMedian
BarcodeReader1.ProcessingSize = 5
BarcodeReader1.Execute
```

SamplingStep

Description

Step used in the tool's last execution to sample the region of interest from the input image. When the tool is model-based, this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = BarcodeReader1.SamplingStep
```

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = BarcodeReader1.SamplingStepCustom
BarcodeReader2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default optimal sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Example

```
Dim Enabled As Boolean
```

```
Enabled = BarcodeReader1.SamplingStepCustomEnabled
BarcodeReader2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

```
control.SamplingStepDefault
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Example

```
Dim Sampling As Single
Sampling = BarcodeReader1.SamplingStepDefault
```

SearchDirection

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Sets the direction used by the tool to scan the image for bar code instances.

Syntax

```
control.SearchDirection [= Value As hsBarcodeSearchDirection]
```

Type

hsBarcodeSearchDirection

Range

The valid range for this property is as follows:

hsBarcodeSearchDirection	Value	Description
hsBarcodeDirectionHorizontal	0	Barcode scanned horizontally only.
hsBarcodeDirectionVertical	1	Barcode scanned vertically only.
hsBarcodeDirectionHorizontalVertical	2	Barcode scanned horizontally then vertically.
hsBarcodeDirectionVerticalHorizontal	3	Barcode scanned vertically then horizontally.

Example

```
' Search for horizontal and vertical barcode instances
BarcodeReader1.SearchDirection = _
    hsBarcodeDirectionHorizontalVertical
BarcodeReader1.Execute
```


StartStopEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **StartStopEnabled** is set to True, a BC412 bar code must contain a start/stop pattern to be correctly decoded. When set to False, BC412 bar codes without a start/stop pattern can also be decoded. This property is specific to BC412 bar codes; setting the **StartStopEnabled** property for a bar code type other than BC412 has no effect.

Syntax

control.**StartStopEnabled**(BarcodeType As hsBarcodeType) [= Value As Boolean]

Type

Value	Description
True	The bar code must contain a start/stop pattern to be correctly decoded.
False	Bar codes without a start/stop pattern can also be decoded.

Parameter(s)

Name	Description	Type	Range
<i>BarcodeType</i>	Type of bar code to configure.	hsBarcodeType	Enum. See table below.

hsBarcodeType	Value	Description
hsBarcodeCode128	0	Code 128 symbology
hsBarcodeCode39	1	Code 39 symbology
hsBarcodeCode93	2	Code 93 symbology
hsBarcodeI2of5	3	I2 of 5 symbology
hsBarcodeCodabar	4	Codabar symbology
hsBarcodeUPCEAN	5	UPC/EAN symbology
hsBarcodeUPCE	6	UPC-E symbology
hsBarcodeUPCSupplemental	7	UPC Supplemental symbology
hsBarcodePDF417	8	PDF417 symbology
hsBarcodeBC412	9	BC412 symbology
hsBarcodePostnet	10	Postnet symbology
hsBarcodePharmacode	11	Pharmacode symbology

Example

```
' Search for a BC412 barcode without a start/stop pattern
BarcodeReader1.BarcodeTypeEnabled(hsBarcodeBC412) = True
BarcodeReader1.BarcodeStartStopEnabled(hsBarcodeBC412) = False
BarcodeReader1.Execute
```

Timeout

Description

Timeout, in milliseconds, after which the tool aborts its process.

Syntax

control.Timeout [= *Value* As Long]

Type

Long

Range

Greater than zero.

Example

```
Dim Timeout As Integer
Timeout = BarcodeReader1.Timeout
BarcodeReader2.Timeout = 250
```

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.ToolHeight [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = BarcodeReader1.ToolHeight
BarcodeReader2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.ToolPositionX [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = BarcodeReader1.ToolPositionX
BarcodeReader2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= *Value* as Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = BarcodeReader1.ToolPositionY
BarcodeReader2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* as Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = BarcodeReader1.ToolRotation
BarcodeReader2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= *Value* as Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = BarcodeReader1.ToolWidth
BarcodeReader2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = BarcodeReader1.TransformRotation
BarcodeReader2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = BarcodeReader1.TransformScale
```

BarcodeReader2.TransformScale = 1.10

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = BarcodeReader1.TransformTranslationX
BarcodeReader2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = BarcodeReader1.TransformTranslationY
BarcodeReader2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSBarcodeReader"

Example

```
Dim ControlType As String
ControlType = BarcodeReader1.TypeName
```

UnitsLength

Description

Unit used to express lengths and coordinates.

Syntax

control.UnitsLength [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = BarcodeReader1.UnitsLength
BarcodeReader2.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Barcode1.UnitsOrientation
Barcode2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.2.000 R 3217 (release)".

Example

```
Dim Version As String
Version = BarcodeReader1.Version
```

VerticalProbeSpacing

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Sets the vertical distance between bar code scanning probes, in pixels. This property is auto-configured after a succesful **Learn**.

Syntax

control.**VerticalProbeSpacing** [= *Value* As Integer]

Type

Integer

Range

Minimum: 8

Maximum: 256

Example

```
Dim VerticalProbe As Integer
' Get the current probe spacing
VerticalProbe = BarcodeReader1.VerticalProbeSpacing
```

Methods

AboutBox

Description

Displays the Barcode Reader tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Range

No return value.

Example

```
BarcodeReader1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.

Syntax

control.**Execute**

Parameter(s)

None

Return

Type	Description	Range
Single	Processing time in milliseconds	Greater than 0

Example

```
BarcodeReader1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the property pages of the tool if they are displayed.

Syntax

control.**HideProperties**

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = BarcodeReader1.HideProperties
```

Learn

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Auto-detects the type of bar code currently present in the image and optimizes the detection parameters accordingly. After a successful call to **Learn** (True), the type of bar code detected in the image is automatically enabled (see **BarcodeTypeEnabled**) and the following detection parameters are optimized: **HorizontalProbeSpacing**, **VerticalProbeSpacing**, **MaximumBarWidth**, **MinimumBarWidth**, **MaximumCodeHeight**, **MinimumCodeHeight**, **MaximumCodeLength**, **MinimumCodeLength**, and **MaximumNumberOfBars**. BC412, Postnet, or Pharmacode symbologies are not auto-detected by a **Learn** and must be enabled using the **BarcodeTypeEnabled** property before calling the **Learn** method.

Syntax

control.**Learn**() As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Status of the Learn process	True: Successful False: Failed

Example

```
Dim SuccessfulLearn As Boolean
SuccessfulLearn = BarcodeReader1.Learn
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Shows the tool's property pages.

Syntax

control.**ShowProperties**(*Modal* as Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

<i>Option</i> parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = BarcodeReader1.ShowProperties(True, _
    hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Blob Analyzer

Description

The Blob Analyzer finds, labels and analyzes irregular shaped objects. This tool detects and computes intrinsic/extrinsic geometric and greylevel properties of blobs that meet user-defined criteria.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the Image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
Result = Blob1.BilinearInterpolationEnabled
Blob2.BilinearInterpolationEnabled = True
```

BlobArea

Description

Area of a given blob. Read only.

Syntax

control.**BlobArea**(*Index* As Integer)

Type

Single

Range

Minimum: **MinimumBlobArea**
Maximum: **MaximumBlobArea**

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property is expressed in the squared length units specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel squared
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property squared
ModelTransformEnabled = True	UnitsLength property squared

Example

```
Dim Area As Single
Area = Blob1.BlobArea(0)
```

BlobBoundingBoxBottom**Description**

The bottommost coordinate of the bounding box aligned with respect to the X-axis of the Tool coordinate system. This value is returned with respect to the selected coordinate system. Read only.

Syntax

control.**BlobBoundingBoxBottom**(*Index* As Integer)

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property

Mode	Length unit
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxBottom As Single
BoundingBoxBottom = Blob1.BoundingBoxBottom(0)
```

BlobBoundingBoxCenterX

Description

X-coordinate of the center of the bounding box aligned with the Tool coordinate system. This value is returned with respect to the selected coordinate system. Read only.

Syntax

control.**BlobBoundingBoxCenterX**(*Index* As Integer)

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxCenterX As Single
BoundingBoxCenterX = Blob1.BlobBoundingBoxCenterX(0)
```

BlobBoundingBoxCenterY

Description

Y-coordinate of the center of the bounding box aligned with the Tool coordinate system. This value is returned with respect to the selected coordinate system. Read only.

Syntax

control.**BlobBoundingBoxCenterY**(*Index* As Integer)

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxCenterY As Single
BoundingBoxCenterX = Blob1.BlobBoundingBoxCenterY(0)
```

BlobBoundingBoxHeight**Description**

Height of the bounding box with respect to the Y-axis of the Tool coordinate system. Read only.

Syntax

control.**BlobBoundingBoxHeight**(*Index* As Integer)

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property

Mode	Length unit
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxHeight As Single
BoundingBoxHeight = Blob1.BlobBoundingBoxHeight(0)
```

BlobBoundingBoxLeft

Description

The leftmost coordinate of the bounding box aligned with respect to the X-axis of the Tool coordinate system. This value is returned with respect to the selected coordinate system. Read only.

Syntax

control.**BlobBoundingBoxLeft**(*Index* As Integer)

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxLeft As Single
BoundingBoxLeft = Blob1.BlobBoundingBoxLeft(0)
```

BlobBoundingBoxRight

Description

The rightmost coordinate of the bounding box aligned with respect to the X-axis of the Tool coordinate system. Read only.

Syntax

control.**BlobBoundingBoxRight**(*Index* As Integer)

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxRight As Single
BoundingBoxRight = Blob1.BlobBoundingBoxRight(0)
```

BlobBoundingBoxRotation

Description

Rotation of the bounding box with respect to the X-axis of the selected coordinate system.
Read only.

Syntax

```
control.BlobBoundingBoxRotation(Index As Integer)
```

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property

UnitsOrientation property value	BlobBoundingBoxRotation property range
hsDegree	Minimum: -180 deg Maximum: 180 deg
hsRadians	Minimum: -Pi rad Maximum: Pi rad

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Example

```
Dim BoundingBoxRotation As Single
BoundingBoxRotation = Blob1.BlobBoundingBoxRotation(0)
```

BlobBoundingBoxTop**Description**

The topmost coordinate of the bounding box aligned with respect to the X-axis of the Tool coordinate system. This value is returned with respect to the selected coordinate system. Read only.

Syntax

```
control.BlobBoundingBoxTop(Index As Integer)
```

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxTop As Single
BoundingBoxTop = Blob1.BlobBoundingBoxTop(0)
```

BlobBoundingBoxWidth**Description**

Width of the bounding box with respect to the X-axis of the Tool coordinate system. Read only.

Syntax

```
control.BlobBoundingBoxWidth(Index As Integer)
```

Type

Single

Range

Boundaries of the input Image.

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BoundingBoxWidth As Single
BoundingBoxWidth = Blob1.BlobBoundingBoxWidth(0)
```

BlobChainCode**Description**

Direction of a given boundary element associated to the chain code. This direction can only be expressed with respect to the Tool coordinate system. Read only.

Syntax

control.**BlobChainCode**(*Position* As Long, *Index* As Integer)

Type

hsChainCodeDirection

Range

The valid range for this property is as follows:

Names	Value	Description
hsDirectionRight	0	Right direction
hsDirectionTop	1	Top direction
hsDirectionLeft	2	Left direction
hsDirectionBottom	3	Bottom direction

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount - 1
<i>Position</i>	Index of the selected boundary element	Long	Minimum: 0 Maximum: BlobChainCodeLength - 1

Example

```
Dim ChainCodeDirection As hsChainCodeDirection
ChainCodeDirection = Blob1.BlobChainCode(0,0)
```

BlobChainCodeDeltaX**Description**

Horizontal length of a boundary element associated to the chain code. Read only.

Syntax

control.**BlobChainCodeDeltaX**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim ChainCodeDeltaX As Single
ChainCodeDeltaX = Blob1.BlobChainCodeDeltaX(0)
```

BlobChainCodeDeltaY**Description**

Vertical length of a boundary element associated to the chain code. Read only.

Syntax

control.**BlobChainCodeDeltaY**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim ChainCodeDeltaY As Single
ChainCodeDeltaY = Blob1.BlobChainCodeDeltaY(0)
```

BlobChainCodeLength

Description

Number of boundary elements in the chain code of the blob. Read only.

Syntax

control.**BlobChainCodeLength**(*Index* As Integer)

Type

Long

Range

Minimum: Greater than 4

Maximum: unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim ChainCodeLength As Long
ChainCodeLength = Blob1.BlobChainCodeLength(0)
```

BlobChainCodeStartX

Description

X position of the first pixel associated to the chain code according to the Tool coordinate system. Read only.

Syntax

control.**BlobChainCodeStartX**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim ChainCodeStartX As Single
ChainCodeStartX = Blob1.BlobChainCodeStartX(0)
```

BlobChainCodeStartY

Description

Y position of the first pixel associated to the chain code according to the Tool coordinate system. Read only.

Syntax

control.**BlobChainCodeStartY**(*Index* As Integer)

Type

Integer

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim ChainCodeStartY As Single
ChainCodeStartY = Blob1.BlobChainCodeStartY(0)
```

BlobConvexPerimeter

Description

Convex perimeter of the selected blob. Read only.

Syntax

control.**BlobConvexPerimeter** (*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0

Maximum: unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim Perimeter As Single
Perimeter = Blob1.BlobConvexPerimeter(2)
```

BlobCount

Description

Number of blobs detected by the tool. Read only.

Syntax

control.**BlobCount**

Type

Integer

Range

Minimum: 0
Maximum: 65534

Example

```
Dim Count As Integer
Count = Blob1.BlobCount
```

BlobElongation

Description

Ratio of the moment of inertia about the blob's minor axis (**BlobInertiaMaximum**) to the moment of inertia about the blob's major axis (**BlobInertiaMinimum**). Read only.

Syntax

control.**BlobElongation**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

No units

Example

```
Dim BlobElongation As Single
BlobElongation = Blob1.BlobElongation(0)
```

BlobExtentBottom

Description

Distance along the Y-axis between the blob's center of mass and the bottom side of the bounding box. Read only.

Syntax

control.**BlobExtentBottom**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobExtentBottom As Single
BlobExtentBottom = Blob1.BlobExtentBottom(0)
```

BlobExtentLeft**Description**

Distance along the X-axis between the blob's center of mass and the left side of the bounding box. Read only.

Syntax

control.**BlobExtentLeft**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property

Mode	Length unit
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobExtentLeft As Single
BlobExtentLeft = Blob1.BlobExtentLeft(0)
```

BlobExtentRight

Description

Distance along the X-axis between the blob's center of mass and the right side of the bounding box. Read only.

Syntax

control.**BlobExtentRight**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobExtentRight As Single
BlobExtentRight = Blob1.BlobExtentRight(0)
```

BlobExtentTop

Description

Distance along the Y-axis between the blob's center of mass and the top side of the bounding box. Read only.

Syntax

control.**BlobExtentTop**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobExtentTop As Single
BlobExtentTop = Blob1.BlobExtentTop(0)
```

BlobGreyLevelMaximum

Description

Highest greylevel value in the blob. Read only.

Syntax

```
control.BlobGreyLevelMaximum(Index As Integer)
```

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim BlobGreyLevelMaximum As Integer
BlobGreyLevelMaximum = Blob1.BlobGreyLevelMaximum(0)
```

BlobGreyLevelMean

Description

Mean greylevel value of the blob. Read only.

Syntax

control.**BlobGreyLevelMean**(*Index* As Integer)

Type

Single

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim BlobGreyLevelMean As Integer
BlobGreyLevelMean = Blob1.BlobGreyLevelMean(0)
```

BlobGreyLevelMinimum

Description

Lowest greylevel value in the blob. Read only.

Syntax

control.**BlobGreyLevelMinimum**(*Index* As Integer)

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim BlobGreyLevelMinimum As Integer
BlobGreyLevelMinimum = Blob1.BlobGreyLevelMinimum(0)
```

BlobGreyLevelRange

Description

Range of the greylevel values in the blob. [**BlobGreyLevelMaximum** - **BlobGreyLevelMinimum** + 1]. Read only.

Syntax

control.**BlobGreyLevelRange**(*Index* As Integer)

Type

Integer

Range

Minimum: 0

Maximum: 256

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim BlobGreyLevelRange As Integer
```

```
BlobGreyLevelRange = Blob1.BlobGreyLevelRange(0)
```

BlobGreyLevelStdDev

Description

Standard deviation of the greylevel values in the blob. Read only.

Syntax

```
control.BlobGreyLevelStdDev(Index As Integer)
```

Type

Single

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim BlobGreyLevelStdDev As Integer
```

```
BlobGreyLevelStdDev = Blob1.BlobGreyLevelStdDev(0)
```

BlobHoleCount

Description

The number of holes found in a blob. Read only.

Syntax

```
control.BlobHoleCount(Index As Integer)
```

Type

Long

Range

Minimum: 0

Maximum: Unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim HoleCount As Long
HoleCount = Blob1.BlobHoleCount(0)
```

BlobInertiaMaximum**Description**

Moment of inertia about the minor axis, which corresponds to the highest moment of inertia.
Read only.

Syntax

control.**BlobInertiaMaximum**(*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0
Maximum: Unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property is expressed in the length units, raised to the power of four, as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	(Pixel) ⁴
CalibratedUnitsEnabled = True ModelTransformEnabled = False	(UnitsLength property) ⁴
ModelTransformEnabled = True	(UnitsLength property) ⁴

Example

```
Dim BlobInertiaMaximum As Single
BlobInertiaMaximum = Blob1.BlobInertiaMaximum(0)
```

BlobInertiaMinimum**Description**

Moment of inertia about the major axis, which corresponds to the lowest moment of inertia.
Read only.

Syntax

control.**BlobInertiaMinimum**(*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0

Maximum: Unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property is expressed in the length units, raised to the power of four, as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	(Pixel) ⁴
CalibratedUnitsEnabled = True ModelTransformEnabled = False	(UnitsLength property) ⁴
ModelTransformEnabled = True	(UnitsLength property) ⁴

Example

```
Dim BlobInertiaMinimum As Single
BlobInertiaMinimum = Blob1.BlobInertiaMinimum(0)
```

BlobInertiaXAxis

Description

Moment of inertia about the X-axis of the Tool coordinate system. Read only.

Syntax

control.**BlobInertiaXAxis**(*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0

Maximum: Unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property is expressed in the length units, raised to the power of four, as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	(Pixel) ⁴
CalibratedUnitsEnabled = True ModelTransformEnabled = False	(UnitsLength property) ⁴
ModelTransformEnabled = True	(UnitsLength property) ⁴

Example

```
Dim BlobInertiaXAxis As Single
BlobInertiaXAxis = Blob1.BlobInertiaXAxis(0)
```

BlobInertiaYAxis

Description

Moment of inertia about the Y-axis of the Tool coordinate system. Read only.

Syntax

control.**BlobInertiaYAxis**(*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0

Maximum: Unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property is expressed in the length units, raised to the power of four, as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	(Pixel) ⁴
CalibratedUnitsEnabled = True ModelTransformEnabled = False	(UnitsLength property) ⁴
ModelTransformEnabled = True	(UnitsLength property) ⁴

Example

```
Dim BlobInertiaYAxis As Single
BlobInertiaYAxis = Blob1.BlobInertiaYAxis(0)
```

BlobIntrinsicBoundingBoxBottom

Description

The bottommost coordinate of the bounding box with respect to the X-axis (major axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxBottom**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxBottom As Single
BlobIntrinsicBoundingBoxBottom = _
    Blob1.BlobIntrinsicBoundingBoxBottom(0)
```

BlobIntrinsicBoundingBoxCenterX

Description

X-coordinate of the center of the bounding box with respect to the X-axis (major axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxCenterX**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxCenterX As Single
BlobIntrinsicBoundingBoxCenterX = _
    Blob1.BlobIntrinsicBoundingBoxCenterX(0)
```

BlobIntrinsicBoundingBoxCenterY

Description

Y-coordinate of the center of the bounding box with respect to the Y-axis (minor axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxCenterY**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel

Mode	Length unit
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxCenterY As Single
BlobIntrinsicBoundingBoxCenterY = _
    Blob1.BlobIntrinsicBoundingBoxCenterY(0)
```

BlobIntrinsicBoundingBoxHeight

Description

Height of the bounding box with respect to the Y-axis (minor axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxHeight**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxHeight As Single
BlobIntrinsicBoundingBoxHeight = _
    Blob1.BlobIntrinsicBoundingBoxHeight(0)
```

BlobIntrinsicBoundingBoxLeft

Description

The leftmost coordinate of the bounding box aligned with respect to the X-axis (major axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxLeft**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxLeft As Single
BlobIntrinsicBoundingBoxLeft = _
    Blob1.BlobIntrinsicBoundingBoxLeft(0)
```

BlobIntrinsicBoundingBoxRight

Description

The rightmost coordinate of the bounding box aligned with the X-axis (major axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxRight**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxRight As Single
BlobIntrinsicBoundingBoxRight = _
    Blob1.BlobIntrinsicBoundingBoxRight(0)
```

BlobIntrinsicBoundingBoxRotation

Description

Rotation of the intrinsic bounding box with respect to the X-axis of the selected coordinate system. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxRotation**(*Index* As Integer)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property

UnitsOrientation property value	IntrinsicBlobBoundingBoxRotation property range
hsDegree	Minimum: -180 deg Maximum: 180 deg
hsRadians	Minimum: -Pi rad Maximum: Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim IntrinsicBoundingBoxRotation As Single
IntrinsicBoundingBoxRotation = _
    Blob1.BlobIntrinsicBoundingBoxRotation(0)
```

BlobIntrinsicBoundingBoxTop**Description**

The topmost coordinate of the bounding box aligned with the Y-axis (minor axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxTop**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxTop As Single
BlobIntrinsicBoundingBoxTop = _
    Blob1.BlobIntrinsicBoundingBoxTop(0)
```

BlobIntrinsicBoundingBoxWidth**Description**

Width of the bounding box with respect to the X-axis (major axis) of the principal axes. Read only.

Syntax

control.**BlobIntrinsicBoundingBoxWidth**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicBoundingBoxWidth As Single
BlobIntrinsicBoundingBoxWidth = _
    Blob1.BlobIntrinsicBoundingBoxWidth(0)
```

BlobIntrinsicExtentBottom

Description

Distance along the minor axis between the blob's center of mass and the bottom side of the intrinsic bounding box. Read only.

Syntax

control.**BlobIntrinsicExtentBottom**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel

Mode	Length unit
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicExtentBottom As Single
BlobIntrinsicExtentBottom = _
    Blob1.BlobIntrinsicExtentBottom(0)
```

BlobIntrinsicExtentLeft

Description

Distance along the major axis between the blob's center of mass and the left side of the intrinsic bounding box. Read only.

Syntax

control.**BlobIntrinsicExtentLeft** (*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicExtentLeft As Single
BlobIntrinsicExtentLeft = _
    Blob1.BlobIntrinsicExtentLeft(0)
```

BlobIntrinsicExtentRight

Description

Distance along the major axis between the blob's center of mass and the right side of the intrinsic bounding box. Read only.

Syntax

control.**BlobIntrinsicExtentRight**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicExtentRight As Single
BlobIntrinsicExtentRight = _
    Blob1.BlobIntrinsicExtentRight(0)
```

BlobIntrinsicExtentTop

Description

Distance along the major axis between the blob's center of mass and the top side of the intrinsic bounding box. Read only.

Syntax

control.**BlobIntrinsicExtentTop**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim BlobIntrinsicExtentTop As Single
BlobIntrinsicExtentTop = _
    Blob1.BlobIntrinsicExtentTop(0)
```

BlobPositionX

Description

X coordinate of the center of mass of a given blob in the currently selected coordinate system.
Read only.

Syntax

control.**BlobPositionX**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property

Mode	Length unit
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim PosX
PosX = Blob1.BlobPositionX(2)
```

BlobPositionY

Description

Y coordinate of the center of mass of a given blob in the currently selected coordinate system. Read only.

Syntax

control.**BlobPositionY**(*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim PosY
PosY = Blob1.BlobPositionY(2)
```

BlobPrincipalAxesRotation

Description

Angle of axis of the smallest moment of inertia with respect to the X-axis of the selected coordinate system. Read only.

Syntax

control.**BlobPrincipaAxesRotation**(*Index* As Integer)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property

Name	Value	Description
hsDegree	Minimum: >-90 deg Maximum: <90 deg	Results are expressed in degrees.
hsRadians	Minimum: >-Pi/2 Maximum: <Pi/2	Results are expressed in radians.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Example

```
Dim PrincipalAxesRotation As Single
PrincipalAxesRotation = _
    Blob1.BlobPrincipalAxesRotation(0)
```

BlobRawPerimeter**Description**

Raw perimeter of the selected blob. Read only.

Syntax

control.**BlobRawPerimeter** (*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0

Maximum: unlimited

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim Perimeter As Single
Perimeter = Blob1.BlobRawPerimeter(0)
```

BlobRoundness**Description**

The degree of similarity between the blob and a circle. The roundness is 1 for a perfectly circular blob. Read only.

Syntax

control.**BlobRoundness**(*Index* As Integer)

Type

Single

Range

Minimum: Greater than 0.0

Maximum: 1

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected blob.	Integer	Minimum: 0 Maximum: BlobCount -1

Remark(s)

No units

Example

```
Dim BlobRoundness As Single
BlobRoundness = Blob1.BlobRoundness(0)
```

CalibratedUnitsEnabled**Description**

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = Blob1.CalibratedUnitsEnabled
```

```
Blob2.CalibratedUnitsEnabled = False
```

ChainCodeResultsEnabled

Description

Enables the computation of the blob chain code properties: **BlobChainCode**, **BlobChainCodeDeltaX**, **BlobChainCodeDeltaY**, **BlobChainCodeLength**, **BlobChainCodeStartX** and **BlobChainCodeStartY**.

Syntax

```
control.ChainCodeResultsEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	The chain code properties will be computed
False	No chain code properties will be computed

Example

```
Dim ChainCodeResultsEnabled As Boolean
ChainCodeResultsEnabled = Blob1.ChainCodeResultsEnabled
Blob2.ChainCodeResultsEnabled = True
```

ClearOutputBlobImageEnabled

Description

Specifies if the image output by the tool will be cleared in the next execution of the tool.

Syntax

```
control.ClearOutputBlobImage Enabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
False	The output image of the blob will not be cleared.
True	The output image of the blob will be cleared.

Example

```
Dim Enabled As Boolean
Enabled = Blob1.ClearOutputBlobImageEnabled
Blob1.ClearOutputBlobImageEnabled = False
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

```
control.CoordinateSystem [= Value As hsCoordinateSystem]
```

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Name	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim CurrentSystem As hsCoordinateSystem
' Get the current coordinate system
CurrentSystem = Blob1.CoordinateSystem
' Express the results in the Object coordinate system
Blob1.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax*control*.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Blob1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax*control*.**ElapsedTime****Type**

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Blob1.ElapsedTime
```

ErrorObject

DescriptionReturns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.**Syntax***control*.**ErrorObject****Type**

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

ExtrinsicBoxResultsEnabled

DescriptionEnables the computation of bounding box and extent properties: **BlobBoundingBoxBottom**, **BlobBoundingBoxCenterX**, **BlobBoundingBoxCenterY**, **BlobBoundingBoxHeight**,

BlobBoundingBoxLeft, **BlobBoundingBoxRight**, **BlobBoundingBoxRotation**, **BlobBoundingBoxTop**, **BlobBoundingBoxWidth**, **BlobExtentBottom**, **BlobExtentLeft**, **BlobExtentRight** and **BlobExtentTop**.

Syntax

control.**ExtrinsicBoxResultsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The extrinsic bounding box properties will be computed.
False	No extrinsic bounding box properties will be computed.

Example

```
Dim ExtrinsicBoxResults As Boolean
ExtrinsicBoxResults = Blob1.ExtrinsicBoxResultsEnabled
Blob2.ExtrinsicBoxResultsEnabled = True
```

ExtrinsicInertiaResultsEnabled

Description

Enables the computation of the following blob properties: **BlobInertiaXAxis**, **BlobInertiaYAxis** and **BlobPrincipalAxesRotation**.

Syntax

control.**ExtrinsicInertiaResultsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The extrinsic inertia properties will be computed.
False	No extrinsic inertia properties will be computed.

Example

```
Dim ExtrinsicInertiaResultsEnabled As Boolean
ExtrinsicInertiaResultsEnabled = _
    Blob1.ExtrinsicInertiaResultsEnabled
Blob2.ExtrinsicInertiaResultsEnabled = True
```

GreyLevelResultsEnabled

Description

Enables the computation of the following blob greylevel properties: **BlobGreyLevelMaximum**, **BlobGreyLevelMean**, **BlobGreyLevelMaximum**, **BlobGreyLevelMinimum**, **BlobGreyLevelRange**, and **BlobGreyLevelStdDev**.

Syntax

control.**GreyLevelResultsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The grey-level blob properties will be computed.
False	No grey-level blob properties will be computed.

Example

```
Dim GreyLevelResultsEnabled As Boolean
GreyLevelResultsEnabled = Blob1.GreyLevelResultsEnabled
Blob2.GreyLevelResultsEnabled = True
```

HoleFillingEnabled

Description

Enables the filling of the holes in each blob.

Syntax

control.**HoleFillingEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	All holes will be filled.
False	No hole will be filled.

Example

```
Dim HoleFillingEnabled As Boolean
HoleFillingEnabled = Blob1.HoleFillingEnabled
Blob2.HoleFillingEnabled = True
```

InputImage

InputGreyScaleImage

Description

Name of the image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Blob1.InputImage
```

```
Blob2.InputImage = "image1"
```

InputView

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

```
control.InputView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Blob1.InputView
Blob2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

```
control.InputInstanceScene [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = Blob1.InputInstanceScene
Blob2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

```
control.InputInstanceView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Blob1.InputInstanceView
Blob2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = Blob1.InputModelScene
Blob2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details.

Syntax

control.**InputModelView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = Blob1.InputModelView
Blob2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

control.InstanceSelectionEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Enabled As Boolean
Enabled = Blob1.InstanceSelectionEnabled
Blob2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** properties are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.InstanceSelectionIndex [= Value As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = Blob1.InstanceSelectionIndex
Blob2.InstanceSelectionIndex = 0
```

IntrinsicBoxResultsEnabled

Description

Enables the computation of the following intrinsic bounding box and intrinsic extent:

BlobIntrinsicBoundingBoxBottom **BlobIntrinsicBoundingBoxCenterX**
BlobIntrinsicBoundingBoxCenterY, **BlobIntrinsicBoundingBoxHeight**,
BlobIntrinsicBoundingBoxLeft, **BlobIntrinsicBoundingBoxRight**,
BlobIntrinsicBoundingBoxRotation, **BlobIntrinsicBoundingBoxTop**,
BlobIntrinsicBoundingBoxWidth, **BlobIntrinsicExtentBottom**, **BlobIntrinsicExtentLeft**,
BlobIntrinsicExtentRight and **BlobIntrinsicExtentTop**.

Syntax

control.IntrinsicBoxResultsEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The intrinsic bounding box properties will be computed.
False	No intrinsic bounding box properties will be computed.

Example

```
Dim IntrinsicBoxResultsEnabled As Boolean
IntrinsicBoxResultsEnabled = _
    Blob1.IntrinsicBoxResultsEnabled
Blob2.IntrinsicBoxResultsEnabled = True
```

IntrinsicInertiaResultsEnabled

Description

Enables the computation of the following blob properties: **BlobInertiaMinimum**, **BlobInertiaMaximum** and **BlobElongation**.

Syntax

```
control.IntrinsicInertiaResultsEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	The intrinsic inertia properties will be computed.
False	No intrinsic inertia properties will be computed.

Example

```
Dim IntrinsicInertiaResultsEnabled As Boolean
IntrinsicInertiaResultsEnabled = _
    Blob1.IntrinsicInertiaResultsEnabled
Blob2.IntrinsicInertiaResultsEnabled = True
```

MaximumBlobArea

Description

Maximum area for a blob. This validation criterion is used to filter out unwanted blobs from the results.

Syntax

```
control.MaximumBlobArea [= Value As Single]
```

Type

Single

Range

Minimum: **MinimumBlobArea**
Maximum: area of the image

Remark(s)

If the tool is calibrated or model-based (**CalibratedUnitsEnabled** or **ModelTransformEnabled** set to True), this property is expressed in the **UnitsLength** property squared. Otherwise, this property is expressed in pixels squared.

Example

```
Dim MaxArea As Single
MaxArea = Blob1.MaximumBlobArea
Blob1.MaximumBlobArea = 100.0
```

MinimumBlobArea

Description

Minimum area for a blob. This validation criterion is used to filter out unwanted blobs from the results.

Syntax

control.**MinimumBlobArea** [= Value As Single]

Type

Single

Range

Minimum: 0.0

Maximum: **MaximumBlobArea**

Remark(s)

If the tool is calibrated or model-based (**CalibratedUnitsEnabled** or **ModelTransformEnabled** set to True), this property is expressed in the **UnitsLength** property squared. Otherwise, this property is expressed in pixels squared.

Example

```
Dim MinArea As Single
MinArea = Blob1.MinimumBlobArea
Blob1.MinimumBlobArea = 100.0
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using the **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = Blob1.ModelTransformEnabled
Blob2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = Blob1.Name
Blob2.Name = "BlobA"
```

OutputBlobImage

Description

Name of the blob image created by the tool. This image is optionally output according to the value of the **OutputBlobImageEnabled** property. It contains the result of the blob segmentation and labelling process.

Syntax

control.**OutputBlobImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Blob1.OutputBlobImage
Blob1.OutputBlobImage = "BlobImage0"
```

OutputBlobImageEnabled

Description

Specifies if a blob image will be output after the blob segmentation and labelling process.

Syntax

control.**OutputBlobImageEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The image will not be output.
True	The image will be output.

Remark(s)

Generating a blob image considerably increases the tool's execution time.

Example

```
dim Enable As Boolean
Enable = Blob1.OutputBlobImageEnabled
Blob1.OutputBlobImageEnabled = False
```

OutputBlobView

Description

Name of the View to which the Output Blob image is output. See **OutputBlobImage** for more details.

Syntax

control.**OutputBlobView** [= Name As String]

Type

String

Range

Not applicable.

Remark(s)**Example**

```
Dim ViewName As String
ViewName = Blob1.OutputBlobView
blob1.OutputBlobView = "BlobView0"
```

PerimeterResultsEnabled

Description

Enables the computation of the following blob properties: **BlobRawPerimeter**, **BlobConvexPerimeter** and **BlobRoundness**.

Syntax

control.**PerimeterResultsEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The perimeter properties will be computed.
False	No perimeter properties will be computed.

Example

```
Dim PerimeterResultsEnabled As Boolean
PerimeterResultsEnabled = Blob1.PerimeterResultsEnabled
Blob2.PerimeterResultsEnabled = True
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Blob extraction will be performed using color information if a color input image is selected. Blob extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for blob extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SamplingStep

Description

Step used in the tool's last execution to sample the box from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled box are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = Blob1.SamplingStep
```

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = Blob1.SamplingStepCustom
Blob2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image

Syntax

control.**SamplingStepCustomEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom .

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Enabled As Boolean
Enabled = Blob1.SamplingStepCustomEnabled
Blob2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

```
control.SamplingStepDefault
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = Blob.SamplingStepDefault
```

SegmentationDark

Description

Indexed property used to access the Dark Segmentation function. Two points are available, from left to right: Top and Bottom.

Syntax

```
control.SegmentationDark(Index As hsSegmentationDarkPoint) [= Value As Integer]
```

Type

Integer

Range

Minimum: 0
Maximum: 255

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Dark segmentation function point index	hsSegmentationDarkPoint	hsDarkTop, hsDarkBottom

Example

```
Dim Thresh As Integer
Thresh = Blob1.SegmentationDark(hsDarkTop)
Blob1.SegmentationDark(hsDarkBottom) = 128
```

SegmentationDynamicDark

Description

Indexed property used to access the Dynamic Dark Segmentation function. Two points are available, from left to right: Top and Bottom.

Syntax

control.**SegmentationDynamicDark**(*Index* As hsSegmentationDarkPoint) [= *Value* As Single]

Type

Single

Range

Minimum: 0.0 (%)

Maximum: 100.0 (%)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Dynamic Dark segmentation function point index	hsSegmentationDarkPoint	hsDarkTop, hsDarkBottom

Example

```
Dim Thresh As Single
Thresh = Blob1.SegmentationDynamicDark(hsDarkTop)
Blob1.SegmentationDynamicDark(hsDarkBottom) = 55.0
```

SegmentationDynamicInside

Description

Indexed property used to access the Dynamic Inside Segmentation function. Four points are available, from left to right: Bottom Left, Top Left, Top Right and Bottom Right.

Syntax

control.**SegmentationDynamicInside**(*Index* As hsSegmentationInsidePoint) [= *Value* As Single]

Type

Single

Range

Minimum: 0.0 (%)

Maximum: 100.0 (%)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Dynamic Inside segmentation function point index	hsSegmentationInsidePoint	hsInsideBottomLeft, hsInsideTopLeft, hsInsideTopRight, hsInsideBottomRight

Example

```
Dim Thresh As Single
Thresh = Blob1.SegmentationDynamicInside(hsInsideTopLeft)
```

```
Blob1.SegmentationDynamicInside(hsInsideBottomRight) = 65.0
```

SegmentationDynamicLight

Description

Indexed property used to access the Dynamic Light Segmentation function. Two points are available, from left to right: Bottom and Top.

Syntax

```
control.SegmentationDynamicLight(Index As hsSegmentationLightPoint) [= Value As Single]
```

Type

Single

Range

Minimum: 0.0 (%)

Maximum: 100.0 (%)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Dynamic Light segmentation function point index	hsSegmentationLightPoint	hsLightBottom, hsLightTop

Example

```
Dim Thresh As Single
```

```
Thresh = Blob1.SegmentationDynamicLight(hsLightTop)
```

```
Blob1.SegmentationDynamicLight(hsLightBottom) = 55.0
```

SegmentationDynamicOutside

Description

Indexed property used to access the Dynamic Outside Segmentation function. Four points are available, from left to right: Top Left, Bottom Left, Bottom Right and Top Right.

Syntax

```
control.SegmentationDynamicOutside(Index As hsSegmentationOutsidePoint) [= Value As Single]
```

Type

Single

Range

Minimum: 0.0 (%)

Maximum: 100.0 (%)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Dynamic Outside segmentation function point index	hsSegmentationOutsidePoint	hsOutsideTopLeft, hsOutsideBottomLeft, hsOutsideBottomRight, hsOutsideTopRight

Example

```
Dim Thresh As Single
```

```
Thresh = Blob1.SegmentationDynamicOutside(hsOutsideTopLeft)
Blob1.SegmentationDynamicOutside(hsOutsideBottomRight) = 65.0
```

SegmentationHSLInside

Description

Indexed property used to access the color HSL Inside Segmentation function. Six values are available: Hue, Saturation, Luminance and their respective tolerances.

Syntax

```
control.SegmentationHSLInside(Index As hsSegmentationHSLParameter)
    [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: 255 (128 for tolerance values)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	HSL Inside segmentation function parameter index	hsSegmentationHSLParameter	hsHSLHue, hsHSLSaturation, hsHSLLuminance, hsHSLHueTolerance, hsHSLSaturationTolerance, hsHSLLuminanceTolerance

Example

```
Dim Value As Integer
Value = Tool.SegmentationHSLInside(hsHSLHue)
Tool.SegmentationHSLInside(hsHSLSaturation) = 110
```

SegmentationHSLOutside

Description

Indexed property used to access the color HSL Outside Segmentation function. Six values are available: Hue, Saturation, Luminance and their respective tolerances.

Syntax

```
control.SegmentationHLSOutside(Index As hsSegmentationHSLParameter)
    [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: 255 (128 for tolerance values)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	HSL Outside segmentation function parameter index	hsSegmentationHSLParameter	hsHSLHue, hsHSLSaturation, hsHSLLuminance, hsHSLHueTolerance, hsHSLSaturationTolerance, hsHSLLuminanceTolerance

Example

```
Dim Value As Integer
Value = Tool.SegmentationHSLOutside(hsHSLHue)
Tool.SegmentationHSLOutside(hsHSLSaturation) = 110
```

SegmentationInside

Description

Indexed property used to access the Inside Segmentation function. Four points are available: from left to right, Bottom Left, Top Left, Top Right and Bottom Right.

Syntax

```
control.SegmentationInside(Index As hsSegmentationInsidePoint) [= Value As Integer]
```

Type

Integer

Range

Minimum: 0
Maximum: 255

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Inside segmentation function point index	hsSegmentationInsidePoint	hsInsideBottomLeft, hsInsideTopLeft, hsInsideTopRight, hsInsideBottomRight

Example

```
Dim Thresh As Integer
Thresh = Blob1.SegmentationInside(hsInsideTopLeft)
Blob1.SegmentationInside(hsInsideTopRight) = 110
```

SegmentationLight

Description

Indexed property used to access the Light Segmentation function. Two points are available, from left to right: Bottom and Top.

Syntax

```
control.SegmentationLight(Index As hsSegmentationLightPoint) [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Light segmentation function point index	hsSegmentationLightPoint	hsLightBottom, hsLightTop

Example

Dim Thresh As Integer

Thresh = Blob1.**SegmentationLight**(hsLightTop)Blob1.**SegmentationLight**(hsLightBottom) = 90

SegmentationMode

Description

Segmentation mode used by the tool to segment the input Image.

Syntax*control*.**SegmentationMode** [= *Value* As hsSegmentation]**Type**

hsSegmentation

Range

Name	Value
hsLight	0
hsDark	1
hsInside	2
hsOutside	3
hsDynamicLight	4
hsDynamicDark	5
hsDynamicInside	6
hsDynamicOutside	7
hsHSLInside	8
hsHSLOutside	9

Example

Dim SegMode As hsSegmentation

SegMode = Blob1.**SegmentationMode**Blob1.**SegmentationMode** = hsInside

SegmentationOutside

Description

Indexed property used to access the Outside Segmentation function. Four points are available, from left to right: Top Left, Bottom Left, Bottom Right and Top Right.

Syntax

control.**SegmentationOutside**(*Index* As hsSegmentationOutsidePoint) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Outside segmentation function point index	hsSegmentationOutsidePoint	hsOutsideTopLeft, hsOutsideBottomLeft, hsOutsideBottomRight, hsOutsideTopRight

Example

```
Dim Thresh As Integer
```

```
Thresh = Blob1.SegmentationOutside(hsOutsideTopLeft)
```

```
Blob1.SegmentationOutside(hsOutsideTopRight) = 110
```

SortBlobsBy

Description

Sorting mode used by the tool to sort blobs found.

Syntax

control.**SortBlobsBy**[= *Value* As hsSortBlobsBy]

Type

hsSortBlobsBy

Range

Name	Value
hsArea	0
hsBoundingBoxBottom	1
hsBoundingBoxCenterX	2
hsBoundingBoxCenterY	3
hsBoundingBoxHeight	4
hsBoundingBoxLeft	5
hsBoundingBoxRight	6
hsBoundingBoxRotation	7
hsBoundingBoxTop	8
hsBoundingBoxWidth	9
hsChainCodeDeltaX	10
hsChainCodeDeltaY	11
hsChainCodeLength	12

Name	Value
hsChainCodeStartX	13
hsChainCodeStartY	14
hsConvexPerimeter	15
hsElongation	16
hsExtentBottom	17
hsExtentLeft	18
hsExtentRight	19
hsExtentTop	20
hsGreyLevelMaximum	21
hsGreyLevelMean	22
hsGreyLevelMinimum	23
hsGreyLevelRange	24
hsGreyLevelStdDev	25
hsHoleCount	26
hsInertiaMaximum	27
hsInertiaMinimum	28
hsInertiaXAxis	29
hsInertiaYAxis	30
hsIntrinsicBoundingBoxBottom	31
hsIntrinsicBoundingBoxCenterX	32
hsIntrinsicBoundingBoxCenterY	33
hsIntrinsicBoundingBoxHeight	34
hsIntrinsicBoundingBoxLeft	35
hsIntrinsicBoundingBoxRight	36
hsIntrinsicBoundingBoxRotation	37
hsIntrinsicBoundingBoxTop	38
hsIntrinsicBoundingBoxWidth	39
hsIntrinsicExtentBottom	40
hsIntrinsicExtentLeft	41
hsIntrinsicExtentRight	42
hsIntrinsicExtentTop	43
hsPositionX	44
hsPositionY	45
hsPrincipalAxesRotation	46
hsRawPerimeter	47
hsRoundness	48

Remark(s)

Default Value: 0:hsArea

Example

```
Dim SortBy As hsSortBlobsBy
SortBy = Blob1.SortBlobsBy
Blob1.SortBlobsBy = hsRawPerimeter
```

SortResultsEnabled

Description

Property that specifies if the blobs are sorted in descending order of the selected sorting mode.

Syntax

control.**SortResultsEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
False	The blobs are not sorted.
True	The blobs are sorted in descending order of the selected sorting mode.

Example

```
Dim SortEnabled As Boolean
SortEnabled = Blob1.SortResultsEnabled
Blob2.SortResultsEnabled = False
```

ToolHeight

Description

Height of the tool box.

Syntax

control.**ToolHeight** [= Value As Single]

Type

Single

Range

Minimum: Greater than 0.0
Maximum: boundaries of the image

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = Blob1.ToolHeight
Blob2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool box.

Syntax

control.**ToolPositionX** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = Blob1.ToolPositionX
Blob2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the position of the tool box.

Syntax

control.**ToolPositionY** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = Blob1.ToolPositionY
Blob2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool box.

Syntax

control.**ToolRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = Blob1.ToolRotation
Blob2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool box.

Syntax

control.**ToolWidth** [= Value As Single]

Type

Single

Range

Minimum: Greater than 0.0
Maximum: boundaries of the image

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = Blob1.ToolWidth
Blob2.ToolWidth = 23.4
```

TopologicalResultsEnabled

Description

Enables the computation of the **BlobHoleCount** property.

Syntax

control.**TopologicalResultsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The topological properties will be computed.
False	No topological properties will be computed.

Example

```
Dim TopologicalResultsEnabled As Boolean
TopologicalResultsEnabled = _
    Blob1.TopologicalResultsEnabled
Blob2.TopologicalResultsEnabled = True
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transform is used to fit the tool to an object in the input image. The different components of the transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg

UnitsOrientation property value	TransformRotation property range
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = Blob1.TransformRotation
Blob2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transform is used to fit the tool to an object in the input image. The different components of the transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = Blob1.TransformScale
Blob2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transform is used to fit the tool to an object in the input image. The different components of the transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = Blob1.TransformTranslationX
Blob2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transform is used to fit the tool to an object in the input image. The different components of the transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = Blob1.TransformTranslationY
Blob2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSBlobAnalyzer".

Example

```
Dim ControlType As String
ControlType = Blob1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Name	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Remark(s)

This property is only applicable if **ModelTransformEnabled** or **CalibratedUnitsEnabled** is set to True.

Example

```
Dim Unit As hsUnitsLength
Unit = Blob1.UnitsLength
Blob2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Name	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Blob1.UnitsOrientation
Blob2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```
Dim Version As String
Version = Blob1.Version
```

Methods

AboutBox

Description

Displays the tool About box.

Syntax

control.**AboutBox()**

Parameter(s)

None

Return

No return value.

Example

```
Blob1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Blob Analyzer process.

Syntax

```
control.Execute()
```

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

Example

```
Blob1.Execute
```

GetDebugParameter

Description

Reserved for internal use

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

```
control.GetProcessingFormatEnabled(ProcessingFormat As hsProcessingFormat
```

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
(hsProcessingFormatGreyscale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

```
control.HideProperties()
```

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Blob1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

<i>Option parameter: bitfield elements</i>		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked.
hsExecuteRemaining	64	The tool and all subsequent tools in the Process Manager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effect provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

ExampleCaliper1.**ShowProperties**(True, hsShowOK + hsShowExecute)

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Builder

Description

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

Properties

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= *Value* As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input Image

Example

```
Dim Result As hsCoordinateSystem
Result = Builder1.CoordinateSystem
Builder2.CoordinateSystem = hsCoordinateSystemImage
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax*control*.**Description****Type**

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Builder1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax*control*.**ElapsedTime****Type**

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Builder1.ElapsedTime
```

ErrorObject

DescriptionReturns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.**Syntax***control*.**ErrorObject****Type**

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Builder.ErrorObject.Description
```

InputEntity

Description

The name of the input entity. The number and type of input entities depends on the current **Mode**.

Syntax

control.**InputEntity** (*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: InputEntityCount - 1

Example

```
Builder1.InputEntity(0) = "Point1"
```

InputEntityCount

Description

Index number of the input entity in the current **Mode**. Read only.

Syntax

control.**Index**

Type

Integer

Range

Greater than or equal to 0.

Example

```
Dim EntityCount As Integer
EntityCount = Builder1.InputEntityCount
```

InputEntityDatabase

Description

Name of the Database handle of the input entity.

Syntax

control.**InputEntityDatabase** (*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: InputEntityCount - 1

Example

```
Builder1.InputEntityDatabase(0) = HSAApplication.Database.Handle
```

InputEntityType

Description

The type of entity required as input. Read only.

Syntax

```
control.InputEntityType (Index As Integer)
```

Type

String

Range

The valid range for this property is as follows:

hsTypeImage	1	Image
hsTypeScene	2	Scene
hsTypePoint	4	Point
hsTypeLine	8	Line
hsTypeArc	16	Arc

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: InputEntityCount - 1

Example

```
Dim InputType As hsBuilderType
InputType = Builder1.InputEntityType
Builder1.InputEntityType = hsTypeArc
```

InputEntityView

Description

Name of the view that provides the input entity.

Syntax

```
control.InputEntityView (Index As Integer)[= Value As String]
```

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: InputEntityCount - 1

Example

```
Builder1.InputEntityView(0) = "View"
```

InputImage

Description

Name of the image on which the builder is applied. This image provides information for the builder output such as the image **Calibration**.

Syntax

```
control.InputImage [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim MyBuilder As HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320      'X
MyBuilder.Parameter(1) = 240      'Y
MyBuilder.Parameter(2) = False    'Calibrated units
MyBuilder.Execute
```

InputGreyScaleImage

Description

Name of the image on which the builder is applied. This image provides information for the builder output such as the image **Calibration**.

Syntax

```
control.InputImage [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim MyBuilder As HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320      'X
```

```

MyBuilder.Parameter(1) = 240      'Y
MyBuilder.Parameter(2) = False   'Calibrated units
MyBuilder.Execute

```

InputView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```

Dim MyBuilder AS HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320      'X
MyBuilder.Parameter(1) = 240      'Y
MyBuilder.Parameter(2) = False   'Calibrated units
MyBuilder.Execute

```

InputGreyScaleView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```

Dim MyBuilder AS HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320      'X
MyBuilder.Parameter(1) = 240      'Y
MyBuilder.Parameter(2) = False   'Calibrated units
MyBuilder.Execute

```

Intersected

Description

Intersected is true if there is any intersection of entities intersect. This property is valid only if the **Builder** is of type **Closest**, for example **hsModeClosestLineArc**. Read only.

Syntax

control.**Intersected**

Type

Boolean

Range

Value	Description
True	There is at least one intersection between the input entities.
False	There is no intersection.

Example

```
Dim Intersected As Boolean
Intersected = Builder1.Intersected
Builder1.Intersected = False
```

Mode

Description

Determines the current mode of the Builder tool. The number and type of entities output by the builder depend on the **Mode** value.

Syntax

control.**Mode**(= *Value* As hsBuilderMode)

Type

hsBuilderMode

Range

The valid range for this property is as follows:

Constant	Value	Description	Type
hsModeFixedPoint	0x1001	Builds a point from coordinates.	HSPoint
hsModeMidwayPoint	0x1002	Builds a point at midway between two points.	HSPoint
hsModeClosestPointLine	0x1010	Builds the closest point between a point and a line	HSPoint
hsModeClosestPointArc	0x1011	Builds the closest point between a point and an arc.	HSPoint
hsModeClosestLineLine	0x1012	Builds the closest points between two lines.	HSPoint
hsModeClosestLineArc	0x1013	Builds the closest points between a line and an arc.	HSPoint
hsModeClosestArcArc	0x1014	Builds the closest points between two arcs.	HSPoint
hsModeVectorPointofLine	0x1020	Builds a line that passes through a given point at a given angle.	HSLine
hsModeCenterPointofArc	0x1021	Builds a point at center point of an arc.	HSPoint
hsModeStartPointofLine	0x1022	Build a point at start point of a line.	HSPoint

Constant	Value	Description	Type
hsModeStartPointofArc	0x1023	Builds a point at start point of an arc.	HSPoint
hsModeEndPointofLine	0x1024	Builds a point at end point of a line.	HSPoint
hsModeEndPointofArc	0x1025	Builds a point at end point of an arc.	HSPoint
hsModeFixedLine	0x2001	Builds a line from parameters. (Center, angle and deltas)	HSLine
hsModeLine	0x2002	Builds a line from two points.	HSLine
hsModeLineParallel	0x2003	Builds a line parallel to a line.	HSLine
hsModeLinePerpendicular	0x2004	Builds a line perpendicular to a line.	HSLine
hsModeFixedLine2	0x2005	Builds a line from coordinates. (start and end points)	HSLine
hsModeFixedArc3	0x3001	Builds an arc from parameters. (Center, angles and radius)	HSArc
hsModeFixedArc	0x3002	Builds an arc from coordinates. (start, Mid and end points)	HSArc
hsModeFixedCircle	0x3003	Builds a circle from parameters. (Center and radius)	HSArc
hsModeFixedCircle2	0x3004	Builds a circle from coordinates. (Center and on circle points)	HSArc
hsModeArc	0x3005	Builds a an arc	HSArc
hsModeFixedCircle3	0x3006	Builds a circle from coordinates. (Three points on circle)	HSArc
hsModeCircle	0x3007	Builds a circle from three given points	HSArc

Parameter(s)

The number and type of the input entities, input parameters and output entities depend on the **Mode**, as listed in the table below.

Name	Description	Type	Range
<i>Index</i>	Index of the entity. The index of the entity depends on the hsBuilderMode value as shown in the table below.	Integer	Minimum: 0 Maximum: OutputEntityCount-1

hsBuilderMode	Input Entity	Input Parameter	Output Entity
hsModeArc	[0] Point (Start) [1] Point (End) [2] Point (Mid)	None	[0] Arc
hsModeCenterPointofArc	[0] Arc	None	[0] Point
hsModeCircle	[0] Point (On Circle) [1] Point (On Circle) [2] Point (On Circle)	None	[0] Arc
hsModeClosestArcArc	[0] Arc [1] Arc	[0] ExtrapolateArc0 [1] ExtrapolateArc1	[0] Point [1] Point

hsBuilderMode	Input Entity	Input Parameter	Output Entity
hsModeClosestLineArc	[0] Line [1] Arc	[0] ExtrapolateLine [1] ExtrapolateArc	[0] Point [1] Point
hsModeClosestLineLine	[0] Line [1] Line	[0] ExtrapolateLine0 [1] ExtrapolateLine1	[0] Point [1] Point
hsModeClosestPointArc	[0] Point [1] Arc	[0] ExtrapolateArc	[0] Point
hsModeClosestPointLine	[0] Point [1] Line	[0] ExtrapolateLine	[0] Point
hsModeEndPointofArc	[0] Arc	None	[0] Point
hsModeEndPointofLine	[0] Line	None	[0] Point
hsModeFixedArc	None	[0] Center X [1] Center Y [2] Radius [3] Start Angle [4] End Angle [5] Calibrated Units	[0] Arc
hsModeFixedArc3	None	[0] X1 (Start) [1] Y1 (Start) [2] X2 (End) [3] Y2 (End) [4] X3 (Mid) [5] Y3 (Mid) [6] Calibrated Units	[0] Arc
hsModeFixedCircle	None	[0] Center X [1] Center Y [2] Radius [3] Calibrated Units	[0] Arc
hsModeFixedCircle2	None	[0] X1 (Center) [1] Y1 (Center) [2] X2 (On Circle) [3] Y2 (On Circle) [4] Calibrated Units	[0] Arc
hsModeFixedCircle3	None	[0] X1 (On Circle) [1] Y1 (On Circle) [2] X2 (On Circle) [3] Y2 (On Circle) [4] X3 (On Circle) [5] Y3 (On Circle) [6] Calibrated Units	[0] Arc
hsModeFixedLine	None	[0] Vector X [1] Vector Y [2] Angle [3] Start Delta [4] End Delta [5] Calibrated Units	[0] Line

hsBuilderMode	Input Entity	Input Parameter	Output Entity
hsModeFixedLine2	None	[0] X1 (Start) [1] Y1 (Start) [2] X2 (End) [3] Y2 (End) [4] Calibrated Units	[0] Line
hsModeFixedPoint	None	[0] X [1] Y [2] Calibrated Units	[0] Point
hsModeLine	[0] Point [1] Point	None	[0] Line
hsModeLineParallel	[0] Point [1] Line	None	[0] Line
hsModeLinePerpendicular	[0] Point [1] Line	None	[0] Line
hsModeMidwayPoint	[0] Point [1] Point	None	[0] Point
hsModeStartPointofArc	[0] Arc	None	[0] Point
hsModeStartPointofLine	[0] Line	None	[0] Point
hsModeVectorPointofLine	[0] Line	None	[0] Point

Example

```

Dim MyBuilder As HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320    'X
MyBuilder.Parameter(1) = 240    'Y
MyBuilder.Parameter(2) = False  'Calibrated units
MyBuilder.Execute

```

Name**Description**

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= Value As String]

Type

String

Range

Not applicable.

Example

```

Dim ProcessName As String
ProcessName = Builder1.Name
Builder2.Name = "PointToArcBuilder"

```

OutputArc

Description

The output arc entity. Read only. Used to retrieve an HSArc object to the output result.

OutputArc is only valid if the output entity was generated by an **Execute**.

Syntax

control.**OutputArc** (*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Builder1.Mode = hsFixedArc
[...]
Builder1.Execute
Dim MyArc as HSArc
MyArc = Builder1.OutputArc(0)
```

OutputEntity

Description

Name of the output entity.

Syntax

control.**OutputEntity**(*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim BuiltEntity As String
BuiltEntity = Builder1.OutputEntity
Builder2.OutputEntity = "ClosestPoint"
```

OutputEntityCount

Description

Number of entities output by the tool in the current **Mode**. Read only.

Syntax

control.**OutputEntityCount**

Type

Integer

Range

Greater than or equal to 0.

Example

```
Dim EntityCount As Integer
EntityCount = Builder1.OutputEntityCount
```

OutputEntityDatabase

Description

The Database handle of the output entity.

Syntax

control.**OutputEntityDatabase** (*Index* As Integer) [= *Value* As OLE_HANDLE]

Type

OLE_HANDLE

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Builder1.OutputEntityDatabase = HSApplication.Database.Handle
```

OutputEntityExtrapolated

Description

Specifies the output entity was extrapolated. Read only. This property is valid only if the **Mode** is of type **Closest**, for example **hsModeClosestLineArc**.

Syntax

control.**OutputEntityExtrapolated** (*Index* As Integer)

Type

Boolean

Range

Value	Description
True	The entity will be extrapolated.
False	The entity will not be extrapolated.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Dim Enable As Boolean
```

```
Enable = Builder1.OutputEntityExtrapolated
Builder1.OutputEntityExtrapolated = False
```

OutputEntityType

Description

The type of the output entity. Read only.

Syntax

```
control.OutputEntityType(Index As Integer)
```

Type

String

Range

The valid range for this property is as follows:

Constant	Value	Description
hsTypeImage	1	Image
hsTypeScene	2	Scene
hsTypePoint	4	Point
hsTypeLine	8	Line
hsTypeArc	16	Arc

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Dim EntityView As String
EntityView = Builder1.OutputEntityType
Builder2.OutputEntityType = 1
```

OutputEntityView

Description

Name of the View to which the entity is output.

Syntax

```
control.OutputEntityView (Index As Integer)[= Value As String]
```

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Dim EntityView As String
EntityView = Builder1.InputEntityView
Builder2.InputEntityView = "View"
```

OutputLine

Description

The output line entity. Read only. This property is used to retrieve an HSLine object to the output result. **OutputLine** is only valid if the output entity was generated by an **Execute**.

Syntax

control.**OutputLine** (*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Builder1.Mode = hsFixedLine
[...]
Builder1.Execute
Dim MyLine as HSLine
MyLine = Builder1.OutputLine(0)
```

OutputPoint

Description

The output point entity. Read only. Used to retrieve an HSPoint object to the output result. **OutputPoint** is only valid if the output entity was generated by an **Execute**.

Syntax

control.**OutputPoint** (*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the entity.	Integer	Minimum: 0 Maximum: OutputEntityCount - 1

Example

```
Builder1.Mode = hsFixedPoint
```

```
[...]
Builder1.Execute
Dim MyPoint as HSPoint
MyPoint = Builder1.OutputPoint(0)
```

Parameter

Description

Value of a given indexed parameter for of the entity built by the tool. The number and type of parameters depends on the current **Mode**.

Syntax

control.**Parameter** (*Index* As Integer) [= *Value* As Variant]

Type

Single

Range

The valid range for this property is as follows:

hsBuilderMode	Value	Input Entity	Input Parameter Index and Name
hsModeArc	0x1001	[0] Point (Start) [1] Point (End) [2] Point (Mid)	None
hsModeCenterPointofArc	0x1002	[0] Arc	None
hsModeCircle	0x1010	[0] Point (On Circle) [1] Point (On Circle) [2] Point (On Circle)	None
hsModeClosestArcArc	0x1011	[0] Arc [1] Arc	[0] ExtrapolateArc0 [1] ExtrapolateArc1
hsModeClosestLineArc	0x1012	[0] Line [1] Arc	[0] ExtrapolateLine [1] ExtrapolateArc
hsModeClosestLineLine	0x1013	[0] Line [1] Line	[0] ExtrapolateLine0 [1] ExtrapolateLine1
hsModeClosestPointArc	0x1014	[0] Point [1] Arc	[0] ExtrapolateArc
hsModeClosestPointLine	0x1020	[0] Point [1] Line	[0] ExtrapolateLine
hsModeEndPointofArc	0x1021	[0] Arc	None
hsModeEndPointofLine	0x1022	[0] Line	None
hsModeFixedArc	0x1023	None	[0] X [1] Y [2] CalibratedUnits
hsModeFixedArc3	0x1024	None	[0] Center X [1] Center Y [2] Radius [3] Start Angle [4] End Angle [5] Calibrated Units

hsBuilderMode	Value	Input Entity	Input Parameter Index and Name
hsModeFixedCircle	0x1025	None	[0] X [1] Y [2] Radius [3] Calibrated Units
hsModeFixedCircle2	0x2001	None	[0] X1 (Center) [1] Y1 (Center) [2] X2 (Center) [3] Y2 (Center) [4] X3 (Center) [5] Y3 (Center) [6] Calibrated Units
hsModeFixedCircle3	0x2002	None	[0] Center X [1] Center Y [2] Radius [3] Calibrated Units
hsModeFixedLine	0x2003	None	[0] Vector X [1] Vector Y [2] Angle [3] Start Delta [4] End Delta [5] Calibrated Units
hsModeFixedLine2	0x2004	None	[0] X1 (Start) [1] Y1 (Start) [2] X2 (End) [3] Y2 (End) [4] Calibrated Units
hsModeFixedPoint	0x2005	None	[0] X [1] Y [2] Calibrated Units
hsModeLine	0x3001	[0] Point [1] Point	None
hsModeLineParallel	0x3002	[0] Point [1] Line	None
hsModeLinePerpendicular	0x3003	[0] Point [1] Line	None
hsModeMidwayPoint	0x3004	[0] Point [1] Point	None
hsModeStartPointofArc	0x3005	[0] Arc	None
hsModeStartPointofLine	0x3006	[0] Line	None
hsModeVectorPointofLine	0x3007	[0] Line	None

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	See Range table.	Minimum: 0 Maximum: ParameterCount - 1

Example

```

Dim MyBuilder AS HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320      'X
MyBuilder.Parameter(1) = 240      'Y
MyBuilder.Parameter(2) = False    'Calibrated units
MyBuilder.Execute

```

ParameterCount

Description

The number of input parameters in the current **Mode**. Read only.

Syntax

control.**ParameterCount**

Type

Integer

Range

Any positive integer.

Example

```

Dim ParametersCount As Integer
ParametersCount = Builder1.ParameterCount

```

ParameterName

Description

Name of the selected parameter. **ParameterName** changes when the **Mode** property is modified. Read only.

Syntax

control.**ParameterName**(*Index* As Integer)

Type

String

Range

Not applicable

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ParameterCount - 1

Example

```

Dim ParameterName As String
ParameterName = Builder1.ParameterName(0)

```

ParameterType

Description

The type of the selected parameter. **ParameterType** changes when the **Mode** property is modified. Read only.

Syntax

control.**ParameterType** (Index As Integer)

Type

hsParameterType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsBool	11	The parameter type is a Boolean.
hsFloat	4	The parameter type is a Float (Single) .

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ParameterCount - 1

Example

```
Dim ParamaterType As hsParameterType
ParameterType = Builder1.ParameterType(0)
```

ShowExtrapolations

Description

Specifies if the extrapolations are shown in the Results display.

Syntax

control.**ShowExtrapolations** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Entity extrapolations are displayed.
False	Entity extrapolations are not displayed.

Example

```
Builder1.ShowIExtrapolations = True
```

ShowImage

Description

Specifies if the image is shown in the Results display.

Syntax

control.**ShowImage** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example`Builder1.ShowImage = True`

ShowGreyScaleImage

Description

Specifies if the image is shown in the Results display.

Syntax`control.ShowImage [= Value As Boolean]`**Type**

Boolean

Range

Not applicable.

Example`Builder1.ShowImage = True`

ShowInputEntities

Description

Specifies if the input entities are shown in the Results display.

Syntax`control.ShowInputEntities [= Value As Boolean]`**Type**

Boolean

Range

Value	Description
True	Input entities are displayed.
False	Input entities are not displayed.

Example`Builder1.ShowInputEntities = True`

ShowOutputEntities

Description

Specifies if the output entities are shown in the Results display.

Syntax`control.ShowOutputEntities [= Value As Boolean]`**Type**

Boolean

Range

Value	Description
True	Output entities are displayed.
False	Output entities are not displayed.

Example

```
Builder1.ShowOutputEntities = True
```

TypeName**Description**

ActiveX type name for the tool. Read only.

Syntax

```
control.TypeName
```

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSBuilder".

Example

```
Dim ControlType As String
ControlType = Builder1.TypeName
```

UnitsLength**Description**

Active length unit.

Syntax

```
control.UnitsLength [= Value As hsUnitsLength]
```

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch

Constant	Value	Description
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = Builder1.UnitsLength
Builder2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Builder1.UnitsOrientation
Builder2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Builder1.Version
```

Methods

AboutBox

Description

Displays the Arc Finder tool **About** box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Return

No return value.

Example

Builder1.**AboutBox**

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Builder in the current **Mode**.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

```
Dim MyBuilder AS HSBuilder
MyBuilder.Mode = hsModeFixedPoint
MyBuilder.InputView = "Acquisition"
MyBuilder.InputImage = "Image"
MyBuilder.Parameter(0) = 320      'X
MyBuilder.Parameter(1) = 240      'Y
MyBuilder.Parameter(2) = False    'Calibrated units
MyBuilder.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the tool properties interface.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Boolean

Range

Value	Description
True	The interface was displayed when the method was called.
False	The interface was not displayed.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Builder1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the table below.

<i>Option</i> parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool process is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool process and the subsequent tools in the ProcessManager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = Builder1.ShowProperties _
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Caliper

Description

The Caliper measures distances between linear edges. Users can select edge pairs of interest by setting Caliper properties such as constraints, thresholds and polarities.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
```

```
Result = Caliper1.BilinearInterpolationEnabled
```

```
Caliper2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = Caliper1.CalibratedUnitsEnabled
Caliper2.CalibratedUnitsEnabled = False
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= *Value* As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Name	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim Result As hsCoordinateSystem
Result = Caliper1.CoordinateSystem
Caliper1.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
```

```
Description = Caliper1.Description
```

Edge1Constraints

Description

Bitwise field used to select the detection constraints for the first edge of the selected pair. Constraints can be set for position and/or magnitude and are used to score edges.

Syntax

control.**Edge1Constraints** (*Pair* as Variant) [= *Value* As hsConstraints]

Type

hsConstraints

Range

The valid range for this property is as follows:

Name	Value	Description
hsNone	0	No constraint
hsPosition	1	Position constraint
hsMagnitude	2	Magnitude constraint
hsAllConstraints	3	Magnitude and position constraints

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair. Other types not supported.

Example

```
Dim EdgeCt As hsConstraints
```

```
EdgeCt = Caliper1.Edge1Constraints("Pair0")
Caliper2.Edge1Constraints(0) = hsPosition
```

Edge1Magnitude

Description

Magnitude of the first edge of the selected pair. Read only.

Syntax

control.**Edge1Magnitude**(*Pair* as Variant)

Type

Single

Range

Minimum: -255

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim EdgeMag As Single
EdgeMag = Caliper1.Edge1Magnitude(0)
EdgeMag = Caliper2.Edge1Magnitude("Pair0")
```

Edge1MagnitudeConstraint

Description

Indexed property used to set the magnitude constraint function of the first edge of the selected pair. Two points are used: Base and Top.

Syntax

control.**Edge1MagnitudeConstraint** (*Pair* as Variant, *Index* As hsMagnitudeConstraintIndex) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the two points of the magnitude constraint function	hsMagnitudeConstraintIndex	hsBase: 1 hsTop: 2

Example

```
Dim MagCt As Integer
MagCt = Caliper1.Edge1MagnitudeConstraint(0,hsBase) = 9
Caliper2.Edge1MagnitudeConstraint("Pair0",2) = 60
```

Edge1MagnitudeScore

Description

Magnitude score of the first edge of the selected pair. Read only.

Syntax

control.**Edge1MagnitudeScore** (*Pair* as Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = Caliper1.Edge1MagnitudeScore ("Pair0")
Score = Caliper2.Edge1MagnitudeScore ( 0 )
```

Edge1Polarity

Description

Selection criterion of the first edge of the selected pair. The grey-scale transition of the edge must respect the polarity set by this property.

Syntax

control.**Edge1Polarity**(*Pair* as Variant) [= *Value* As hsPolarity]

Type

hsPolarity

Range

The valid range for this property is as follows:

Name	Value	Description
hsDarkToLight	0	The greylevel value must go from dark to light when crossing an edge.
hsLightToDark	1	The greylevel value must go from light to dark when crossing an edge.
hsEitherPolarity	2	The change in greylevel value change is not a criterion for locating an edge.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Pol As hsPolarity
Pol = Caliper1.Edge1Polarity ("Pair0")
Caliper2.Edge1Polarity ( 0 ) = hsEitherPolarity
```

Edge1PositionConstraint

Description

Indexed property used to set the position constraint function of the first edge of the selected pair. Four points are used: Base Left, Top Left, Top Right, Base Right.

Syntax

control.**Edge1PositionConstraint** (*Pair* as Variant, *Index* As hsPositionConstraintIndex) [= *Value* As Integer]

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the four points of the position constraint function.	hsPositionConstraintIndex	hsBaseLeft: 1 hsTopLeft: 2 hsTopRight: 3 hsBaseRight: 4

Remark(s)

A position constraint value of 0.0 represents the left of the tool while 1.0 represents the right of the tool.

Example

```
Dim PosCt As Single
PosCt = Caliper1.Edge1PositionConstraint(0,hsBaseLeft)
Caliper2.Edge1PositionConstraint("Pair0",hsTopLeft) = 0.8
```

Edge1PositionScore

Description

Position score of the first edge of the selected pair. Read only.

Syntax

```
control.Edge1PositionScore(Pair as Variant)
```

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = Caliper1.Edge1PositionScore(0)
Score = Caliper1.Edge1PositionScore("Pair0")
```


Edge1PositionX

Description

X coordinate of the center of the first edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge1PositionX**(*Pair* as Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = Caliper1.Edge1PositionX(0)
X_Pos = Caliper1.Edge1PositionX("Pair0")
```

Edge1PositionY

Description

Y coordinate of the center of the first edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge1PositionY**(*Pair* as Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim Y_Pos As Single
Y_Pos = Caliper1.Edge1PositionY(0)
Y_Pos = Caliper1.Edge1PositionY("Pair0")
```

Edge1Rotation

Description

Rotation of the first edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge1Rotation**(*Pair* as Variant)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	Edge1Rotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation As Single
Rotation = Caliper1.Edge1Rotation(0)
Rotation = Caliper1.Edge1Rotation("Pair0")
```

Edge1Score

Description

Score of the first edge of the selected pair. The score is computed according to the constraints set by the **Edge1Constraints** property. Read only.

Syntax

control.**Edge1Score**(Pair as Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = Caliper1.Edge1Score(0)
Score = Caliper1.Edge1Score("Pair0")
```

Edge1ScoreThreshold

Description

Minimum score to accept an edge as the first edge of the selected pair. The score of the first edge is returned by the **Edge1Score** property.

Syntax

control.**Edge1ScoreThreshold**(Pair as Variant) [= Value As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim ScoreTh As Single
```

```
ScoreTh = Caliper1.Edge1ScoreThreshold(1)
Caliper.Edge1ScoreThreshold("Pair0") = 0.75
```

Edge2Constraints

Description

Bitwise field used to select the detection constraints for the second edge of the selected pair. Constraints can be set for position and/or magnitude and are used to score edges.

Syntax

```
control.Edge2Constraints ( Pair as Variant) [= Value As hsConstraints]
```

Type

hsConstraints

Range

The valid range for this property is as follows:

Name	Value	Description
hsNone	0	No constraint
hsPosition	1	Position constraint
hsMagnitude	2	Magnitude constraint
hsAllConstraints	3	Magnitude and position constraints

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim EdgeCt As hsConstraints
EdgeCt = Caliper1.Edge2Constraints("Pair0")
Caliper2.Edge2Constraints(0) = hsPosition
```

Edge2Magnitude

Description

Magnitude of the second edge of the selected pair. Read only.

Syntax

```
control.Edge2Magnitude(Pair as Variant)
```

Type

Single

Range

Minimum: -255
Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim EdgeMag As Single
EdgeMag = Caliper1.Edge2Magnitude(0)
EdgeMag = Caliper2.Edge2Magnitude("Pair0")
```

Edge2MagnitudeConstraint

Description

Indexed property used to set the magnitude constraint function of the second edge of the selected pair. Two points are used: Base and Top.

Syntax

control.**Edge2MagnitudeConstraint** (*Pair* as Variant, *Index* As hsMagnitudeConstraintIndex) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0
Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the two points of the magnitude constraint function.	hsMagnitudeConstraintIndex	hsBase: 1 hsTop: 2

Example

```
Dim MagCt As Integer
MagCt = Caliper1.Edge2MagnitudeConstraint(0,hsBase) = 9
Caliper2.Edge2MagnitudeConstraint("Pair0",2) = 60
```

Edge2MagnitudeScore

Description

Magnitude score of the second edge of the selected pair. Read only.

Syntax

control.**Edge2MagnitudeScore** (*Pair* as Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

Dim Score As Single

Score = Caliper1.**Edge2MagnitudeScore**("Pair0")

Score = Caliper2.**Edge2MagnitudeScore**(0)

Edge2Polarity

Description

Validation criterion of the second edge of the selected pair. The grey-scale transition for the second edge must respect the polarity set by this property.

Syntax

control.**Edge2Polarity**(*Pair* as Variant) [= *Value* As hsPolarity]

Type

hsPolarity

Range

The valid range for this property is as follows:

Name	Value	Description
hsDarkToLight	0	The greylevel value must go from dark to light when crossing an edge.
hsLightToDark	1	The greylevel value must go from light to dark when crossing an edge.
hsEitherPolarity	2	The change in greylevel value change is not a criterion for locating an edge.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Pol As hsPolarity
Pol = Caliper1.Edge2Polarity("Pair0")
Caliper2.Edge2Polarity(0) = hsAllSlopes
```

Edge2PositionConstraint**Description**

Indexed property used to set the position constraint function of the second edge of the selected pair. Four points are used: Base Left, Top Left, Top Right, Base Right.

Syntax

control.**Edge2PositionConstraint** (*Pair* as Variant, *Index* As hsPositionConstraintIndex) [= *Value* As Integer]

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.
<i>Index</i>	One of the four points of the position constraint function.	hsPositionConstraintIndex	hsBaseLeft: 1 hsTopLeft: 2 hsTopright: 3 hsBaseRight: 4

Remark(s)

A position constraint value of 0.0 represents the left of the tool while 1.0 represents the right of the tool.

Example

```
Dim PosCt As Single
PosCt = Caliper1.Edge2PositionConstraint(0,hsBaseLeft)
Caliper2.Edge2PositionConstraint("Pair0",hsTopLeft) = 0.8
```

Edge2PositionScore

Description

Position score of the second edge of the selected pair. Read only.

Syntax

control.**Edge2PositionScore**(*Pair* as Variant)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
```

```
Score = Caliper1.Edge2PositionScore(0)
```

```
Score = Caliper1.Edge2PositionScore("Pair0")
```

Edge2PositionX

Description

X coordinate of the center of the second edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge2PositionX**(*Pair* as Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = Caliper1.Edge2PositionX(0)
X_Pos = Caliper1.Edge2PositionX("Pair0")
```

Edge2PositionY

Description

Y coordinate of the center of the second edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge2PositionY**(*Pair* as Variant)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim Y_Pos As Single
Y_Pos = Caliper1.Edge2PositionY(0)
Y_Pos = Caliper1.Edge2PositionY("Pair0")
```

Edge2Rotation

Description

Rotation of the second edge of the selected pair in the currently selected coordinate system. Read only.

Syntax

control.**Edge2Rotation**(*Pair* as Variant)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	Edge1Rotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation As Single
Rotation = Caliper1.Edge2Rotation(0)
Rotation = Caliper1.Edge2Rotation("Pair0")
```

Edge2Score

Description

Score of the second edge of the selected pair. The score is computed according to the constraints set by the **Edge2Constraints** property. Read only.

Syntax

control.**Edge2Score**(*Pair* as Variant)

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim Score As Single
Score = Caliper1.Edge2Score(0)
```

```
Score = Caliper1.Edge2Score("Pair0")
```

Edge2ScoreThreshold

Description

Minimum score to accept an edge as the second edge of the selected pair. The score of the second edge is returned by the **Edge2Score** property.

Syntax

```
control.Edge2ScoreThreshold(Pair as Variant) [= Value As Single]
```

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim ScoreTh As Single
ScoreTh = Caliper1.Edge2ScoreThreshold(1)
Caliper.Edge2ScoreThreshold("Pair0") = 0.75
```

EdgeFilterHalfWidth

Description

Half-width of the convolution filter used to compute the edge magnitude curve from which actual edges are detected. The filter approximates the first derivative of the projection curve. The half-width of the filter should be set in order to match the width of the edge in the projection curve (the extent of the grey-scale transition, expressed in number of pixels).

Syntax

```
control.EdgeFilterHalfWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: 1

Maximum: 25

Example

```
Dim HalfWidth As Integer
HalfWidth = Caliper1.EdgeFilterHalfWidth
Caliper2.EdgeFilterHalfWidth = 4
```

EdgeMagnitudeThreshold

Description

Magnitude threshold is used to find edges on the magnitude curve. A subpixel peak detection algorithm is applied on the region of every minimum or maximum of the curve that exceeds this threshold in order to locate edges.

Syntax

control.**EdgeMagnitudeThreshold** [= *Value* As Single]

Type

Single

Range

Minimum: 0

Maximum: 255

Example

```
Dim MagTh As Single
MagTh = Caliper1.EdgeMagnitudeThreshold
Caliper2.EdgeMagnitudeThreshold = 50.0
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Caliper1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

InputImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

```
control.InputImage [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Caliper1.InputImage
Caliper2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

```
control.InputImage [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Caliper1.InputImage
Caliper2.InputImage = "image1"
```

InputView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

```
control.InputView [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
```

```
ViewName = Caliper1.InputView
Caliper2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

```
control.InputView [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Caliper1.InputView
Caliper2.InputView = "view0"
```

InputInstanceScene

Description

Name of the scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

```
control.InputInstanceScene [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = Caliper1.InputInstanceScene
Caliper2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

```
control.InputInstanceView [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Caliper1.InputInstanceView
Caliper2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = Caliper1.InputModelScene
Caliper2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details.

Syntax

control.**InputModelView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = Caliper1.InputModelView
Caliper2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

Specifies if the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property.

Syntax

control.**InstanceSelectionEnabled** [= Value As Boolean]

Type

Boolean

Range

The valid range for this property is as follows:

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Remark(s)This property is only used if the **ModelTransformEnabled** property is set to True.**Example**

```
Dim Enabled As Boolean
Enabled = Caliper1.InstanceSelectionEnabled
Caliper2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **InstanceSelectionEnabled** property is set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = Caliper1.InstanceSelectionIndex
Caliper2.InstanceSelectionIndex = 0
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

```
control.ModelTransformEnabled [= Value As Boolean]
```

Type

Boolean

Range

The valid range for this property is as follows:

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = Caliper1.ModelTransformEnabled
Caliper2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = Caliper1.Name
Caliper2.Name = "CaliperB"
```

PairCount

Description

Number of pairs defined in the tool. Read only.

Syntax

control.**PairCount**

Type

Integer

Range

Minimum: 0

Maximum: unlimited

Example

```
Dim Count As Integer
Count = Caliper2.PairCount
```

PairIndex

Description

Index of the selected pair. Read only. This property is used to retrieve the index of a pair from its name.

Syntax

control.**PairIndex**(*Name* As String) As Integer

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of a pair	String	Not applicable.

Return

Integer

Range

Minimum: 0

Maximum: **PairCount** - 1

Example

```
Dim PairNdx As Integer
PairNdx = Caliper1.PairIndex("pad_width")
```

PairName

Description

Name of the selected pair.

Syntax

control.**PairName**(*Pair* as Variant) [=*NewName* As String]

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim PName As String
PName = Caliper1.PairName(0)
Caliper2.PairName("Pair0") = "FirstPair"
Caliper3.PairName(1) = PName
```

PairPositionX

Description

X coordinate of the selected pair in the currently selected coordinate system. The position of a pair is defined as the middle of the line segment drawn from the X-Y coordinates of the first and the second edge of that pair. Read only.

Syntax

control.**PairPositionX**(*Pair* as Variant)

Type

Single

Range

Bounded by the image. The minimum and maximum values depend on the **UnitsLength** property which sets the length units and the **CoordinateSystem** property which defines the coordinate system used by the tool to return its results.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = Caliper1.PairPositionX("Pair0")
X_Pos = Caliper2.PairPositionX(0)
```

PairPositionY

Description

Y coordinate of the selected pair in the currently selected coordinate system. The position of a pair is defined as the middle of the line segment drawn from the X-Y coordinates of the first and the second edge of that pair. Read only.

Syntax

control.**PairPositionY**(*Pair* as Variant)

Type

Single

Range

Bounded by the image. The minimum and maximum values depend on the **UnitsLength** property which sets the length units and the **CoordinateSystem** property which defines the coordinate system used by the tool to return its results.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim X_Pos As Single
X_Pos = Caliper1.PairPositionY("Pair0")
X_Pos = Caliper2.PairPositionY(0)
```

PairRotation**Description**

Angle of rotation of the selected pair in the currently selected coordinate system. The rotation of a given pair is always the same as the rotation of its first and second edges. Read only.

Syntax

Control.**PairRotation** (*Pair* as Variant)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	Edge1Rotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Angle As Single
Angle = Caliper1.PairRotation ( 0 )
Angle = Caliper1.PairRotation ("Pair0")
```

PairScore

Description

Score of the selected pair. The score of the pair is equal to the mean score of the two edges (**Edge1Score** and **Edge2Score**) that comprise the pair. Read only.

Syntax

```
control.PairScore(Pair as Variant)
```

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Example

```
Dim PScore As Single
PScore = Caliper1.PairScore(0)
PScore = Caliper1.PairScore("Pair0")
```

PairSize

Description

Distance between the two edges of a selected pair in the currently selected coordinate system. Read only.

Syntax

```
control.PairSize(Pair as Variant)
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair.

Remark(s)

The property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** as specified in the following table:

Mode	Length unit
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property
ModelTransformEnabled = True	UnitsLength property

Example

```
Dim Distance As Single
Distance = Caliper1.PairSize(0)
Distance = Caliper1.PairSize("Pair0")
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be performed using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SamplingStep

Description

Step used in the tool's last execution to sample the rectangle from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = Caliper1.SamplingStep
```

SamplingStepCustom

Description

Sampling-step factor defined by the user to override the default sampling step that can be computed by the tool. This value is used only if the **SamplingStepCustomEnabled** property is set to True.

Syntax

control.**SamplingStepCustom** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = Caliper1.SamplingStepCustom
Caliper2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

Value that specifies if the default sampling step is overridden by the **SamplingStepCustom** property.

Syntax

control.SamplingStepCustomEnabled [= *Value* As Boolean]

Type

Boolean

Range

The valid range for this property is as follows:

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Enabled As Boolean
Enabled = Caliper1.SamplingStepCustomEnabled
Caliper2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default sampling step for the current input image. Read only.

Syntax

control.SamplingStepDefault

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = Caliper.SamplingStepDefault
```

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.ToolHeight [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = Caliper1.ToolHeight
Caliper2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

```
control.ToolPositionX [= Value As Single]
```

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = Caliper1.ToolPositionX
Caliper2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the position of the tool rectangle.

Syntax

```
control.ToolPositionY [= Value As Single]
```

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = Caliper1.ToolPositionY
Caliper2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = Caliper1.ToolRotation
Caliper2.ToolRotation = -10.0
```

ToolSkew

Description

Skew angle of the tool rectangle.

Syntax

control.**ToolSkew** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolSkew property range
hsDegree	Minimum: -90 deg Maximum: +90 deg
hsRadian	Minimum: -Pi/2 rad Maximum: +Pi/2 rad

Example

```
Dim Skew As Single
Skew = Caliper1.ToolSkew
Caliper1.ToolSkew = 33.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = Caliper1.ToolWidth
Caliper2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale,

Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = Caliper1.TransformRotation
Caliper2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = Caliper1.TransformScale
Caliper2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = Caliper1.TransformTranslationX
Caliper2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = Caliper1.TransformTranslationY
Caliper2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSCaliper".

Example

```
Dim ControlType As String
ControlType = Caliper1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Name	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Remark(s)

This property is only applicable if **ModelTransformEnabled** or **CalibratedUnitsEnabled** is set to True.

Example

```
Dim Unit As hsUnitsLength
Unit = Caliper1.UnitsLength
Caliper2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Name	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Caliper1.UnitsOrientation
Caliper2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Caliper1.Version
```

Methods

AboutBox

Description

Displays the Caliper tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example`Caliper1.AboutBox`

AddPair

Description

Creates a new pair.

Syntax*control*.**AddPair**(Name As String) As Integer**Return**

Integer: index of the newly created pair.

Range

Minimum: 0

Maximum: unlimited

Parameter(s)

Name	Description	Type	Range
Name	Name of the Caliper pair to be added	String	Not applicable.

Example

```
Dim NewIndex As Integer
NewIndex = Caliper1.AddPair("ObjectWidth")
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Caliper process.

Syntax*control*.**Execute**()**Parameter(s)**

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

Example`Caliper1.Execute`

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

control.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
(hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise returns False.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Caliper1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

RemovePair

Description

Removes a pair from the tool.

Syntax

control.**RemovePair**(*Pair* as Variant)

Return

No return value.

Parameter(s)

Name	Description	Type	Range
<i>Pair</i>	Caliper pair selected.	Variant	Integer: Minimum: 0 Maximum: PairCount - 1 String: Range not applicable, parameter corresponds to the name of the pair. Other types not supported.

Example

```
Caliper1.RemovePair("Pair0")
```

```
Caliper1.RemovePair(0)
```

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True

Name	Description	Type	Range
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and what is the behavior of the control when the OK button is clicked. When this parameter is not provided all the elements of the bitfield are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the table below.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the bitfields hsExecuteOnOK and hsExecuteRemaining are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise returns False.

Example

Dim Success As Boolean

Success = Caliper1.**ShowProperties**(True, hsShowOK + hsShowExecute)

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Class Library

Description

The Class Library contains six ActiveX controls that provides an external access to an application's runtime database (HSDatabase, HSImage, HSScene, HSView), to error objects (HSErrorObject) and to calibrations (HSCalibration).

Class Library ActiveX Controls

HSArc

HSCalibration

HSDatabase

HSErrorObject

HSImage

HSLine

HSPoint

HSScene

HSView

HSArc

Description

The HSArc control allows the user to access and manipulate Arc entities. Entities can be generated by a HexSight Finder tool, the Builder tool or created programmatically within the Class Library.

An entity is defined by the properties illustrated Figure 1. The start and end of the arc are considered with respect to an X-to-Y trajectory along the arc.

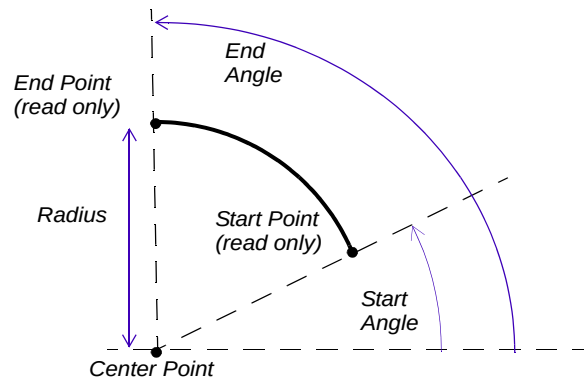


FIGURE 1 Arc Properties

Properties

Calibrated

Description

Specifies if the entity is defined in a calibrated or non-calibrated coordinate system. When **Calibrated** is false, the entity is defined in the Image Coordinate system and length units are in pixels. When **Calibrated** is True, the entity is defined in the World Coordinate system.

Syntax

control.Caibrated [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Line properties are expressed in calibrated units.
False	Line properties are expressed in pixels.

Remark(s)

Angles defined in a calibrated coordinate system cannot be recalculated into a non-calibrated coordinate system. The inverse is also true.

Example

```
'Build an arc within Class library
Dim MyArc As HSArc
Set MyArc = MyDatabase.AddArc ("View", "Arc")
With MyArc
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .CenterPoint.WorldX = 10
    .CenterPoint.WorldY = 10
    .Radius = 20
    .StartAngle = 45
    .EndAngle = 90
End With
```

Calibration

Description

Returns the internal **HSCalibration** object of the control. It is used to access or modify the calibration information.

Syntax

control.**Calibration** [= *Value* As HSCalibration]

Type

HSCalibration

Range

Not applicable.

Example

```
' Copy the image calibration to the entity
Dim MyArc As HSArc
MyArc.Calibration = MyImage.Calibration
```

CenterPoint

Description

The Point entity that is the center of the Arc entity.

Syntax

control.**CenterPoint** [= *Value* as HSPoint]

Type

HSPoint

Range

Any valid HSPoint object.

Example

```
' Build an arc within Class library
```

```
Dim MyArc As HSArc
Set MyArc = MyDatabase.AddArc ("View", "Arc")
With MyArc
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .CenterPoint.WorldX = 10
    .CenterPoint.WorldY = 10
    .Radius = 20
    .StartAngle = 45
    .EndAngle = 90
End With
```

Circle

Description

When True, indicates that the Arc entity is a circle.

Syntax

control.**Circle** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The arc entity is a circle.
False	The arc entity is a not a circle.

Remark(s)

When the value of Circle is modified, **StartAngle**, **EndAngle**, **StartPoint** and **EndPoint** properties are automatically updated.

Example

```
'Build an arc within Class library
Dim MyCircle As HSArc
Set MyCircle = MyDatabase.AddArc ("View", "Arc")
With MyCircle
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .Circle = True
    .CenterPoint.WorldX = 10
    .CenterPoint.WorldY = 10
    .Radius = 20
End With
```


Description

Description

String that describes the standard usage of the HSArc control. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = HSArc1.Description
```

EndAngle

Description

EndAngle is the angle at the end of an Arc entity, encountered in an X to Y trajectory along the arc. The value of the **Calibrated** property determines whether the **EndAngle**, **StartAngle** and **Radius** properties are defined in the Image coordinate system (non-calibrated) or in the World coordinate system (calibrated).

Syntax

control.**EndAngle** [= Value As Single]

Type

Single

Range

Any Single value.

Remark(s)

Angles defined in a calibrated coordinate system cannot be recalculated into a non-calibrated coordinate system. The inverse is also true.

Example

```
' Build an arc within Class library
Dim MyArc As HSArc
Set MyArc = MyDatabase.AddArc ("View", "Arc")
With MyArc
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .CenterPoint.WorldX = 10
    .CenterPoint.WorldY = 10
    .Radius = 20
    .StartAngle = 45
    .EndAngle = 90
End With
```

EndPoint

Description

EndPoint is the last point encountered in an X to Y trajectory along an Arc entity. Read only.
Any value set for this property will be ignored.

Syntax

control.**EndPoint**

Type

HSPoint

Range

Any valid HSPoint entity.

Example

```
Dim MyArc as HSArc
MyArc.EndPoint.ImageX = 320
MyArc.EndPoint.ImageY = 240
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

Information

Description

String that provides information pertaining to the object. For example: Name, Boundary coordinates, Image Coordinate System, Timestamp, etc.

Syntax

control.**Information** [= Value As String]

Type

String

Range

Not applicable

Example

```
Dim Info As String
```

Info = HSArc1.**Information**

ModifyCount

Description

Number of modifications that have been made to the Image. Read only.

Syntax

control.**ModifyCount**

Type

Long

Range

Greater than or equal to zero.

Example

```
Dim Count as Long
Count = HSArc1.ModifyCount
```

Name

Description

Name used to access the object.

Syntax

control.**Name**[= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ArcName As String
ArcName = HSArc1.Name
HSArc2.Name = "Arc_00"
```

Radius

Description

Radius of the Arc entity. The value of the **Calibrated** property determines whether the **EndAngle**, **StartAngle** and **Radius** properties are defined in the Image coordinate system (non-calibrated) or in the World coordinate system (calibrated).

Syntax

control.**Radius** [= *Value* As Single]

Type

Single

Range

Any Single value.

Remark(s)

A radius defined in a calibrated coordinate system cannot be recalculated into a non-calibrated coordinate system. The inverse is also true.

Example

```
'Build an arc within Class library
```

```

Dim MyArc As HSArc
Set MyArc = MyDatabase.AddArc ("View", "Arc")
With MyArc
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .CenterPoint.WorldX = 10
    .CenterPoint.WorldY = 10
    .Radius = 20
    .StartAngle = 45
    .EndAngle = 90
End With

```

StartAngle

Description

StartAngle is the angle at the start of an Arc entity, encountered in an X to Y trajectory along the arc. The value of the **Calibrated** property determines whether the **EndAngle**, **StartAngle** and **Radius** properties are defined in the Image coordinate system (non-calibrated) or in the World coordinate system (calibrated).

Syntax

control.**StartAngle**[= *Value* As Single]

Type

Single

Range

Any Single value.

Remark(s)

Angles defined in a calibrated coordinate system cannot be recalculated into a non-calibrated coordinate system. The inverse is also true.

Example

```

'Build an arc within Class library
Dim MyArc As HSArc
Set MyArc = MyDatabase.AddArc ("View", "Arc")
With MyArc
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .CenterPoint.WorldX = 10
    .CenterPoint.WorldY = 10
    .Radius = 20
    .StartAngle = 45
    .EndAngle = 90
End With

```

StartPoint

Description

StartPoint is the first point encountered in an X to Y trajectory along an Arc entity. Read only.
Any value set for this property will be ignored.

Syntax

control.**StartPoint**

Type

HSPoint

Range

Any valid HSPoint entity.

Example

```
Dim MyArc as HSArc
MyArc.StartPoint.ImageX = 400
MyArc.StartPoint.ImageY = 320
```

TypeName

Description

ActiveX type name for the Arc object. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSArc"

Example

```
Dim ObjectType As String
ObjectType = HSArc1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* as hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter

Constant	Value	Description
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit as hsUnitsLength
Unit = HSArc1.UnitsLength
HSArc2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* as hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit as hsUnitsOrientation
Unit = HSArc1.UnitsOrientation
HSArc2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```
Dim Version As String
Version = HSArc1.Version
```

Methods

AboutBox

Description

Displays the HSArc About box.

Syntax

```
control.AboutBox( )
```

Parameter(s)

None

Return

No return value.

Example

```
HSArc1.AboutBox
```

AngleWithinRange

Description

Evaluates if the specified angle is within the range that is defined by the **StartAngle** and **EndAngle** properties.

Syntax

```
control.AngleWithinRange (Angle As Single) As Boolean
```

Parameter(s)

Name	Description	Type	Range
<i>Angle</i>	The Angle value that will be evaluated.	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the angle is within range. Expressed in Units specified by UnitsOrientation property.	True, False

Example

```
Dim InRange As Boolean
InRange = Arc.AngleWithinRange(45.0)
```

CopyFormatFrom

Description

Copies the format of a specified object to the current object.

Syntax

control.**CopyFormatFrom**(*SourceObject* As Object)

Parameter(s)

Name	Description	Type	Range
<i>SourceObject</i>	Object from which the format is copied.	HSScene, HSImage, HSArc, HSLine, or HSPoint.	Not applicable.

Return

No return value.

Example

Arc1.**CopyFormatFrom**(Arc0)

GetDispatchPointer

Description

Reserved for internal use.

Load

Description

Loads the content of an arc file (*.hac)

Syntax

control.**Load**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the hac file to load. Optional.	Variant	Not applicable

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSArc.Load("Arc0.hac")
```

NearestPoint

Description

Computes the coordinates of the point on the current HSArc object that is nearest to the specified object called the *SourceEntity*. The value of the **Calibration** property determines whether coordinates are specified in the non-calibrated Image coordinate system or the calibrated World coordinate system.

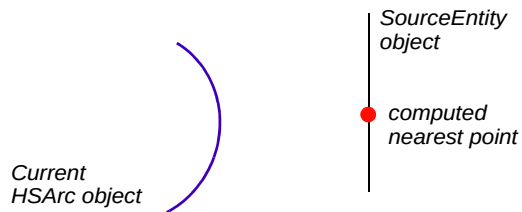


FIGURE 2 *HSPoint — NearestPoint Method*

Syntax

control.**NearestPoint**(*SourceEntity* As Object, *ExtrapolateThis* As Boolean, *ExtrapolateEntity* As Boolean, *Intersected* As Boolean) As Object

Parameter(s)

Name	Description	Type	Range
<i>SourceEntity</i>	Input Entity used to compute nearest point.	HSPoint, HSLine, HSArc	Not applicable.
<i>ExtrapolateThis</i>	Extrapolate current object.	Boolean	True, False
<i>ExtrapolateEntity</i>	Extrapolate the specified SourceEntity.	Boolean	True, False
<i>Intersected</i>	Determine if the calculated point coordinates comes from an real intersection (as opposed as an intersection from an extrapolated entity.)	Boolean	True, False

Return

HSPoint

Example

```
Dim Intersect As Boolean
Dim NearestPoint As Boolean
Set NearestPoint = Arc.NearestPoint(Line, False Intersect)
```

Reset

Description

Reserved for internal use.

Save

Description

Saves the Arc entity to a HexSight Arc file(*.hac).

Syntax

control.**Save**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the arc file (*.hac) to which the arc is saved. Optional.	Variant	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean  
Succeed = HSArc1.Save("Arc0.hac")  
Succeed = HSArc2.Save( )
```

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

HSCalibration

Description

An HSCalibration control contains all the information about a specific calibration within HexSight. The HSImage, HSScene and HSAcquisitionDevice controls contain such a calibration object which can be accessed through the Calibration property.

Properties

AveragePixelHeight

Description

Average calibrated height of a pixel, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**AveragePixelHeight**

Type

Single

Range

Greater than or equal to zero.

Example

```
Dim H As Single
H = Calibration.AveragePixelHeight
```

AveragePixelSize

Description

Average calibrated size of a pixel, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**AveragePixelSize**

Type

Single

Range

Greater than or equal to zero.

Example

```
Dim S As Single
S = Calibration.AveragePixelHeight
```

AveragePixelWidth

Description

Average calibrated width of a pixel, expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax*control.AveragePixelWidth***Type**

Single

Range

Greater than or equal to zero.

Example

```
Dim W As Single
W = Calibration.AveragePixelWidth
```

CalibrationPointer

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the control. Read only.

Syntax*control.Description***Type**

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Calibration.Description
```

ErrorObject

Description

Returns a reference to the internal HSErrorObject object of the control. It is used for handling errors in an application. Read only.

Syntax*control.ErrorObject***Type**

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
Handler:
MsgBox "Error: " & Calibration.ErrorObject.Description
```

ImageHeight

Description

Height of the calibration target image expressed in pixels. Read only.

Syntax

control.**ImageHeight**

Type

Long

Range

Any positive integer value.

Example

Dim Value as Long

Value = Calibration.**ImageHeight**

ImageWidth

Description

Height of the calibration target image expressed in pixels. Read only.

Syntax

control.**ImageWidth**

Type

Long

Range

Any positive integer value.

Example

Dim Value as Long

Value = Calibration.**ImageWidth**

Information

Description

String that provides information pertaining to the object. For example: Name.

Syntax

control.**Information** [= Value As String]

Type

String

Range

Not applicable.

Example

Dim CalibrationInfo as String

CalibrationInfo = Calibration.**Information**

Parameter

Description

Value of a given indexed parameter for the calibration. Read only if calibration **Type** is not hsCalibrationAuto.

Syntax

control.**CalibrationParameter**(*Index* As Integer) [= *Value* As Single]

Type

Single

Range

Depends on the selected parameter and the calibration type.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ParameterCount - 1

Remark(s)

When a new calibration is created programmatically using the `hsCalibrationAuto` **Type**, the indexed **Parameter** property is used to set its parameters. A description of each parameter index is provided in the table below.

Index	Description
0	Number of pairs of calibration points used for the calibration
1	World coordinate system type. A value of 0 defines a right-handed coordinate system and value of 1 defines a left-handed coordinate system
2	Reserved for future use
3	Reserved for future use
4	Reserved for future use
5	Reserved for future use
6	Width of the image in pixels
7	Height of the image in pixels
8	Read only: average width of an image pixel
9	Read only: average height of an image pixel
10	X coordinate of point 1 expressed with respect to the Image coordinate system
11	Y coordinate of point 1 expressed with respect to the Image coordinate system
12	X coordinate of point 1 expressed with respect to the World coordinate system
13	Y coordinate of point 1 expressed with respect to the World coordinate system
14	X coordinate of point 2 expressed with respect to the Image coordinate system
15	Y coordinate of point 2 expressed with respect to the Image coordinate system
16	X coordinate of point 2 expressed with respect to the World coordinate system
17	Y coordinate of point 2 expressed with respect to the World coordinate system
...	...
$10+4(n-1)$	X coordinate of point n expressed with respect to the Image coordinate system
$10+4(n-1)+1$	Y coordinate of point n expressed with respect to the Image coordinate system
$10+4(n-1)+2$	X coordinate of point n expressed with respect to the World coordinate system
$10+4(n-1)+3$	Y coordinate of point n expressed with respect to the World coordinate system

Example

```

' Create a new hsCalibrationAuto calibration
Calibration.Create
Calibration.Type = hsCalibrationAuto
' Set the number of pairs of calibration points to 4
Calibration.Parameter(0) = 4
' Calibrate into a left-handed coordinate system
Calibration.Parameter(1) = 1
' Calibration point 1
Calibration.Parameter(10) = 100.0
Calibration.Parameter(11) = 50.0
Calibration.Parameter(12) = -1500.0
Calibration.Parameter(13) = 2300.0
' Calibration point 2
Calibration.Parameter(14) = 500.0
Calibration.Parameter(15) = 50.0
Calibration.Parameter(16) = 1500.0
Calibration.Parameter(17) = 2300.0
' Calibration point 3
Calibration.Parameter(18) = 100.0
Calibration.Parameter(19) = 400.0
Calibration.Parameter(20) = -1500.0
Calibration.Parameter(21) = 6100.0
' Calibration Point 4
Calibration.Parameter(22) = 500.0
Calibration.Parameter(23) = 400.0
Calibration.Parameter(24) = 1500.0
Calibration.Parameter(25) = 6100.0
' Solve the calibration
Calibration.Solve
' Check if the calibration is valid
If Not Calibration.State = hsCalibrationReady Then
    GoTo ErrorHandler
End If

```

ParameterCount

Description

Number of parameters for the calibration. Read only.

Syntax

control.**ParameterCount**

Type

Integer

Range

Greater than 0.

Example

```
Dim Count As Integer
Count = Calibration.ParameterCount
```

ParameterDescription

Description

Text description of a given indexed parameter for the calibration. Read only.

Syntax

control.**ParameterDescription**(*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ParameterCount - 1

Example

```
Dim Description As String
Description = Calibration.ParameterDescription(2)
```

ParameterName

Description

Name of a given indexed parameter for the calibration. Read only.

Syntax

control.**ParameterName**(*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the parameter.	Integer	Minimum: 0 Maximum: ParameterCount - 1

Example

```
Dim Name As String
Name = Calibration.ParameterName(4)
```


State

Description

State of the calibration. Read only.

Syntax

control.**State**

Type

hsCalibrationState

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCalibrationReady	1	Calibration valid and ready to be used.
hsCalibrationSolveFailed	2	Calibration solving failed.
hsCalibrationParameterChanged	0x100	A calibration parameter has changed.
hsCalibrationPointChanged	0x200	A calibration point has changed.

Example

```
Dim State As hsCalibrationState
State = Calibration.State
```

Type

Description

Type of the calibration. Read only for a calibration object created internally by the HSAcquisitionDevice control. Can be set to **hsCalibrationAuto** to create a new calibration object programmatically. When a new **hsCalibrationAuto** calibration is created, the actual calibration method that is used depends on the number of pairs of World-Image coordinates that are provided upon its creation. With 2 to 3 pairs of World-Image coordinates, X and Y scale factors calibration is used. With 4 to 8 pairs of World-Image coordinates, perspective correction calibration is used. With 9 or more pairs of World-Image coordinates, distortion correction calibration is used. The parameters of a **hsCalibrationAuto** calibration, including the pairs of World-Image coordinates, are set through the indexed **Parameter** property.

Syntax

control.**Type** [= *Value* As hsCalibrationType]

Type

hsCalibrationType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCalibrationUndefined	0	Undefined.
hsCalibrationXYScale	1	X and Y scale calibration factors.
hsCalibrationPerspectiveOld	2	Old perspective correction. (obsolete)
hsCalibrationDistortionLUT	3	Lens and perspective distortion correction by lookup table.

Constant	Value	Description
hsCalibrationDistortionModel	4	Lens and perspective distortion correction using a non-linear model.
hsCalibrationPerspective	5	Perspective correction.
hsCalibrationAuto	6	Calibration method automatically selected according to the number of calibration points provided. Available only programmatically.

Example

```
Dim Type As hsCalibrationType
Type = Calibration.CalibrationType
```

TypeName

Description

ActiveX type name for the object. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSCalibration"

Example

```
Dim ObjectType As String
ObjectType = HSCalibration1.TypeName
```

UnitsLength

Description

Units used for the calibration. Read only.

Syntax

control.**UnitsLength**

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot

Constant	Value	Description
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = Calibration.UnitsLength
```

Version

Description

String containing the current HexSight version and build information. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Calibration.Version
```

WorldCoordinateSystemType

Description

Type of World coordinate system used to express coordinates in calibrated units. Read only. Can either be right-handed or left-handed. In a right-handed World coordinate system, the Y-axis is 90 degrees counterclockwise from the X-axis. In a left-handed World coordinate system, the Y-axis is 90 degrees clockwise from the X-axis.

Syntax

control.**WorldCoordinateSystemType**

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRightHanded	0	World coordinate is right-handed (Y-axis 90 degrees counter clockwise from the X-axis).
hsLeftHanded	1	World coordinate is left-handed (Y-axis 90 degrees clockwise from the X-axis).

Remark(s)

The World coordinate system type for a calibration is set at its creation. The **WorldCoordinateSystemType** property is read only and cannot be used to dynamically modify the World coordinate system type of a calibration. When a new calibration is created programmatically using the `hsCalibrationAuto` **Type**, its World coordinate system type is set through its indexed parameters using the **Parameter** property.

Example

```
Dim CurrentMode As hsCoordinateSystemType
CurrentMode = HSCalibration.WorldCoordinateSystemType
```

Methods

AboutBox

Description

Displays the Calibration object About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value

Example

Calibration.**AboutBox**

Create

Description

Creates a new calibration.

Syntax

control.**Create** As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

Calibration.**Create**

GetDispatchPointer

Description

Reserved for internal use.

ImageToWorld

Description

This method takes two parameters (X and Y) that represent a point in the Image coordinate system. When the method is executed the two entered parameters are modified to express the same point with respect to the World coordinate system.

Syntax

control.**ImageToWorld**(X As Single, Y As Single) As Boolean

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point.	Single	Any Single value.
Y	Y coordinate of the point.	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If the Image is not calibrated then the method will return False.

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 200
Y1 = 350
Succeed = Calibration.ImageToWorld(X1,Y1)
```

Load

Description

Loads the content of a calibration file (*.cal).

Syntax

control.**Load**([FileName As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file where the content of the Calibration will be saved.	Variant	If the parameter is a String then the concept of range does not apply. All others variable types are not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file. Calibration files (*.cal) are compatible with the Acquisition Device.

Example

```
Dim Succeed As Boolean
Succeed =
    Calibration1.Load("..\CalibrationFiles\CalFile2.hdb")
Succeed = Calibration2.Load( )
```

Save

Description

Saves the contents of the calibration (*.cal) to the specified file.

Syntax

control.**Save**([*FileName* As Variant]) *Value* As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file where the content of the Calibration will be saved.	Variant	If the parameter is a String then the concept of range does not apply. All others variable types are not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file. Calibration files (*.cal) are compatible with the Acquisition Device.

Example

```
Dim Succeed As Boolean
Succeed =
    Calibration1.Save("..\CalibrationFiles\Calib1.hdb")
Succeed = Calibration2.Save( )
```

Solve

Description

Solves the calibration.

Syntax

control.**Solve**

Parameter(s)

None

Return

No return value.

Example

Calibration.**Solve**

WorldToImage

Description

This method takes two parameters (X and Y) that represent a point in the World coordinate system. When the method is executed the two entered parameters are modified to express the same point with respect to the Image coordinate system.

Syntax

control.**WorldToImage**(X As Single, Y As Single) As Boolean

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point	Single	Any Single value.
Y	Y coordinate of the point	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If the Image is not calibrated then the method will return False.

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 120.3
Y1 = 35.0
Succeed = Calibration.WorldToImage(X1,Y1)
```

Events

Error

Description

Reserved for internal use.

HSDatabase

Description

An HSDatabase control is 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 HexSight processes.

Properties

Arc

Description

Arc object in the Database.

Syntax

control.Arc(ViewName As String, ArcName As String) [= Value As HSArc]

Type

HSArc

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>ArcName</i>	Name of the object.	String	Not applicable.

Example

```
Dim MyArc As HSArc
Set MyArc = HSDatabase1.Arc("Finder","Hole")
```

Description

Description

String that describes the standard use of the control. Read only.

Syntax

control.Description

Type

String

Range

Not applicable.

Example

```
Dim Descrip As String
Descrip = HSDatabase1.Description
```


ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

Filename

Description

Path and name of the file where the Database has been retrieved. Read only.

Syntax

control.**Filename**

Type

String

Range

Not applicable.

Example

```
Dim NameOfFile As String
NameOfFile = HSDatabase1.Filename
```

FirstImageName

Description

Name of the first Image in the selected View. Returns an empty string when the list of Images is empty. Returns "(undefined)" when the name of the first Image is not defined. Accessing this property resets the internal Image name iterator to the first Image in the list. This internal iterator can be used to browse the Image names with the following properties:

FirstImageName, **LastImageName**, **NextImageName** and **PreviousImageName**. Read only.

Syntax

control.**FirstImageName**(*ViewName* As String)

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.FirstImageName("Acquisition")
```

FirstSceneName

Description

Name of the first Scene in the selected View. Returns an empty string when the list of Scenes is empty. Returns "(undefined)" when the name of the first Scene is not defined. Accessing this property resets the internal Scene name iterator to the first Scene in the list. This internal iterator can be used to browse the Scene names with the following properties:

FirstSceneName, **LastSceneName**, **NextSceneName** and **PreviousSceneName**. Read only.

Syntax

control.**FirstSceneName**(*ViewName* As String)

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.FirstSceneName("Acquisition")
```

FirstViewName

Description

Name of the first View in the Database. Returns an empty string when the list of Views is empty. Returns "(undefined)" when the name of the first View is not defined. Accessing this property resets the internal View name iterator to the first View in the list. This internal iterator can be used to browse the View names with the following properties: **FirstViewName**, **LastViewName**, **NextViewName** and **PreviousViewName**. Read only.

Syntax

control.**FirstViewName**

Type

String

Range

Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.FirstViewName
```

Handle

Description

Provides a OLE_HANDLE to the Database. Read only.

Syntax

control.**Handle**

Type

OLE_HANDLE

Range

Not applicable.

Example

```
Dim handle As OLE_HANDLE
handle = HSDatabase1.Handle
```

Image

Description

Image object in the Database.

Syntax

control.**Image**(*ViewName* As String, *ImageName* As String)

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>ImageName</i>	Name of the object.	String	Not applicable.

Type

HSImage

Range

Not applicable.

Example

```
Dim MyImage As HSImage
Set MyImage = HSDatabase1.Image("Acquisition","Image")
```

Information

Description

String that provides information pertaining to the object. For example: Name, Timestamp, etc.

Syntax

control.**Information** [= *Value* As String]

Type

String

Range

Not applicable

Example

```
Dim Info As String
Info = Database1.Information
```

LastImageName

Description

Name of the last Image in a selected View. Returns an empty string when the list of Images is empty. Returns "(undefined)" when the name of the last Image is not defined. Accessing this property resets the internal Image name iterator to the last Image in the list. This internal iterator can be used to browse the Image names with the following properties:

FirstImageName, **LastImageName**, **NextImageName** and **PreviousImageName**. Read only.

Syntax

```
control.LastImageName(ViewName As String)
```

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.LastImageName("Acquisition")
```

LastSceneName

Description

Name of the last Scene in the selected View. Returns an empty string when the list of Scenes is empty. Returns "(undefined)" when the name of the last Scene is not defined. Accessing this property resets the internal Scene name iterator to the last Scene in the list. This internal iterator can be used to browse the Scene names with the following properties:

FirstSceneName, **LastImageName**, **NextSceneName** and **PreviousSceneName**. Read only.

Syntax

```
control.LastSceneName(ViewName As String)
```

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.LastSceneName("Acquisition")
```

LastViewName

Description

Name of the last View in the Database. Returns an empty string when the list of Views is empty. Returns "(undefined)" when the name of the last View is not defined. Accessing this property resets the internal View name iterator to the last View in the list. This internal iterator can be used to browse the View names with the following properties: **FirstViewName**, **LastViewName**, **NextViewName** and **PreviousViewName**. Read only.

Syntax

```
control.LastViewName
```

Type

String

Range

Not applicable.

Example

```
Dim name As String
name = HSDatabase1.LastViewName
```

Line

Description

Line object in the Database.

Syntax

```
control.Line(ViewName As String, LineName As String)
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>LineName</i>	Name of the object.	String	Not applicable.

Type

HSLine

Range

Not applicable.

Example

```
Dim MyLine As HSLine
Set MyLine = HSDatabase1.Line("Finder","LeftSide")
```

ModifyCount

Description

Number of modifications that have been made to the Database. Read only.

Syntax

```
control.ModifyCount
```

Type

Integer

Range

Greater than or equal to zero.

Example

```
Dim Count As Integer
Count = HSDatabase1.ModifyCount
```

Name

Description

Name used to access the object.

Syntax*control*.**Name** [= *Value* As String]**Type**

String

Range

Not applicable.

Example

```
Dim Name As String
Name = Database1.Name
Database.Name = "HookInspection"
```

NextImageName

Description

Moves the internal Image name iterator to the next Image in the selected View and returns its name. Returns an empty string after the last Image is reached. Returns "(undefined)" if the name of the Image is not defined. This internal iterator can be used to browse the Image names with the following properties: **FirstImageName**, **LastImageName**, **NextImageName** and **PreviousImageName**. Read only.

Syntax*control*.**NextImageName**(*ViewName* As String)**Type**

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.NextImageName("Acquisition")
```

NextSceneName

Description

Moves the internal Scene name iterator to the next Scene in the selected View and returns its name. Returns an empty string after the last Scene is reached. Returns "(undefined)" if the name of the Scene is not defined. This internal iterator can be used to browse the Scene names with the following properties: **FirstViewName**, **LastSceneName**, **NextSceneName** and **PreviousSceneName**. Read only.

Syntax

control.**NextSceneName**(*ViewName* As String)

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable

Example

```
Dim Name As String
Name = HSDatabase1.NextSceneName("Acquisition")
```

NextViewName

Description

Moves the internal View name iterator to the next View in the Database and returns its name. Returns an empty string after the last View is reached. Returns "(undefined)" if the name of the View is not defined. This internal iterator can be used to browse the View names with the following properties: **FirstViewName**, **LastViewName**, **NextViewName** and **PreviousViewName**. Read only.

Syntax

control.**NextViewName**

Type

String

Range

Not applicable.

Parameter(s)

Example

```
Dim Name As String
Name = HSDatabase1.NextViewName
```

Password

Description

Reserved for internal use.

Point

Description

Point object in the Database.

Syntax

control.**Point**(*ViewName* As String, *PointName* As String) [= *Value* As HSPoint]

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>PointName</i>	Name of the object.	String	Not applicable.

Type

HSPoint

Range

Not applicable.

Example

```
Dim MyPoint As HSPoint
Set MyPoint = HSDatabase1.Point("Finder", "Corner")
```

PreviousImageName

Description

Moves the internal Image name iterator to the previous Image in the selected View and returns its name. Returns an empty string after the first Image is reached. Returns "(undefined)" if the name of the Image is not defined. This internal iterator can be used to browse the Image names with the following properties: **FirstImageName**, **LastImageName**, **NextImageName** and **PreviousImageName**. Read only.

Syntax

control.**PreviousImageName**(*ViewName* As String)

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.PreviousImageName("Acquisition")
```

PreviousSceneName

Description

Moves the internal Scene name iterator to the previous Scene in the selected View and returns its name. Returns an empty string after the first Scene is reached. Returns "(undefined)" if the name of the Scene is not defined. This internal iterator can be used to browse the Scene

names with the following properties: **FirstSceneName**, **LastSceneName**, **NextSceneName** and **PreviousSceneName**. Read only.

Syntax

control.**PreviousSceneName**(*ViewName* As String)

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View.	String	Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.PreviousSceneName("Acquisition")
```

PreviousViewName

Description

Moves the internal View name iterator to the previous View in the Database and returns its name. Returns an empty string after the first View is reached. Returns "(undefined)" if the name of the View is not defined. This internal iterator can be used to browse the View names with the following properties: **FirstViewName**, **LastViewName**, **NextViewName** and **PreviousViewName**. Read only.

Syntax

control.**PreviousViewName**

Type

String

Range

Not applicable.

Example

```
Dim Name As String
Name = HSDatabase1.PreviousViewName
```

Scene

Description

Scene object in the Database.

Syntax

control.**Scene**(*ViewName* As String, *SceneName* As String) [= *Value* As HSScene]

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>SceneName</i>	Name of the object.	String	Not applicable.

Type

HSScene

Range

Not applicable.

Example

```
Dim MyScene As HSScene
Set MyScene = HSDatabase1.Scene("Locator", "ContourScene")
```

TypeName

Description

ActiveX type name for the object. Read only.

Syntax*control*.**TypeName****Type**

String

Range

Not applicable.

Remark(s)

This property is always "HSDatabase"

Example

```
Dim ObjectType As String
ObjectType = Database.TypeName
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax*control*.**Version****Type**

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```
Dim Version As String
Version = HSDatabase1.Version
```

View

Description

View object in the database.

Syntax*control*.**View**(*ViewName* As String) [= *Value* As HSView]

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.

Type

HSView

Range

Not applicable.

Example

```
Dim lView As HSView
Set lView = HSDatabase1.View("Acquisition")
```

ViewCount

Description

Number of View objects in the database. Read only.

Syntax*control*.**ViewCount****Type**

Long

Range

Greater than or equal to 0.

Example

```
Dim lCount as Long
lCount = Database.ViewCount
```

ViewExists

Description

Specifies if the View object exists in the database. Read only.

Syntax*control*.**ViewExists**(*ViewName* As String)**Type**

Boolean

Range

Value	Description
True	View object exists.
False	View object does not exist.

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View object.	String	Not applicable.

Example

```
Dim ExistingView As Boolean
ExistingView = Database.ViewExists("Acquisition")
```

ViewIndex

Description

Index of the View object. Read only.

Syntax

control.**ViewIndex** (*ViewName* As String)

Type

Long

Range

Minimum: 0

Maximum: **ViewCount** -1

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View object.	String	Not applicable.

Example

```
Dim lIndex as Long
("Acquisition")
```

ViewName

Description

Name of the View object in the database. Read only.

Syntax

control.**ViewName**(*Index* As Long)

Type

String

Range

Not applicable

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the View object.	Long	Minimum: 0 Maximum: ViewCount -1

Example

```
Dim Name as String
Name = Database.ViewName(0)
```

Methods

AboutBox

Description

Displays the HSDatabase control About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

HSDatabase1.AboutBox

AddArc

Description

Adds an Arc object to the specified View in the Database.

Syntax

control.**AddArc**(*ViewName* As String, *ArcName* As String) As HSArc

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that will contain the new Arc object.	String	Not applicable.
<i>ArcName</i>	Name of the Arc object that will be added.	String	Not applicable.

Return

HSArc

Range

Not applicable.

Example

```
Dim NewArcObject As HSArc
Set NewArcObject = _
    HSDatabase1.AddArc("Builder", "ArcEntity")
```

AddImage

Description

Adds a new Image to the specified View in the Database.

Syntax

control.**AddImage**(*ViewName* As String, *ImageName* As String, *ImageType* As VARIANT) As HSIImage

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that will contain the new Image.	String	Not applicable.
<i>ImageName</i>	Name of the Image that will be added.	String	Not applicable.
<i>ImageType</i>	Type of the Image that will be added.	VARIANT	hsImageReserved, hsImage8bppGreyScale, hsImage16bppGreyScale, hsImage16bppHeightMap, hsImage32bppRgb

Return

HSImage

Range

Not applicable.

Example

```
Dim NewImage As HSImage
Set NewImage = _
    HSDatabase1.AddImage("Acquisition","Modified Image",_
    hsImage8bppGreyScale)
```

AddLine

Description

Adds a Line object to the specified View in the Database.

Syntax

```
control.AddLine(ViewName As String, LineName As String) As HSLine
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that will contain the new Line object.	String	Not applicable.
<i>LineName</i>	Name of the Line object that will be added.	String	Not applicable.

Return

HSLine

Range

Not applicable.

Example

```
'Build a Line with Class Library
Dim lLine As HSLine
Set lLine = myDatabase.AddLine ("View", "Line")
    lLine.Calibration = lImageCalibration
    lLine.Calibrated = False
    lLine.StartPoint.ImageX = 100
    lLine.StartPoint.ImageY = 100
    lLine.EndPoint.ImageX = 300
```

```
lLine.EndPoint.ImageY = 300
```

AddPoint

Description

Adds a Point object to the specified View in the Database.

Syntax

```
control.AddPoint(ViewName As String, PointName As String) As HSPoint
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that will contain the new Point object.	String	Not applicable.
<i>PointName</i>	Name of the Point object that will be added.	String	Not applicable.

Return

HSPoint

Range

Not applicable.

Example

```
'Build a Point with Class Library
Dim lPoint as HSPoint
Set lPoint = myDatabase.AddPoint ("View", "Point")
    lPoint.Calibration = lImageCalibration
    lPoint.ImageX = 320
    lPoint.ImageY = 240
```

AddView

Description

Adds a new View to the Database.

Syntax

```
control.AddView(ViewName As String) As HSView
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View to be added.	String	Not applicable.

Return

HSView

Range

Not applicable.

Example

```
Dim MyView As HSView
Set MyView = HSDatabase1.AddView("View9")
```

CopyArc

Description

Creates a copy of a selected Arc object.

Syntax

control.**CopyArc**(SourceArc As HSArc, ViewName As String, NewArcName As String) As HSArc

Parameter(s)

Parameter	Description	Type	Range
SourceArc	Arc object to be copied.	HSArc	Not applicable.
ViewName	Name of the View that will contain the new Arc object.	String	Not applicable.
NewArcName	Name of the Arc object that will be added.	String	Not applicable.

Return

HSArc

Range

Not applicable.

Example

```
Dim NewArc As HSArc
Dim SourceArc As HSArc
Set SourceArc = HSDatabase1.Arc("ArcFinder","Arc")
Set NewArc = _
    HSDatabase1.CopyArc(SourceArc, "ArcFinder","NewArc")
```

CopyImage

Description

Creates a copy of a selected Image.

Syntax

control.**CopyImage**(SourceImage As HSImage, ViewName As HSView, ImageName As String) As HSImage

Parameter(s)

Parameter	Description	Type	Range
SourceImage	Image to be copied.	HSImage	Not applicable.
ViewName	Name of the View that will contain the new Image.	String	Not applicable.
ImageName	Name of the Image that will be added.	String	Not applicable.

Return

HSImage

Range

Not applicable.

Example

```
Dim MyNewImage As HSImage
Dim SourceImage As HSImage
Set SourceImage = _
    HSDatabase1.Image("Acquisition","Image")
```



```
Set MyNewImage = _
    HSDatabase1.CopyImage(SourceImage, "Acquisition","MyImage")
```

CopyLine

Description

Creates a copy of a selected Line object.

Syntax

```
control.CopyLine(SourceArc As HSLine, ViewName As String, NewLineName As String) As
HSLine
```

Parameter(s)

Parameter	Description	Type	Range
<i>SourceLine</i>	Line object to be copied.	HSLine	Not applicable.
<i>ViewName</i>	Name of the View that will contain the new Line object.	String	Not applicable.
<i>NewLineName</i>	Name of the Line object that will be added.	String	Not applicable.

Return

HSLine

Range

Not applicable.

Example

```
Dim NewLine As HSLine
Dim SourceLine As HSLine
Set SourceLine = HSDatabase1.Image("LineFinder","Line")
Set NewLine = _
    HSDatabase1.CopyLine(SourceLine, "LineFinder","NewLine")
```

CopyPoint

Description

Creates a copy of a selected Point object.

Syntax

```
control.CopyPoint(SourcePoint As HSPoint, ViewName As String, NewPointName As String) As
HSPoint
```

Parameter(s)

Parameter	Description	Type	Range
<i>SourcePoint</i>	Point object to be copied.	HSPoint	Not applicable.
<i>ViewName</i>	Name of the View that will contain the new Point object.	String	Not applicable.
<i>NewPointName</i>	Name of the Point object that will be added.	String	Not applicable.

Return

HSPoint

Range

Not applicable.

Example

```
Dim NewPoint As HSPoint
Dim SourcePoint As HSPoint
Set SourcePoint = HSDatabase1.Image("PointFinder","Point")
Set NewPoint = _
    HSDatabase1.CopyPoint(SourcePoint, "PointFinder","NewPoint")
```

CopyScene

Description

Creates a copy of a selected Scene.

Syntax

control.**CopyScene**(*SourceScene* As HSScene, *ViewName* As String, *Scene* As String)

Parameter(s)

Parameter	Description	Type	Range
<i>SourceScene</i>	Scene to be copied.	HSScene	Not applicable.
<i>ViewName</i>	Name of the View that will contain the new Scene.	String	Not applicable.
<i>SceneName</i>	Name of the Scene that will be added.	String	Not applicable.

Return

HSScene

Range

Not applicable.

Example

```
Dim MyNewScene As HSScene
Dim SourceScene As HSScene
Set SourceScene = HSDatabase1.Scene ("Acquisition","Model Scene")
Set MyNewScene = HSDatabase1.CopyScene _
    (SourceScene, "Acquisition","MyScene")
```

CopyView

Description

Creates a copy of a selected View.

Syntax

control.**CopyView**(*SourceView* As HSView, *ViewName* As String) As HSView

Parameter(s)

Parameter	Description	Type	Range
<i>SourceView</i>	Scene to be copied.	HSView	Not applicable.
<i>ViewName</i>	Name of the new View that will be created.	String	Not applicable.

Return

HSView

Range

Not applicable.

Example

```
Dim NewView As HSVView
Dim SourceView As HSVView
Set SourceView = HSDatabase1.View ("Locator")
Set NewView = HSDatabase1.CopyView(SourceView, "NewView")
```

GetDispatchPointer

Description

Reserved for internal use.

Load

Description

Loads the content of a Database file in the current Database.

Syntax

control.**Load**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Path and name of the file that will be loaded.	Variant	If the parameter is a String then the concept of range does not apply. All others variable types are not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.Load("../DatabaseFiles\DbFile2.hdb")
Succeed = HSDatabase2.Load()
```

LoadConfiguration

Description

Reserved for internal use.

Modify

Description

Increments the **ModifyCount** by one and updates the content of the Database.

Syntax

control.**Modify**()

Parameter(s)

None

Return

No return value.

Example

HSDatabase1.Modify

RemoveArc

Description

Removes an Arc object from the selected View.

Syntax*control*.**RemoveArc**(*ViewName* As String, *ArcName* As String) As Boolean**Parameter(s)**

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>ArcName</i>	Name of the object to be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RemoveArc _
    ("Builder","ArcEntity")
```

RemoveImage

Description

Removes an Image from the selected View.

Syntax*control*.**RemoveImage**(*ViewName* As String, *ImageName* As String) As Boolean**Parameter(s)**

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the Image.	String	Not applicable.
<i>ImageName</i>	Name of the Image to be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
```

```
Succeed = HSDatabase1.RemoveImage _
    ("Acquisition", "Image")
```

RemoveLine

Description

Removes a Line object from the selected View.

Syntax

```
control.RemoveLine(ViewName As String, LineName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>LineName</i>	Name of the object to be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RemoveLine _
    ("LineFinder", "Line")
```

RemovePoint

Description

Removes a Point object from the selected View.

Syntax

```
control.RemovePoint(ViewName As String, PointName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>PointName</i>	Name of the object to be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RemovePoint _
    ("PointFinder", "PointEntity")
```

RemoveScene

Description

Removes a Scene from the selected View.

Syntax

control.**RemoveScene**(*ViewName* As String, *SceneName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the Scene.	String	Not applicable.
<i>SceneName</i>	Name of the Scene to be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RemoveScene "Locator", "Contours Scene")
```

RemoveView

Description

Removes the selected View from the Database.

Syntax

control.**RemoveView**(*ViewName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View to be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RemoveView("Acquisition")
```

RenameArc

Description

Renames the selected Arc object.

Syntax

control.**RenameArc**(*ViewName* As String, *CurrentArcName* As String, *NewArcName* As String) As Boolean]

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>CurrentArcName</i>	Current name of the object.	String	Not applicable.
<i>NewArcName</i>	New name of the object.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RenameArc("ArcFinder","Arc","Arc_00")
```

RenameImage

Description

Renames the selected Image.

Syntax

```
control.RenameImage(ViewName As String, CurrentImageName As String, NewImageName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the Image.	String	Not applicable.
<i>CurrentImageName</i>	Current name of the Image to be renamed.	String	Not applicable.
<i>NewImageName</i>	New name of the Image to be renamed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RenameImage("Acquisition","Image","Input Image")
```

RenameLine

Description

Renames the selected Line object.

Syntax

```
control.RenameLine(ViewName As String, CurrentLineName As String, NewLineName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>CurrentLineName</i>	Current name of the object.	String	Not applicable.
<i>NewLineName</i>	New name of the object.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RenameLine("LineFinder","Line","Line_00")
```

RenamePoint

Description

Renames the selected Point object.

Syntax

```
control.RenamePoint(ViewName As String, CurrentPointName As String, NewPointName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the object.	String	Not applicable.
<i>CurrentPointName</i>	Current name of the object.	String	Not applicable.
<i>NewPointName</i>	New name of the object.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RenamePoint("PointFinder","Point","Point00")
```

RenameScene

Description

Renames the selected Scene.

Syntax

```
control.RenameScene(ViewName As String, CurrentSceneName As String, NewSceneName As String) As Boolean
```


Parameter(s)

Parameter	Description	Type	Range
<i>ViewName</i>	Name of the View that contains the Image.	String	Not applicable.
<i>CurrentSceneName</i>	Current name of the Scene to be renamed.	String	Not applicable.
<i>NewSceneName</i>	New name of the Scene to be renamed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RenameScene _
    ("Locator", "Contours Scene", "Edges Scene")
```

RenameView

Description

Renames the selected View.

Syntax

control.**RenameView**(*CurrentViewName* As String, *NewViewName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>CurrentViewName</i>	Current name of the View to be renamed.	String	Not applicable.
<i>NewViewName</i>	New name of the View to be renamed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.RenameView ("Acquisition", "Input View")
```

Reset

Description

Clears the content of the Database.

Syntax

control.**Reset**()

Parameter(s)

None

Return

No return value.

Example

```
HSDatabase1.Reset
```

Save

Description

Saves the contents of the Database in the specified file.

Syntax

```
control.Save([FileName As Variant]) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file where the content of the Database will be saved.	Variant	If the parameter is a String then the concept of range does not apply. All others variable types are not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSDatabase1.Save("../DatabaseFiles\DBase1.hdb")
Succeed = HSDatabase2.Save()
```

SaveConfiguration

Description

Reserved for internal use.

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

HSErrorObject

Description

An HSErrorObject control contains information on an error that occurs within a specific control. Each instance of a HexSight control contains such an error object which can be accessed through the **ErrorObject** property.

Properties

Description

Description

Short description of the error.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Information As String
Information = Process.ErrorObject.Description
```

HelpContext

Description

Context ID for a topic in a Help file.

Syntax

control.**HelpContext**

Type

Long

Range

Not applicable.

Remark(s)

Always 0

Example

```
Dim Value As Long
Value = Process.ErrorObject.HelpContext
```

HelpFile

Description

Fully qualified path to a Help file.

Syntax*control.HelpFile***Type**

String

Range

Not applicable.

Remark(s)

Always returns empty string " ".

Example

```
Dim Filename As String
Filename = Process.ErrorObject.HelpFile
```

Number

Description

Identification number of the error.

Syntax*control.Number***Type**

Long

Range

Not applicable.

Example

```
Dim ErrorNumber As Long
ErrorNumber = Process.ErrorObject.Number
```

Source

Description

Name of the component that has generated the error.

Syntax*control.Source***Type**

String

Range

Not applicable.

Example

```
Dim Name As String
Name = Process.ErrorObject.Source
```

TypeName

Description

ActiveX type name for the object. Read only.

Syntax*control.TypeName***Type**

String

Range

Not applicable.

Remark(s)

This property is always "HSErrorObject"

Example

```
Dim ObjectType As String
ObjectType = Error1.TypeName
```

Version

Description

String containing the current HexSight version and build information. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = ErrorObject.Version.
```

Methods

AboutBoxCustom

Description

Reserved for internal use.

Clear

Description

Clears all property settings of the **HSErrorObject** object.

Syntax

control.**Clear**

Parameter(s)

None

Return

No return value.

Example

```
Process.ErrorObject.Clear
```

Events

This object has no event.

HSImage

Description

An HSImage control contains a matrix of pixels, typically grey-scale, that represents an object's 2D appearance. An HSImage control can be included in an **HSVView** control.

Properties

AveragePixelHeight

Description

Average calibrated height of the Image pixels, expressed in the calibrated units specified by the **UnitsLength** property.

Syntax

control.AveragePixelHeight [= Value As Single]

Type

Single

Range

Greater than or equal to zero.

Remark(s)

If the image is not calibrated then the **AveragePixelHeight** will be equal to zero.

Example

```
Dim Value As Single
Value = HSImage1.AveragePixelHeight
```

AveragePixelWidth

Description

Average calibrated width of the Image pixels, expressed in the calibrated units specified by the **UnitsLength** property.

Syntax

control.AveragePixelWidth [= Value As Single]

Type

Single

Range

Greater than or equal to zero.

Remark(s)

If the image is not calibrated then the **AveragePixelWidth** will be equal to zero.

Example

```
Dim Value As Single
Value = HSImage1.AveragePixelWidth
```

BackgroundColor

Description

Sets the greylevel value used as a uniform background when generating a synthetic Image. This background value is applied only when the synthetic image is rendered upon calling the **Render** method.

Syntax

control.**BackgroundColor** [= Value as Long]

Type

Long

Range

Minimum: 0

Maximum: 255

Example

```
Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
' Draw the synthetic bitmap over a uniform background
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
HSImage1.AddCircleShape 320, 240, 50, 100
HSImage1.Render
```

BackgroundImage

Description

Sets the bitmap used as the background when generating a synthetic Image. This background bitmap is applied only when the synthetic image is rendered upon calling the **Render** method.

Syntax

control.**BackgroundImage** [= Value as HSImage]

Type

HSImage

Range

Any valid HSImage object.

Example

```
Dim HSImage1 As HSImage
Dim HSImage2 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")
Set HSImage2 = AppCtl.Database.AddImage("Locator", "Image")
```



```

HSImage1.Width = 640
HSImage1.Height = 480
' Draw the synthetic bitmap over an existing image
Set HSImage1.BackgroundImage = HSImage2
HSImage1.ResetShapes
HSImage1.AddCircleShape 320, 240, 50, 100
HSImage1.Render

```

BoundaryBottom

Description

Minimum Y coordinate value of the image with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryBottom** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```

Dim Value As Single
Value = HSImage1.BoundaryBottom

```

BoundaryLeft

Description

Minimum X coordinate value of the image with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryLeft** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```

Dim Value As Single
Value = HSImage1.BoundaryLeft

```

BoundaryRight

Description

Maximum X coordinate value of the image with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryRight** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
Dim Value As Single
Value = HSImage1.BoundaryRight
```

BoundaryTop

Description

Maximum Y coordinate value of the image with respect to the World or the Image coordinate system.

Syntax

*control.***BoundaryTop** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
Dim Value As Single
Value = HSImage1.BoundaryTop
```

Calibration

Description

Returns the internal **HSCalibration** object of the control. It is used to access or modify the calibration information.

Syntax

*control.***Calibration** [= *Value* As HSCalibration]

Type

HSCalibration

Range

Not applicable.

Example

```
Dim Calibration as HSCalibration
Calibration = InputImage.Calibration
OutputImage.Calibration = Calibration
```

Description

Description

String that describes the standard usage of the HSImage control.

Syntax

*control.***Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = HSImage1.Description
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

Height

Description

Height of the Image expressed in pixels.

Syntax

control.**Height** [= *Value* As Single]

Type

Single

Range

Greater than or equal to zero.

Example

```
Dim Value As Single
Value = HSImage1.Height
```

ImageCoordinateSystemType

Description

Selects whether the coordinate system used to express pixel coordinates in the image is right-handed or left-handed. In a right-handed Image coordinate system (default), the origin is located at the bottom left corner of the image and the Y-axis is directed upward. In a left-handed Image coordinate system, the origin is located at the top left corner of the image and the Y-axis is directed downward.

Syntax

control.**ImageCoordinateSystemType** [= *Value* as hsCoordinateSystemType]

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRightHanded	0	The Image coordinate is right-handed (origin located at the bottom left corner and Y-axis directed upward).
hsLeftHanded	1	The Image coordinate is left-handed (origin located at the top left corner and Y-axis directed downward).

Remark(s)

A left-handed image as well as a right-handed image can be calibrated into either a left-handed or a right-handed World coordinate system. When the **ImageCoordinateSystemType** property of an image is changed, the image must be recalibrated to avoid any inconsistencies.

Example

```
Dim CurrentType as hsCoordinateSystemType
CurrentType = HSImage1.ImageCoordinateSystemType
```

Information

Description

String that provides information pertaining to the object. For example: Name, Boundary coordinates, Image Coordinate System, Timestamp, etc.

Syntax

```
control.Information [= Value As String]
```

Type

String

Range

Not applicable

Example

```
Dim Info As String
Info = HSImage1.Information
```

ModifyCount

Description

Number of modifications that have been made to the Image. Read only.

Syntax

```
control.ModifyCount
```

Type

Long

Range

Greater than or equal to zero.

Example

```
Dim Count as Long
Count = HSImage1.ModifyCount
```

Name

Description

Name used to access the Image object.

Syntax

control.**Name** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = HSImage1.Name
HSImage2.Name = "Image_00"
```

OriginX

Description

Horizontal (X) origin of the Image, expressed in pixel units. Along with **OriginY**, this property can be used to specify the origin of an image cropped externally to HexSight.

Syntax

control.**OriginX** [= Value As Long]

Type

Long

Range

Greater than or equal to zero.

Example

```
Dim X0 as Long
Dim Y0 as Long
' Set the origin of the cropped image to (110, 240)
HSImage1.OriginX = 110
HSImage1.OriginY = 240
...
' Use the origin of the cropped bitmap to offset it
' correctly in a Picture Box
X0 = HSImage1.OriginX
Y0 = HSImage1.OriginY
```

OriginY

Description

Vertical (Y) origin of the Image, expressed in pixel units. Along with **OriginX**, this property can be used to specific the origin of an image cropped externally to HexSight.

Syntax

control.**OriginY** [= Value As Long]

Type

Long

Range

Greater than or equal to zero.

Example

```

Dim X0 as Long
Dim Y0 as Long
' Set the origin of the cropped image to (110, 240)
HSImage1.OriginX = 110
HSImage1.OriginY = 240
...
' Use the origin of the cropped bitmap to offset it
' correctly in a Picture Box
X0 = HSImage1.OriginX
Y0 = HSImage1.OriginY

```

Pixel

Description

Grey-scale value of the selected pixel.

Syntax*control.Pixel*(X as Long, Y as Long) [= *NewValue* as Integer]**Type**

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Parameter	Description	Type	Range
X	Column number of the selected pixel in the Image.	Long	Minimum: 0 Maximum: Height
Y	Row number of the selected pixel in the Image.	Long	Minimum: 0 Maximum: Width

Example

```

Dim Value as Integer
Value = HSImage1.Pixel(500,230)
HSImage2.Pixel(100,50) = 110

```

RawImagePointer

Description

OLE_HANDLE that represents the Image's start address in the memory

Syntax*control.RawImagePointer* [= *Value* As OLE_HANDLE]

Type

OLE_HANDLE

Range

Not applicable.

Example

```
Dim ImagePtr As OLE_HANDLE
ImagePtr = HSImage1.RawImagePointer
```

ShapesCalibrated

Description

When True, indicates that shapes are drawn in synthetic images using calibrated units. Otherwise, the shapes are drawn using pixel units.

Syntax

control.**ShapesCalibrated** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
' Draw the synthetic bitmap over a uniform background
HSImage1.BackgroundColor = 200
HSImage1.CalibratedShapes = True
HSImage1.ResetShapes
' Draw a 3cm radius circle at world coordinate (10cm, 10cm)
HSImage1.AddCircleShape 10, 10, 3, 100
HSImage1.Render
```

SubSamplingFactor

Description

Reserved for internal use.

Type

Description

Type of the Image object. Read only.

Syntax

control.**Type**

Type

hsImageType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsImageReserved	0	Reserved internal type.
hsImage8bppGreyScale	1	8 bits per pixel grey-scale image.
hsImage16bppGreyScale	20	16 bits per pixel grey-scale image.
hsImage16bppHeightMap	21	16 bits per pixel height map.
hsImage32bppRgb	10	32 bits per pixel color image.

Example

```
Dim ImageType As hsImageType
ImageType = Image.Type
```

TypeName

Description

ActiveX type name for the Image object. Read only.

Syntax*control*.**TypeName****Type**

String

Range

Not applicable.

Remark(s)

This property is always "HSImage"

Example

```
Dim ObjectType As String
ObjectType = HSImage1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax*control*.**UnitsLength** [= *Value* as hsUnitsLength]**Type**

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter

Constant	Value	Description
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit as hsUnitsLength
Unit = HSImage1.UnitsLength
HSImage2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* as hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit as hsUnitsOrientation
Unit = HSImage1.UnitsOrientation
HSImage2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```
Dim Version As String
Version = HSImage1.Version
```

Width

Description

Width of the Image expressed in pixels.

Syntax

```
control.Width[= Value as Single]
```

Type

Single

Range

Greater than or equal to zero.

Example

```
Dim Value As Single
Value = HSImage1.Width
```

Methods

AboutBox

Description

Displays the HSImage About box.

Syntax

```
control.AboutBox
```

Parameter(s)

None

Return

No return value.

Example

```
HSImage1.AboutBox
```

AddCircleShape

Description

Draws a filled circle in a synthetic image. The shape is actually drawn only when the synthetic image is rendered upon calling the **Render** method. A transform can be applied to the shape using the **ApplyTransform** method before rendering the image.

Syntax

```
control.AddCircleShape(X As Single, Y As Single, Radius As Single, Color As Long)
```

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the center.	Single	Not applicable.

Parameter	Description	Type	Range
<i>Y</i>	Y coordinate of the center.	Single	Not applicable.
<i>Radius</i>	Radius of the circle.	Single	Greater than or equal to zero.
<i>Color</i>	Greylevel used to draw and fill the circle.	Long	Minimum:0 Maximum:255

Return

No return value.

Example

```
Dim HSImage1 As HSImage
```

```
AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")
```

```
HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
HSImage1.AddCircleShape 320, 240, 50, 100
HSImage1.Render
```

AddRectangleShape

Description

Draws a filled rectangle in a synthetic image. The shape is actually drawn only when the synthetic image is rendered upon calling the **Render** method. A transform can be applied to the shape using the **ApplyTransform** method before rendering the image.

Syntax

control.**AddRectangleShape** (*X1* As Single, *Y1* As Single, *X2* As Single, *Y2* As Single, *Color* as Long)

Parameter(s)

Parameter	Description	Type	Range
<i>X1</i>	X coordinate of the first corner.	Single	Not applicable.
<i>Y1</i>	Y coordinate of the first corner.	Single	Not applicable.
<i>X2</i>	X coordinate of the opposite corner.	Single	Not applicable.
<i>Y2</i>	Y coordinate of the opposite corner.	Single	Not applicable.
<i>Color</i>	Greylevel used to draw and fill the rectangle.	Long	Minimum:0 Maximum:255

Return

No return value.

Example

```
Dim HSImage1 As HSImage
```

```

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
HSImage1.AddRectangleShape 100, 100, 540, 380, 100
HSImage1.Render

```

ApplyTransform

Description

Applies a rigid transform to all the shapes drawn in a synthetic image. The shapes are actually transformed only when the synthetic image is rendered upon calling the **Render** method. A transform can be applied to the shapes in a synthetic image even though another transform has already been applied and the image has been rendered. In that case, the shapes will be transformed according to their original location in the image. The transform components are applied in the following order: rotation, scale and (X,Y) translation.

Syntax

control.**ApplyTransform**(X As Single, Y As Single, Rotation As Single, Scale As Single)

Parameter(s)

Parameter	Description	Type	Range
X	X translation component.	Single	Not applicable.
Y	Y translation component.	Single	Not applicable.
<i>Rotation</i>	Rotation component.	Single	Not applicable.
<i>Scale</i>	Scale components.	Single	Not applicable.

Return

No return value.

Example

```

Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
HSImage1.AddRectangleShape -10, -10, 10, 10, 100
' Rotate the rectangle by 45 degrees and translate it
' to the center of the image

```

```
HSImage1.ApplyTransform 320, 240, 45, 1
HSImage1.Render
```

ArcTo

Description

Adds an arc segment to the last shape created in a synthetic image. The arc segment is created between the end point of last segment added to the shape and the end point specified by the **XEnd** and **YEnd** parameters. A new shape is created in a synthetic image by calling the **StartShape** method. The shape is actually drawn only when the synthetic image is rendered upon calling the **Render** method. A transform can be applied to the shape using the **ApplyTransform** method before rendering the image.

Syntax

Sub *control*.**ArcTo**(XEnd As Single, YEnd As Single, Radius As Single, Type As hsArcType)

Parameter(s)

Parameter	Description	Type	Range
<i>XEnd</i>	X coordinate of the end point of the segment.	Single	Not applicable.
<i>YEnd</i>	Y coordinate of the end point of the segment.	Single	Not applicable.
<i>Radius</i>	Radius of the arc segment.	Single	Not applicable.
<i>Type</i>	Scale components.	hsArcType	Enum. See table below.

hsArcType	Value	Description
hsConvexOutside	0	Convex arc, with the center toward the outside of the shape.
hsConcaveOutside	1	Concave arc, with the center toward the outside of the shape.
hsConvexInside	2	Convex arc, with the center toward the inside of the shape.
hsConcaveInside	3	Concave arc, with the center toward the inside of the shape.

Return

No return value.

Example

```
Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
' Draw a 100 pixel radius dark circle centered at 320,240
HSImage1.StartShape 220, 240, 50
HSImage1.ArcTo 320, 340, 100, hsConvexInside
HSImage1.ArcTo 420, 240, 100, hsConvexInside
```

```

HSImage1.ArcTo 320, 140, 100, hsConvexInside
HSImage1.ArcTo 220, 240, 100, hsConvexInside
HSImage1.Render

```

AttachImagePointer

Description

Reserved for internal use.

Calibrate

Description

Reserved for internal use.

ComputeArea

Description

Computes the area of the specified shape in a synthetic image. The **ComputeShapes** method must be called before **ComputeArea**.

Syntax

control.**ComputeArea**(Index As Long) As Single

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the shape	Long	Not applicable.

Return

Single.

Example

```

Dim lArea AS Single
Dim Image As HSImage
Set lImage As HSImage = lDatabase.Image("Synthetic", "Part")
lImage.BackgroundColor = 200
lImage.ResetShapes
lImage.AddRectangleShape 100,100,540,380,100
lImage = ComputeShapes
lArea = ComputeArea
lImage.Render

```

ComputeShapes

Description

Computes the properties of the shapes in a synthetic image. The ComputeShapes method must be called before **ComputeArea**.

Syntax

control.**ComputeShapes**()

Parameter(s)

None

Return

No return value.

Example

```
Dim lArea AS Single
Dim Image AS HSImage
Set lImage AS HSImage = lDatabase.Image("Synthetic","Part")
lImage.BackgroundColor = 200
lImage.ResetShapes
lImage.AddRectangleShape 100,100,540,380,100
lImage = ComputeShapes
lArea = ComputeArea
lImage.Render
```

Copy

Description

Copies in the current Image the Image provided through the *RawImagePointer* parameter.

Syntax

control.**Copy**(*RawImagePointer* As OLE_HANDLE, *Width* as Integer, *Height* as Integer) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>RawImagePointer</i>	OLE_HANDLE that gives an access to the content of the Image to be copied.	OLE_HANDLE	Not applicable.
<i>Width</i>	Width of the Image to be copied.	Integer	Greater than or equal to zero.
<i>Height</i>	Height of the Image to be copied.	Integer	Greater than or equal to zero.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed AS Boolean
Dim ImagePtr AS OLE_HANDLE
Dim W AS Integer
Dim H AS Integer
ImagePtr = HSImage2.RawImagePointer
W = HSImage2.Width
H = HSImage2.Height
Succeed = HSImage1.Copy(ImagePtr, W,H)
```

CopyFromSafeArray

Description

Copies the data contained in a generic array into the pixel array of the Image.

Syntax

control.**CopyFromSafeArray**(Array As Variant, VerticalFlip As Boolean) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>Array</i>	Array that contains the data to be copied.	Variant	Not applicable
<i>VerticalFlip</i>	When True, the data is flipped vertically during the copy.	Boolean	True, False

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim HSImage1 As HSImage
Dim HSImage2 As HSImage
Dim HSImageArray() As Byte
...
' Make a flipped copy of Image1 into Image2
ReDim HSImageArray( HSImage1.Width * HSImage1.Height )
HSImage1.CopyToSafeArray( HSImageArray, FALSE )
HSImage2.CopyFromSafeArray( HSImageArray, TRUE )
```

CopyToSafeArray

Description

Copies the data contained in the pixel array of the Image into a generic array.

Syntax

control.**CopyToSafeArray**(Array As Variant, VerticalFlip As Boolean) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>Array</i>	Array into which the data will be copied.	Variant	Not applicable
<i>VerticalFlip</i>	When True, the data is flipped vertically during the copy.	Boolean	True, False

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim HSImage1 As HSImage
Dim HSImage2 As HSImage
Dim HSImageArray() As Byte
...
```



```
' Make a flipped copy of Image1 into Image2
ReDim HSImageArray( HSImage1.Width * HSImage1.Height )
HSImage1.CopyToSafeArray( HSImageArray, FALSE )
HSImage2.CopyFromSafeArray( HSImageArray, TRUE )
```

Create

Description

Reserved for internal use

CreateEx

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

ImageToWorld

Description

This method takes two parameters (X and Y) that represent a point in the Image coordinate system. When the method is executed, the two entered parameters are modified to express the same point with respect to the World coordinate system.

Syntax

control.**ImageToWorld**(X As Single, Y As Single) As Boolean

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point.	Single	Any Single value.
Y	Y coordinate of the point.	Single	Any Single value.

ReturnReturn

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If the Image is not calibrated then the method will return False.

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 200
Y1 = 350
Succeed = HSImage1.ImageToWorld(X1, Y1)
```

ImageToWorldArray

Description

Reserved for internal use.

LineTo

Description

Adds a linear segment to the last shape created in a synthetic image. The linear segment is created between the end point of last segment added to the shape and the end point specified by the **XEnd** and **YEnd** parameters. A new shape is created in a synthetic image by calling the **StartShape** method. The shape is actually drawn only when the synthetic image is rendered upon calling the **Render** method. A transform can be applied to the shape using the **ApplyTransform** method before rendering the image.

Syntax

control.**LineTo**(XEnd As Single, YEnd As Single)

Parameter(s)

Parameter	Description	Type	Range
<i>XEnd</i>	X coordinate of the end point of the segment.	Single	Not applicable.
<i>YEnd</i>	Y coordinate of the end point of the segment.	Single	Not applicable.

Return

No return value.

Example

```
Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
' Draw a dark square centered at 320,240
HSImage1.StartShape 220, 140, 50
HSImage1.LineTo 420, 140
HSImage1.LineTo 420, 340
HSImage1.LineTo 220, 340
HSImage1.LineTo 220, 140
HSImage1.Render
```

Load

Description

Loads an Image from a Windows Bitmap file. (*.bmp) or a HexSight Image file (*.hig). A hig image file conserves the image calibration as well as all other image properties.

Syntax

control.**Load**([*FileName* As Variant], [BMPCoordinateSystem As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file to load. Optional.	Variant	Not applicable.
<i>BMPCoordinateSystem</i>	Coordinate system to be used to express pixel coordinates in a bmp (Bitmap) image. Not applicable to hig image format.	hsCoordinateSystemType	Enum, see table below.

Constant	Value	Description
hsRightHanded	0	The Image coordinate is right-handed (origin located at the bottom left corner and Y-axis directed upward).
hsLeftHanded	1	The Image coordinate is left-handed (origin located at the top left corner and Y-axis directed downward).

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file open browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSImage1.Load("..\BMPFiles\Image.bmp")
Succeed = HSImage2.Load("..\Images\Image.hig")
```

Modify

Description

Modifies the **ModifyCount** property.

Syntax

control.**Modify**()

Parameter(s)

None

Return

No return value.

Example

```
HSImage1.Modify
```

Render

Description

Renders a synthetic image. This method must be called after the background of the image is set and the shapes are drawn to generate the pixel map of a synthetic image.

Syntax

control.**Render** As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
HSImage1.ResetShapes
HSImage1.AddRectangleShape 100, 100, 540, 380, 100
HSImage1.Render
```

ResetShapes

Description

Clears all the shapes in a synthetic image.

Syntax

control.**ResetShapes**()

Parameter(s)

None

Return

No return value.

Example

```
Dim HSImage1 As HSImage

AppCtl.Database.AddView "MyView"
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")

HSImage1.Width = 640
HSImage1.Height = 480
HSImage1.BackgroundColor = 200
```

HSImage1.**ResetShapes**

HSImage1.AddRectangleShape 100, 100, 540, 380, 100

HSImage1.Render

Save

Description

Saves the contents of the Image as a Windows Bitmap file to a specified destination.

Syntax

control.**Save**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the image file where the content of the Image will be saved. Optional.	Variant	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file save browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSImage1.Save("..\Images\Image00.hig")
Succeed = HSImage2.Save( )
```

SetColor

Description

Sets the greylevel to be used for drawing a specific shape in a synthetic image.

Syntax

control.**SetColor**(*Index* As Long, *Color* As Long)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the shape.	Long	Not applicable.
<i>Color</i>	Greylevel used to draw and fill the shape.	Long	Minimum: 0 Maximum: 255.

Return

No return value.

Example

```
lImage.SetColor 0, 127
```

SetFirst

Description

Forces a specific shape in a synthetic image to be the first one drawn during rendering. See **Render** method.

Syntax

control.**SetFirst** (*Index* As Long)

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the shape.	Long	Not applicable.

Return

No return value.

Example

```
lImage.SetFirst 4
```

SetForcePositiveSign

Description

Ensures that all shapes in a synthetic image will be drawn counterclockwise during rendering. See **Render** method.

Syntax

control.**SetForcePositiveSign** (*ForcePositiveSign* As Boolean)

Parameter(s)

Parameter	Description	Type	Range
<i>ForcePositiveSign</i>	When True, shapes are rendered counterclockwise.	Boolean	True or False

Return

No return value.

Example

```
lImage.SetForcePositiveSign TRUE
```

StartShape

Description

Creates a filled shape in a synthetic image. The new shape starts at the coordinates specified by the **X** and **Y** parameters. The shape is drawn using arc or linear segments by calling the **ArcTo** and **LineTo** methods. The greylevel used to fill the shape is specified by the **Color** parameter. The shape is actually drawn only when the synthetic image is rendered upon calling the **Render** method. A transform can be applied to the shape using the **ApplyTransform** method before rendering the image.

Syntax

control.**StartShape**(*X* As Single, *Y* As Single, *Color* As Long)

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the starting point of the shape.	Single	Not applicable.
Y	Y coordinate of the starting point of the shape.	Single	Not applicable.
Color	Greylevel used to draw and fill the shape.	Long	Minimum:0 Maximum:255

Return

No return value.

Example

```
Dim HSImage1 As HSImage
```

```
AppCtl.Database.AddView "MyView"
```

```
Set HSImage1 = AppCtl.Database.AddImage("MyView", "Image")
```

```
HSImage1.Width = 640
```

```
HSImage1.Height = 480
```

```
HSImage1.BackgroundColor = 200
```

```
HSImage1.ResetShapes
```

```
HSImage1.StartShape 220, 140, 50
```

```
lImage.ArcTo 420, 140, 100, hsConvexInside
```

```
lImage.LineTo 420, 340
```

```
lImage.ArcTo 220, 340, 100, hsConvexInside
```

```
lImage.LineTo 220, 140
```

```
HSImage1.Render
```

SubSampleFrom**Description**

Reserved for internal use.

WorldToImage**Description**

This method takes two parameters (X and Y) that represent a point in the World coordinate system. When the method is executed the two entered parameters are modified to express the same point with respect to the Image coordinate system.

Syntax

```
control.WorldToImage(X As Single, Y As Single) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point	Single	Any Single value.
Y	Y coordinate of the point	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If the Image is not calibrated then the method will return False.

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 120.3
Y1 = 35.0
Succeed = HSImage1.WorldToImage(X1,Y1)
```

WorldToImageArray

Description

Reserved for internal use

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

HSLine

Description

The HSLine control allows the user to access and manipulate Line entities. Entities can be generated by a HexSight Finder tool, the Builder tool or created programmatically within the Class Library.

A Line entity is defined as a line vector as illustrated in Figure 3. The Vector Point and Angle define the line. Delta Start and Delta End define a segment on the line. Start and End points are the end points of the line segment.

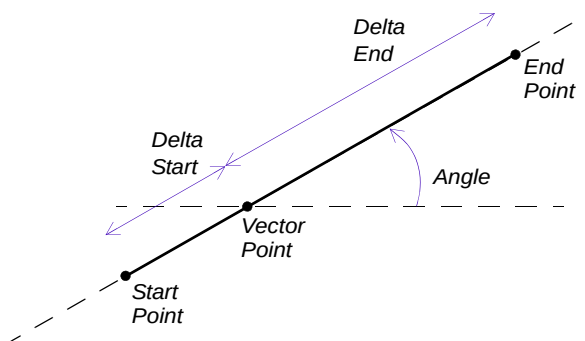


FIGURE 3 *Line Entity Parameters*

Properties

Angle

Description

The angle of the HSLine object. The value of the **Calibrated** property determines whether **Angle** is defined in the Image coordinate system (non-calibrated) or in the World coordinate system (calibrated). When the **Angle** value is modified, the **StartPoint** and **EndPoint** values are automatically recalculated.

Syntax

`control.Angle [= Value As Single]`

Type

Single

Range

Any single value.

Remark(s)

Angles defined in a calibrated coordinate system cannot be recalculated into a non-calibrated coordinate system. The inverse is also true.

Example

```
'Build a Line in Class Library
Dim MyLine As HSLine
```

```

Set MyLine = MyImage.AddLine ("View", "Line")
With MyLine
    .Calibration = MyImage.Calibration
    .Calibrated = False
    .VectorPoint.ImageX = 320
    .VectorPoint.ImageY = 240
    .Angle = 45
    .StartDelta = -10
    .EndDelta = 20
End With

```

Calibrated

Description

Specifies if the entity is defined in a calibrated or non-calibrated coordinate system. When **Calibrated** is false, the entity is defined in the Image Coordinate system and length units are in pixels. When **Calibrated** is True, the entity is defined in the World Coordinate system.

Syntax

control.**Calibrated** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Line properties are expressed in calibrated units.
False	Line properties are expressed in pixels.

Remark(s)

Angles defined in a calibrated coordinate system cannot be recalculated into a non-calibrated coordinate system. The inverse is also true.

Example

```

'Build a Line in Class Library
Dim MyLine As HSLine
Set MyLine = MyImage.AddLine ("View", "Line")
With MyLine
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .VectorPoint.WorldX = 100
    .VectorPoint.WorldY = 100
    .Angle = 90
End With

```

Calibration

Returns the internal **HSCalibration** object of the control. It is used to access or modify the calibration information.

Syntax

control.**Calibration** [= *Value* As HSCalibration]

Type

HSCalibration

Range

Not applicable.

Example

```
'Copy the image calibration to the entity
Dim MyLine As HSLine
MyLine.Calibration = MyImage.Calibration
```

Description

Description

String that describes the standard usage of the HSLine control. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = HSLine1.Description
```

EndDelta

Description

Specifies the end of the Line segment by defining the distance between the end of the segment and the **VectorPoint**. When the **EndDelta** value is modified, the **EndPoint** value is automatically recalculated.

Syntax

control.**EndPoint** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
'Build a Line in Class Library
Dim MyLine As HSLine
Set MyLine = MyImage.AddLine ("View", "Line")
With MyLine
    .Calibration = MyImage.Calibration
```

```

        .Calibrated = False
        .VectorPoint.ImageX = 320
        .VectorPoint.ImageY = 240
        .Angle = 45
        .StartDelta = -10
        .EndDelta = 20
    End With

```

EndPoint

Description

The point at the end of the line segment. When the **EndPoint** value is modified, the **VectorPoint**, **Angle**, **StartDelta** and **EndDelta** values are automatically recalculated.

Syntax

```
control.EndPoint [= Value As HSPoint]
```

Type

HSPoint

Range

Any valid HSPoint object.

Example

```

'Build a Line from two points
Dim MyLine As HSLine
Set MyLine = MyImage.AddLine ("View", "Line")
With MyLine
    .Calibration = MyImage.Calibration
    .Calibrated = False
    .StartPoint.ImageX = 100
    .StartPoint.ImageY = 100
    .EndPoint.ImageX = 300
    .EndPoint.ImageY = 300
End With

```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

```
control.ErrorObject
```

Type

HSErrorObject

Range

Any valid HSPoint object.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

Information

Description

String that provides information pertaining to the object. For example: Name, Boundary coordinates, Image Coordinate System, Timestamp, etc.

Syntax

```
control.Information [= Value As String]
```

Type

String

Range

Not applicable

Example

```
Dim Info As String
Info = HSLine1.Information
```

ModifyCount

Description

Number of modifications that have been made to the Line object. Read only.

Syntax

```
control.ModifyCount
```

Type

Long

Range

Greater than or equal to zero.

Example

```
Dim Count as Long
Count = HSLine1.ModifyCount
```

Name

Description

Name used to access the object.

Syntax

```
control.Name[= Value As String]
```

Type

String

Range

Not applicable.

Example

```
Dim LineName As String
LineName = HSLine1.Name
```

```
HSLine2.Name = "Line_00"
```

StartDelta

Description

Specifies the start of the Line segment by defining the distance between the start of the segment and the **VectorPoint**. When the **StartDelta** value is modified, the **StartPoint** value is automatically recalculated.

Syntax

```
control.EndPoint [= Value As Single]
```

Type

Single

Range

Any Single value.

Remark(s)

Example

```
'Build a Line in Class Library
Dim MyLine As HSLine
Set MyLine = MyImage.AddLine ("View", "Line")
With MyLine
    .Calibration = MyImage.Calibration
    .Calibrated = False
    .VectorPoint.ImageX = 320
    .VectorPoint.ImageY = 240
    .Angle = 45
    .StartDelta = -10
    .EndDelta = 20
End With
```

StartPoint

Description

The point at the start of the line segment. When the **StartPoint** value is modified, the **VectorPoint**, **Angle**, **StartDelta** and **EndDelta** values are automatically recalculated.

Syntax

```
control.StartPoint [= Value As HSPoint]
```

Type

HSPoint

Range

Any valid HSPoint object.

Example

```
'Build a Line from two points
Dim MyLine As HSLine
Set MyLine = MyImage.AddLine ("View", "Line")
```

```

With MyLine
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .StartPoint.ImageX = 100
    .StartPoint.ImageY = 100
    .EndPoint.ImageX = 300
    .EndPoint.ImageY = 300
End With

```

TypeName

Description

ActiveX type name for the Line object. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSLine"

Example

```

Dim ObjectType As String
ObjectType = HSLine1.TypeName

```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* as hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch

Constant	Value	Description
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit as hsUnitsLength
Unit = HSLine1.UnitsLength
HSLine2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* as hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit as hsUnitsOrientation
Unit = HSLine1.UnitsOrientation
HSLine2.UnitsOrientation = hsDegree
```

VectorPoint

Description

Specifies a point through which the line (vector) passes. **VectorPoint** and **Angle** define the line vector. When the **VectorPoint** value is modified, **StartPoint** and **EndPoint** values are automatically recalculated.

Syntax

control.**VectorPoint** [= *Value* as HSPoint]

Type

HSPoint

Range

Any valid HSPoint object.

Example

```
'Build a Line with Class Library
Dim MyLine As HSLine
Set MyLine = MyImage.AddLine ("View", "Line")
With MyLine
```



```

        .Calibration = MyImage.Calibration
        .Calibrated = False
        .VectorPoint.ImageX = 320
        .VectorPoint.ImageY = 240
        .Angle = 45
        .StartDelta = -10
        .EndDelta = 20
    End With

```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```

Dim Version As String
Version = HSLine1.Version

```

Methods

AboutBox

Description

Displays the HSLine About box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Return

No return value.

Example

```

HSLine1.AboutBox

```

CopyFormatFrom

Description

Copies the format of a specified object to the current object.

Syntax

control.**CopyFormatFrom**(*SourceObject* As Object)

Parameter(s)

Name	Description	Type	Range
<i>SourceObject</i>	Object from which the format is copied.	HSScene, HSImage, HSArc, HSLine, or HSPoint.	Not applicable.

Return

No return value.

Example

Line1.**CopyFormatFrom**(Line0)

GetDispatchPointer

Description

Reserved for internal use.

Load

Description

Loads the content of a line file (*.hln).

Syntax

control.**Load**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the hln file to load. Optional.	Variant	Not applicable

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSLine.Load("Line0.hln")
```

NearestPoint

Description

Computes the coordinates of the point on the current HSLine object that is nearest to a specified object called the *SourceEntity*. The value of the **Calibration** property determines whether coordinates are specified in the non-calibrated Image coordinate system or the calibrated World coordinate system.

Syntax

control.**NearestPoint**(*SourceEntity* As Object, *ExtrapolateThis* As Boolean, *ExtrapolateEntity* As Boolean, *Intersected* As Boolean) As HsPoint

Parameter(s)

Name	Description	Type	Range
<i>SourceEntity</i>	Input Entity used to compute nearest point.	HSPoint, HSLine, HSArc	Not applicable.
<i>ExtrapolateThis</i>	Extrapolate current object.	Boolean	True, False
<i>ExtrapolateEntity</i>	Extrapolate the specified SourceEntity.	Boolean	True, False
<i>Intersected</i>	Determine if the calculated point coordinates come from an real intersection (as opposed as an intersection from an extrapolated entity.)	Boolean	True, False

Return

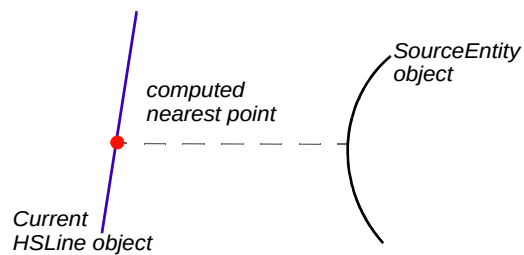
HSPoint

Example

```

Dim Intersect As Boolean
Dim NearestPoint As HSPoint
Set NearestPoint = Line0.NearestPoint(Line1,True,True,Intersect)

```

FIGURE 4 *HSLine— NearestPoint Method***Perpendicular**

Computes the coordinates of a line that is perpendicular to the current HSLine object and that passes through a specified seed point called the *SourcePoint* object.

Syntax

```
control.Perpendicular (SourcePoint As HSPoint) As Object
```

Parameter(s)

Name	Description	Type	Range
<i>SourcePoint</i>	Seed for the perpendicular line	HSPoint	Not applicable.

Return

HSLine

Example

```

Dim Perpendicular As HSLine
Set Perpendicular = Line0.Perpendicular(Point0)
Line1.Perpendicular(Point0)

```

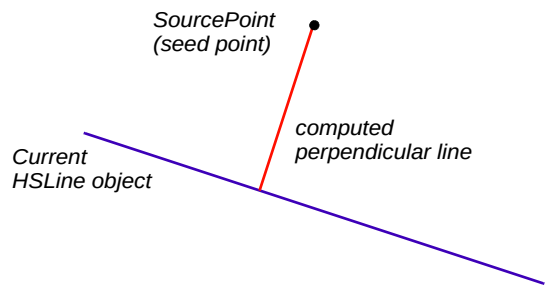


FIGURE 5 *HSLine — Perpendicular Method*

Reset

Description

Reserved for internal use.

Save

Description

Saves the Line entity to a HexSight Line file (*.hln).

Syntax

control.**Save**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the line file (*.hln) to which the line is saved. Optional.	Variant	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSLine1.Save("Line0.hln")
Succeed = HSLine2.Save( )
```

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

HSPoint

Description

The HSPoint control allows the user to access and manipulate Point entities. Entities can be generated by a HexSight Finder tool, the Builder tool or created programatically within the Class Library.

Properties

Calibrated

Description

Specifies if the entity is defined in a calibrated or non-calibrated coordinate system. When **Calibrated** is false, the entity is defined in the Image Coordinate system and length units are in pixels. When **Calibrated** is True, the entity is defined in the World Coordinate system.

Syntax

`control.Calibrated [= Value As Boolean]`

Type

Boolean

Range

Value	Description
True	Point properties are expressed in calibrated units.
False	Point properties are expressed in pixels.

Example

```
' Build a Point with Class Library
Dim MyPoint as HSPoint
Set MyPoint = MyDatabase.AddPoint ("View", "Point")
With MyPoint
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .ImageX = 320
    .ImageY = 240
End With
```

Calibration

Description

Returns the internal **HSCalibration** object of the control. It is used to access or modify the calibration information.

Syntax

control.**Calibration** [= *Value* As HSCalibration]

Type

HSCalibration

Range

Not applicable.

Example

```
' Build a Point with Class Library
Dim MyPoint as HSPoint
Set MyPoint = MyDatabase.AddPoint ("View", "Point")
With MyPoint
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .ImageX = 320
    .ImageY = 240
End With
```

Description

Description

String that describes the standard usage of the HSPoint control.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = HSPoint1.Description
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

ImageX

Description

Y coordinate of the Point object in the Image coordinate system. This property is expressed in pixels.

Syntax

```
control.ImageX [= Value As Single]
```

Type

Single

Range

Unbounded.

Remark(s)

When **ImageX** and **ImageY** values are modified, the **WorldX** and **WorldY** values are automatically recalculated.

Example

```
'Build a Point with Class Library
Dim MyPoint as HSPoint
Set MyPoint = MyDatabase.AddPoint ("View", "Point")
With MyPoint
    .Calibration = MyImage.Calibration
    .Calibrated = True
    .ImageX = 320
    .ImageY = 240
End With
```

ImageY

Description

Y coordinate of the Point object in the Image coordinate system. This property is expressed in pixels.

Syntax

```
control.ImageY [= Value As Single]
```

Type

Single

Range

Unbounded.

Remark(s)

When **ImageX** and **ImageY** values are modified, the **WorldX** and **WorldY** values are automatically recalculated.

Example

```
' Build a Point with Class Library
Dim MyPoint as HSPoint
Set MyPoint = MyDatabase.AddPoint ("View", "Point")
```



```

With MyPoint
    .Calibration = MyImage.Calibration
    .v
    .ImageX = 320
    .ImageY = 240
End With

```

Information

Description

String that provides information pertaining to the object. For example: Name, Boundary coordinates, Image Coordinate System, Timestamp, etc.

Syntax

control.**Information** [= *Value* As String]

Type

String

Range

Not applicable

Example

```

Dim Info As String
Info = HSPoint1.Information

```

ModifyCount

Description

Number of modifications that have been made to the Point. Read only.

Syntax

control.**ModifyCount**

Type

Long

Range

Greater than or equal to zero.

Example

```

Dim Count as Long
Count = HSPoint1.ModifyCount

```

Name

Description

Name used to access the object.

Syntax

control.**Name**[= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim PointName As String
PointName = HSPoint1.Name
HSPoint2.Name = "Point_00"
```

TypeName

Description

ActiveX type name for the object. Read only

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSPoint"

Example

```
Dim ObjectType As String
ObjectType = HSPoint1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* as hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit as hsUnitsLength
Unit = HSPoint1.UnitsLength
```

`HSPoint2.UnitsLength = hsInch`

UnitsOrientation

Description

Active orientation unit.

Syntax

`control.UnitsOrientation [= Value as hsUnitsOrientation]`

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit as hsUnitsOrientation
Unit = HSPoint1.UnitsOrientation
HSPoint2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

`control.Version`

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```
Dim Version As String
Version = HSPoint1.Version
```

WorldX

Description

X coordinate of the Point entity in the World coordinate system, expressed in units set by the **UnitsLength** property.

Syntax

`control.WorldX [=Value As Single]`

Type

Single

Range

Unbounded.

Remark(s)

When **WorldX** and **WorldY** values are modified, the **ImageX** and **ImageY** values are automatically recalculated.

Example

```
' Build a Point with Class Library
Dim MyPoint as HSPoint
Set MyPoint = MyDatabase.AddPoint ("View", "Point")
With MyPoint
    .Calibration = MyImage.Calibration
    . Calibrated = True
    .WorldX = 25
    .WorldY = 15
End With
```

WorldY**Description**

Y coordinate of the Point entity, in the World coordinate system, expressed in units set by the **UnitsLength** property.

Syntax

*control.***WorldY** [=Value As Single]

Type

Single

Range

Unbounded.

Remark(s)

When **WorldX** and **WorldY** values are modified, the **ImageX** and **ImageY** values are automatically recalculated.

Example

```
' Build a Point with Class Library
Dim MyPoint as HSPoint
Set MyPoint = MyDatabase.AddPoint ("View", "Point")
With MyPoint
    .Calibration = MyImage.Calibration
    .WorldX = 25
    .WorldY = 15
End With
```

Methods

AboutBox

Description
Displays the HSPoint About box.

Syntax
`control.AboutBox`

Parameter(s)
None

Return
No return value.

Example
`HSPoint1.AboutBox`

Angle

Description
Computes the angle of a line that joins the current HSPoint object to another point object called the *SourcePoint* object. The value of the **Calibrated** property determines whether the angle is defined in the non-calibrated Image coordinate system or the calibrated World coordinate system. Expressed in units set by the **UnitsOrientation** property.

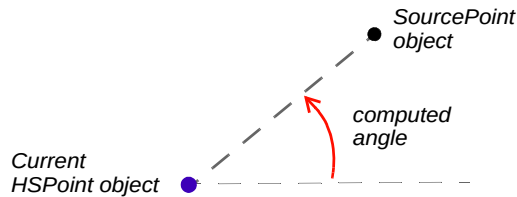


FIGURE 6 *HSPoint — Angle Method*

Syntax
`control.Angle (SourcePoint As Object) As Single`

Parameter(s)

Name	Description	Type	Range
<i>SourcePoint</i>	Reference HSPoint object used to compute angle.	HSPoint	Not applicable.

Return
Single

Example
`Dim Angle as Single`
`Angle = Point0.Angle(Point1)`

CopyFormatFrom

Description
Copies the format of a specified object to the current object.

Syntax

control.CopyFormatFrom(*SourceObject* As Object)

Parameter(s)

Name	Description	Type	Range
<i>SourceObject</i>	Object from which the format is copied.	HSScene, HSImage, HSArc, HSLine, or HSPoint.	Not applicable.

Return

No return value.

Example

Point1.CopyFormatFrom(Point0)

Distance

Description

Computes the distance between the current HSPoint object and another point object called the *SourcePoint* object. The value of the **Calibrated** property determines whether the angle is defined in the non-calibrated Image coordinate system or the calibrated World coordinate system. Expressed in units set by the **UnitsLength** property.

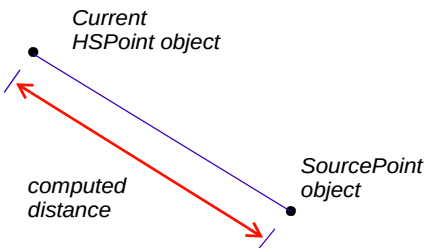


FIGURE 7 *HSPoint — Angle Method*

Syntax

control.Distance (*SourcePoint* As Object) As Single

Parameter(s)

Name	Description	Type	Range
<i>SourcePoint</i>	Reference HSPoint object used to compute distance.	HSPoint	Not applicable.

Return

Single

Example

Dim Distance as Single
Distance = Point0.Distance(Point1)

GetDispatchPointer

Description

Reserved for internal use.

Load

Description

Loads the content of a point file (*.hpt)

Syntax

control.**Load**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the hpt file to load. Optional.	Variant	Not applicable

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file open browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSPoint1.Load("Point0.hpt")
Succeed = HSPoint2.Load( )
```

Reset

Description

Reserved for internal use.

Save

Description

Saves the Point entity to a HexSight Point file (*.hpt).

Syntax

control.**Save**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the point file (*.hpt) to which the point is saved. Optional.	Variant	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file save browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean  
Succeed = HSPoint1.Save("Point0.hpt")  
Succeed = HSPoint2.Save( )
```

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

HSScene

Description

An HSScene control contains a vectorized description of an object's contours, as opposed to an Image, which is made up of pixels.

Properties

AveragePixelHeight

Description

Average calibrated height of the Scene pixels, expressed in calibrated units specified by the **UnitsLength** property.

Syntax

control.AveragePixelHeight [= Value As Single]

Type

Single

Range

Greater than or equal to zero.

Remark(s)

If the Scene is not calibrated then the **AveragePixelHeight** will be equal to zero.

Example

Dim Value As Single

Value = HSScene1.AveragePixelHeight

AveragePixelWidth

Description

Average calibrated width of the Scene pixels, expressed in calibrated units specified by the **UnitsLength** property.

Syntax

control.AveragePixelWidth [= Value As Single]

Type

Single

Range

Greater than or equal to zero.

Remark(s)

If the Scene is not calibrated then the **AveragePixelHeight** will be equal to zero.

Example

Dim Value As Single

Value = HSScene1.AveragePixelWidth

BoundaryBottom

Description

Minimum Y coordinate value of the Scene with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryBottom** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
Dim Value As Single
Value = HSScene1.BoundaryBottom
```

BoundaryLeft

Description

Minimum X coordinate value of the Scene with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryLeft** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
Dim Value As Single
Value = HSScene1.BoundaryLeft
```

BoundaryRight

Description

Maximum X coordinate value of the Scene with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryRight** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
Dim Value As Single
Value = HSScene1.BoundaryRight
```

BoundaryTop

Description

Maximum Y coordinate value of the Scene with respect to the World or the Image coordinate system.

Syntax

control.**BoundaryTop** [= *Value* As Single]

Type

Single

Range

Any Single value.

Example

```
Dim Value As Single
Value = HSScene1.BoundaryTop
```

Calibration

Description

Returns the internal **HSCalibration** object of the control. It is used to access or modify the calibration information.

Syntax

control.**Calibration** [= *Value* As HSCalibration]

Type

HSCalibration

Range

Not applicable.

Example

```
Dim Calibration As HSCalibration
Calibration = InstanceScene.Calibration
```

Description

Description

String that describes the standard usage of the HSScene control.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = HSScene1.Description
```

EdgeCount

Description

Number of edgels in the Scene. Read only.

Syntax*control*.**EdgeCount****Type**

Long

Range

Greater than or equal to 0.

Example

```
Dim Count As Long
Count = Scene1.EdgeCount
```

EdgeImageX

Description

The X image coordinate of an edge element in the scene. Because of its overhead, this property should only be used to access few coordinates in applications where speed is not critical. To access a large number of coordinates, **EdgeImageXArray** should be used instead. The **EdgeImageXArray** property is supported only for C++ applications however. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements. Read only.

Syntax*control*.**EdgeImageX**(*EdgeIndex* As Long)**Type**

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgeIndex</i>	Index of the edge element.	Long	Minimum: 0 Maximum: Last edge element of the last edge path

Example

```
Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path
```

```

For j = Scene.EdgePathFirstEdgeIndex(i) To _
    Scene.EdgePathLastEdgeIndex(i) - 1

    X0 = Scene.EdgeImageX(j)
    Y0 = Scene.EdgeImageY(j)
    X1 = Scene.EdgeImageX(j+1)
    Y1 = Scene.EdgeImageY(j+1)

    ' Draw a line between (X0,Y0) and (X1,Y1)
    ' using a dark color

    DrawLine( X0, Y0, X1, Y1, Blue)

Next j

Next i

```

EdgeImageXArray

Description

A pointer to the array containing the X image coordinates of all the edge elements in the scene. This property is supported only for C++ applications. For Visual Basic applications, **EdgeImageX** should be used instead. Index 0 of this array contains the X coordinate of the first edge element in the scene, while the X coordinate of the last edge element is stored at the index corresponding to the last edge element of the last edge path in the scene. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements.

Syntax

control.GetEdgeImageXArray()

Type

OLE_HANDLE (Can be cast to a float pointer in C++ applications)

Range

Not applicable.

Example

```

float * EdgeIX;
float * EdgeIY;
int FirstEdgeI;
int LastEdgeI;
int i;
int j;
float X0;
float X1;

```

```

float Y0;
float Y1;

...

// Get the edgel arrays

EdgelX = (float *) Scene.GetEdgelImageXArray( );
EdgelY = (float *) Scene.GetEdgelImageYArray( );

// Scan the edge paths of the scene to display all the
// contours

for ( i = 0; i < Scene.GetEdgePathCount( ); i++ )
{

    // Scan the edge elements of the current edge path

    FirstEdgel = Scene.GetEdgePathFirstEdgelIndex( i );
    LastEdgel = Scene.GetEdgePathLastEdgelIndex( i );

    for ( j = FirstEdgel; j < LastEdgel; j++ )
    {
        X0 = EdgelX[j];
        Y0 = EdgelY[j];
        X1 = EdgelX[j+1];
        Y1 = EdgelY[j+1];

        // Draw a line between (X0,Y0) and (X1,Y1)
        // using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue )
    }
}

```

EdgelImageY

Description

The Y image coordinate of an edge element in the scene. Because of its overhead, this property should only be used to access few coordinates in applications where speed is not critical. To access a large number of coordinates, **EdgelImageYArray** should be used instead.

The **EdgeImageYArray** property is supported only for C++ applications however. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements. Read only.

Syntax

control.**EdgeImageY**(*EdgeIndex* As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgeIndex</i>	Index of the edge element.	Long	Minimum: 0 Maximum: Last edge element of the last edge path

Example

```

Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeImageX(j)
        Y0 = Scene.EdgeImageY(j)
        X1 = Scene.EdgeImageX(j+1)
        Y1 = Scene.EdgeImageY(j+1)

        ' Draw a line between (X0,Y0) and (X1,Y1)
        ' using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue)

```

Next j

Next i

EdgeImageYArray

Description

A pointer to the array containing the Y image coordinates of all the edge elements in the scene. This property is supported only for C++ applications. For Visual Basic applications, **EdgeImageY** should be used instead. Index 0 of this array contains the Y coordinate of the first edge element in the scene, while the Y coordinate of the last edge element is stored at the index corresponding to the last edge element of the last edge path in the scene. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements.

Syntax

control.GetEdgeImageYArray()

Type

OLE_HANDLE (Can be cast to a float pointer in C++ applications)

Range

Not applicable.

Example

```
float * EdgelX;
float * EdgelY;
int FirstEdgel;
int LastEdgel;
int i;
int j;
float X0;
float X1;
float Y0;
float Y1;

...

// Get the edgel arrays

EdgelX = (float *) Scene.GetEdgeImageXArray( );
EdgelY = (float *) Scene.GetEdgeImageYArray( );

// Scan the edge paths of the scene to display all the
// contours

for ( i = 0; i < Scene.GetEdgePathCount( ); i++ )
```



```

{

    // Scan the edge elements of the current edge path

    FirstEdgel = Scene.GetEdgePathFirstEdgelIndex( i );
    LastEdgel = Scene.GetEdgePathLastEdgelIndex( i );

    for ( j = FirstEdgel; j < LastEdgel; j++ )
    {
        X0 = EdgelX[j];
        Y0 = EdgelY[j];
        X1 = EdgelX[j+1];
        Y1 = EdgelY[j+1];

        // Draw a line between (X0,Y0) and (X1,Y1)
        // using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue )
    }
}

```

EdgelWorldX

Description

The calibrated X World coordinate of an edge element in the scene. Because of its overhead, this property should only be used to access few coordinates in applications where speed is not critical. To access a large number of coordinates, **EdgelWorldXArray** should be used instead. The **EdgelWorldXArray** property is supported only for C++ applications however. See also the **EdgePathFirstEdgelIndex** and **EdgePathLastEdgelIndex** properties for more details on accessing edge elements. Read only.

Syntax

control.**EdgelWorldX**(*EdgelIndex* As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgelIndex</i>	Index of the edge element.	Long	Minimum: 0 Maximum: Last edge element of the last edge path

Example

```

Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours in calibrated units

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeWorldX(j)
        Y0 = Scene.EdgeWorldY(j)
        X1 = Scene.EdgeWorldX(j+1)
        Y1 = Scene.EdgeWorldY(j+1)

        ' Draw a line between (X0,Y0) and (X1,Y1)
        ' using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue)

    Next j

Next i

```

EdgeWorldXArray**Description**

A pointer to the array containing the calibrated X World coordinates of all the edge elements in the scene. This property is supported only for C++ applications. For Visual Basic applications, **EdgeWorldX** should be used instead. Index 0 of this array contains the X coordinate of the first edge element in the scene, while the X coordinate of the last edge element is stored at the index corresponding to the last edge element of the last edge path in the scene. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements.

Syntax

control.**GetEdgelWorldXArray**()

Type

OLE_HANDLE (Can be cast to a float pointer in C++ applications)

Range

Not applicable.

Example

```
float * EdgelX;
float * EdgelY;
int FirstEdgel;
int LastEdgel;
int i;
int j;
float X0;
float X1;
float Y0;
float Y1;

...

// Get the edgel arrays

EdgelX = (float *) Scene.GetEdgelWorldXArray( );
EdgelY = (float *) Scene.GetEdgelWorldYArray( );

// Scan the edge paths of the scene to display all the
// contours in calibrated units

for ( i = 0; i < Scene.GetEdgePathCount( ); i++ )
{

    // Scan the edge elements of the current edge path

    FirstEdgel = Scene.GetEdgePathFirstEdgelIndex( i );
    LastEdgel = Scene.GetEdgePathLastEdgelIndex( i );

    for ( j = FirstEdgel; j < LastEdgel; j++ )
    {
        X0 = EdgelX[j];
        Y0 = EdgelY[j];
        X1 = EdgelX[j+1];
        Y1 = EdgelY[j+1];
    }
}
```

```

        // Draw a line between (X0,Y0) and (X1,Y1)
        // using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue )
    }
}

```

EdgeWorldY

Description

The calibrated Y World coordinate of an edge element in the scene. Because of its overhead, this property should only be used to access few coordinates in applications where speed is not critical. To access a large number of coordinates, **EdgeWorldYArray** should be used instead. The **EdgeWorldYArray** property is supported only for C++ applications however. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements. Read only.

Syntax

control.**EdgeWorldY**(*EdgeIndex* As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgeIndex</i>	Index of the edge element.	Long	Minimum: 0 Maximum: Last edge element of the last edge path

Example

```

Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours in calibrated units

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

```

```

For j = Scene.EdgePathFirstEdgeIndex(i) To _
    Scene.EdgePathLastEdgeIndex(i) - 1

    X0 = Scene.EdgeWorldX(j)
    Y0 = Scene.EdgeWorldY(j)
    X1 = Scene.EdgeWorldX(j+1)
    Y1 = Scene.EdgeWorldY(j+1)

    ' Draw a line between (X0,Y0) and (X1,Y1)
    ' using a dark color

    DrawLine( X0, Y0, X1, Y1, Blue)

Next j

Next i

```

EdgeWorldYArray

Description

A pointer to the array containing the calibrated Y World coordinates of all the edge elements in the scene. This property is supported only for C++ applications. For Visual Basic applications, **EdgeWorldY** should be used instead. Index 0 of this array contains the Y coordinate of the first edge element in the scene, while the Y coordinate of the last edge element is stored at the index corresponding to the last edge element of the last edge path in the scene. See also the **EdgePathFirstEdgeIndex** and **EdgePathLastEdgeIndex** properties for more details on accessing edge elements.

Syntax

control.GetEdgeWorldXArray()

Type

OLE_HANDLE (Can be cast to a float pointer in C++ applications)

Range

Not applicable.

Example

```

float * EdgeX;
float * EdgeY;
int FirstEdge;
int LastEdge;
int i;
int j;
float X0;
float X1;

```

```

float Y0;
float Y1;

...

// Get the edgel arrays

EdgelX = (float *) Scene.GetEdgelWorldXArray( );
EdgelX = (float *) Scene.GetEdgelWorldYArray( );

// Scan the edge paths of the scene to display all the
// contours in calibrated units

for ( i = 0; i < Scene.GetEdgePathCount( ); i++ )
{

    // Scan the edge elements of the current edge path

    FirstEdgel = Scene.GetEdgePathFirstEdgelIndex( i );
    LastEdgel = Scene.GetEdgePathLastEdgelIndex( i );

    for ( j = FirstEdgel; j < LastEdgel; j++ )
    {
        X0 = EdgelX[j];
        Y0 = EdgelY[j];
        X1 = EdgelX[j+1];
        Y1 = EdgelY[j+1];

        // Draw a line between (X0,Y0) and (X1,Y1)
        // using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue )
    }
}

```

EdgePathArea

Description

Area of a given edge path in the scene, expressed in calibrated units. Read only. An edge path is a sequence of connected edge elements in a scene. After a new scene is created, the

ComputeProperties method must be called to compute the properties of its edge paths before using the **EdgePathArea** property.

Syntax

control.**EdgePathArea** (*EdgePathIndex* As Long)

Type

Single

Range

Any positive Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgePathIndex</i>	Index of the edge path.	Long	Minimum: 0 Maximum: EdgePathCount -1

Example

```
Dim Area As Single
Dim i As Long
Scene.ComputeProperties
For i = 0 To Scene.EdgePathCount - 1
    Area = Scene.EdgePathArea(i)
Next i
```

EdgePathCentroidX

Description

X coordinate of the centroid of a given edge path in the scene, expressed in calibrated units. Read only. An edge path is a sequence of connected edge elements in a scene. After a new scene is created, the **ComputeProperties** method must be called to compute the properties of its edge paths before using the **EdgePathCentroidX** property.

Syntax

control.**EdgePathCentroidX**(*EdgePathIndex*As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgePathIndex</i>	Index of the edge path.	Long	Minimum: 0 Maximum: EdgePathCount -1

Example

```
Dim CenterX As Single
Dim CenterY As Single
Dim i As Long
Scene.ComputeProperties
```

```

For i = 0 To Scene.EdgePathCount - 1
    CenterX = Scene.EdgeCentroidX(i)
    CenterY = Scene.EdgeCentroidY(i)
Next i

```

EdgePathCentroidY

Description

Y coordinate of the centroid of a given edge path in the scene, expressed in calibrated units. Read only. An edge path is a sequence of connected edge elements in a scene. After a new scene is created, the **ComputeProperties** method must be called to compute the properties of its edge paths before using the **EdgePathCentroidY** property.

Syntax

control.**EdgePathCentroidY**(*EdgePathIndex* As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgePathIndex</i>	Index of the edge path.	Long	Minimum: 0 Maximum: EdgePathCount -1

Example

```

Dim CenterX As Single
Dim CenterY As Single
Dim i As Long
Scene.ComputeProperties
For i = 0 To Scene.EdgePathCount - 1
    CenterX = Scene.EdgeCentroidX(i)
    CenterY = Scene.EdgeCentroidY(i)
Next i

```

EdgePathClosed

Description

When True, indicates that a given edge path in the scene is closed. Read only. An edge path is a sequence of connected edge elements in a scene. An edge path is closed when the coordinates of its first and last edge elements are identical. After a new scene is created, the **ComputeProperties** method must be called to compute the properties of its edge paths before using the **EdgePathClosed** property.

Syntax

control.**EdgePathClosed**(*EdgePathIndex* As Long) [= *Value* As Boolean]

Type

Boolean

Range

True, False

Example

```

Dim i As Long
For i = 0 To Scene.EdgePathCount - 1

    ' Process the edge path only if is closed

    If Scene.EdgePathClosed(i) Then

        ' Process edge path

    End If

Next i

```

EdgePathCount

Description

Number of edge paths in the scene. Read only. An edge path is a sequence of connected edge elements in a scene. The edge elements forming an edge path are stored sequentially in the scene's edge elements array, starting at index **EdgePathFirstEdgeIndex(i)** and ending at index **EdgePathLastEdgeIndex(i)**, where i is the index of the edge path.

Syntax

control.**EdgePathCount**

Type

Long

Range

Greater than or equal to 0.

Example

```

Dim i As Long
Dim j As Long

' Scan the edge paths of the scene

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathLastEdgeIndex(i) - 1

        ' Process edge element at index j
    
```

Next j

Next i

EdgePathFirstEdgeIndex

Description

Index of the first edge element of a given edge path. Read only. An edge path is a sequence of connected edge elements in a scene. The edge elements forming an edge path are stored sequentially in the scene's edge elements array, starting at index

EdgePathFirstEdgeIndex(i) and ending at index **EdgePathLastEdgeIndex(i)**, where i is the index of the edge path.

Syntax

control.**EdgePathFirstEdgeIndex**(*EdgePathIndex* As Long)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>EdgePathIndex</i>	Index of the edge path.	Long	Minimum: 0 Maximum: EdgePathCount -1

Example

```
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathLastEdgeIndex(i) - 1

        ' Process edge element at index j

    Next j

Next i
```

EdgePathLastEdgeIndex

Description

Index of the last edge element of a given edge path. Read only. An edge path is a sequence of connected edge elements in a scene. The edge elements forming an edge path are stored sequentially in the scene's edge elements array, starting at index

EdgePathFirstEdgeIndex(i) and ending at index **EdgePathLastEdgeIndex(i)**, where i is the index of the edge path.

Syntax

control.**EdgePathLastEdgeIndex**(*EdgePathIndex* As Long)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>EdgePathIndex</i>	Index of the edge path.	Long	Minimum: 0 Maximum: EdgePathCount -1

Example

```
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathLastEdgeIndex(i) - 1

        ' Process edge element at index j

    Next j

Next i
```

EdgePathPerimeter

Description

Perimeter of a given edge path in the scene, expressed in calibrated units. Read only. An edge path is a sequence of connected edge elements in a scene. After a new scene is created, the

ComputeProperties method must be called to compute the properties of its edge paths before using the **EdgePathPerimeter** property.

Syntax

control.**EdgePathPerimeter**(*EdgePathIndex* As Long)

Type

Single

Range

Any positive Single value.

Parameter(s)

Name	Description	Type	Range
<i>EdgePathIndex</i>	Index of the edge path.	Long	Minimum: 0 Maximum: EdgePathCount -1

Example

```
Dim Perimeter As Single
Dim i As Long
Scene.ComputeProperties
For i = 0 To Scene.EdgePathCount - 1
    Perimeter = Scene.EdgePathPerimeter(i)
Next i
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

MsgBox "Error: " & Locator.**ErrorObject**.Description

Height

Description

Height of the Scene expressed in a unit length specified by the **UnitsLength** property.

Syntax

control.**Height** [= *Value* As Single]

Type

Single

Range

Greater than or equal to zero.

Example

```
Dim Value As Single
Value = HSScene1.Height
```

ImageCoordinateSystemType

Description

Selects whether the coordinate system used to express pixel coordinates in the scene is right-handed or left-handed. In a right-handed Image coordinate system (default), the origin is located at the bottom left corner of the scene and the Y-axis is directed upward. In a left-handed Image coordinate system, the origin is located at the top left corner of the scene and the Y-axis is directed downward.

Syntax

control.**ImageCoordinateSystemType** [= *Value* As hsCoordinateSystemType]

Type

hsCoordinateSystemType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRightHanded	0	The Image coordinate is right-handed (origin located at the bottom left corner and Y-axis directed upward).
hsLeftHanded	1	The Image coordinate is left-handed (origin located at the top left corner and Y-axis directed downward).

Remark(s)

A left-handed scene as well as a right-handed scene can be calibrated into either a left-handed or a right-handed World coordinate system. When the **ImageCoordinateSystemType** property of a scene is changed, the scene must be recalibrated to avoid any inconsistencies.

Example

```
Dim CurrentType As hsCoordinateSystemType
CurrentType = HSScene1.ImageCoordinateSystemType
```

Information

Description

String that provides information pertaining to the object. For example: Name, Boundary coordinates, Image Coordinate System, Timestamp, etc.

Syntax

control.**Information** [= *Value* As String]

Type

String

Range

Not applicable

Example

```
Dim Info As String
```

Info = HSScene1.**Information**

InstanceCount

Description

Number of instances in the scene. Read only. An instance is a group of edge paths that belong to a single object instance in the scene. The edge paths forming an instance are stored sequentially in the scene's edge path array, starting at index

InstanceFirstEdgePathIndex(i) and ending at index **InstanceLastEdgePathIndex(i)**, where i is the index of the instance.

Syntax

control.**InstanceCount**

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim Instance As Long
Dim First As Long
Dim Last As Long
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeImageX(j)
        Y0 = Scene.EdgeImageY(j)
        X1 = Scene.EdgeImageX(j+1)
        Y1 = Scene.EdgeImageY(j+1)

        ' Draw a line between (X0,Y0) and (X1,Y1)
```

```

        ' using a dark color

        DrawLine( X0, Y0, X1, Y1, Blue)

    Next j

Next i

' Highlight the contours belonging to an instance selected
' in the user interface

Instance = Combo1.ListIndex

If Instance < 0 Or Instance > Scene.InstanceCount Then
    Exit Sub
End If

First = Scene.InstanceFirstEdgePathIndex(Instance)
Last = Scene.InstanceLastEdgePathIndex(Instance)

For i = First To Last

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeImageX(j)
        Y0 = Scene.EdgeImageY(j)
        X1 = Scene.EdgeImageX(j+1)
        Y1 = Scene.EdgeImageY(j+1)

        ' Draw a line between (X0,Y0) and (X1,Y1)
        ' using a bright color

        DrawLine( X0, Y0, X1, Y1, Red)

    Next j

```

Next `i`

InstanceFirstEdgePathIndex

Description

Index of the first edge path of a given instance. Read only. An instance is a group of edge paths that belong to a single object instance in the scene. The edge paths forming an instance are stored sequentially in the scene's edge path array, starting at index **InstanceFirstEdgePathIndex(i)** and ending at index **InstanceLastEdgePathIndex(i)**, where `i` is the index of the instance.

Syntax

`control.InstanceFirstEdgePathIndex(InstanceIndex As Long)`

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Long	Minimum: 0 Maximum: InstanceCount -1

Example

```
Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim Instance As Long
Dim First As Long
Dim Last As Long
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeImageX(j)
```



```

    Y0 = Scene.EdgeImageY(j)
    X1 = Scene.EdgeImageX(j+1)
    Y1 = Scene.EdgeImageY(j+1)

    ' Draw a line between (X0,Y0) and (X1,Y1)
    ' using a dark color

    DrawLine( X0, Y0, X1, Y1, Blue)

Next j

Next i

' Highlight the contours belonging to an instance selected
' in the user interface

Instance = Combo1.ListIndex

If Instance < 0 Or Instance > Scene.InstanceCount Then
    Exit Sub
End If

First = Scene.InstanceFirstEdgePathIndex(Instance)
Last = Scene.InstanceLastEdgePathIndex(Instance)

For i = First To Last

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeImageX(j)
        Y0 = Scene.EdgeImageY(j)
        X1 = Scene.EdgeImageX(j+1)
        Y1 = Scene.EdgeImageY(j+1)

        ' Draw a line between (X0,Y0) and (X1,Y1)
        ' using a bright color

```

```
DrawLine( X0, Y0, X1, Y1, Red)
```

```
Next j
```

```
Next i
```

InstanceLastEdgePathIndex

Description

Index of the first edge path of a given instance. Read only. An instance is a group of edge paths that belong to a single object instance in the scene. The edge paths forming an instance are stored sequentially in the scene's edge path array, starting at index **InstanceFirstEdgePathIndex(i)** and ending at index **InstanceLastEdgePathIndex(i)**, where i is the index of the instance.

Syntax

control.InstanceLastEdgePathIndex(InstanceIndex As Long)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Long	Minimum: 0 Maximum: InstanceCount -1

Example

```
Dim X0 As Single
Dim Y0 As Single
Dim X1 As Single
Dim Y1 As Single
Dim Instance As Long
Dim First As Long
Dim Last As Long
Dim i As Long
Dim j As Long

' Scan the edge paths of the scene to display all the
' contours

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path
```

```

For j = Scene.EdgePathFirstEdgeIndex(i) To _
    Scene.EdgePathFLastEdgeIndex(i) - 1

    X0 = Scene.EdgeImageX(j)
    Y0 = Scene.EdgeImageY(j)
    X1 = Scene.EdgeImageX(j+1)
    Y1 = Scene.EdgeImageY(j+1)

    ' Draw a line between (X0,Y0) and (X1,Y1)
    ' using a dark color

    DrawLine( X0, Y0, X1, Y1, Blue)

Next j

Next i

' Highlight the contours belonging to an instance selected
' in the user interface

Instance = Combo1.ListIndex

If Instance < 0 Or Instance > Scene.InstanceCount Then
    Exit Sub
End If

First = Scene.InstanceFirstEdgePathIndex(Instance)
Last = Scene.InstanceLastEdgePathIndex(Instance)

For i = First To Last

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathFLastEdgeIndex(i) - 1

        X0 = Scene.EdgeImageX(j)
        Y0 = Scene.EdgeImageY(j)

```

```

X1 = Scene.EdgeImageX(j+1)
Y1 = Scene.EdgeImageY(j+1)

' Draw a line between (X0,Y0) and (X1,Y1)
' using a bright color

DrawLine( X0, Y0, X1, Y1, Red)

```

```
Next j
```

```
Next i
```

ModelEdgePathArea

Description

Area of a given Model edge path in the Model contour scene, expressed in calibrated units. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image.

Syntax

control.**ModelEdgePathArea** (*ModelEdgePathIndex* As Long)

Type

Single

Range

Any positive Single value.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

```

Dim Area As Single
Dim i As Long
Area = 0
For i = 0 To Scene.ModelEdgePathCount - 1
    Area = Area + Scene.ModelEdgePathArea(i)
Next i

```

ModelEdgePathCentroidX

Description

X coordinate of the centroid of a given Model edge path in the Model contour scene, expressed in calibrated units. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image.

Syntax

*control.***ModelEdgePathCentroidX**(*ModelEdgePathIndex* As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

```
Dim CenterX As Single
Dim CenterY As Single
Dim i As Long
For i = 0 To Scene.EdgePathCount - 1
    CenterX = Scene.ModelEdgePathCentroidX(i)
    CenterY = Scene.ModelEdgePathCentroidY(i)
    ...
Next i
```

ModelEdgePathCentroidY

Description

Y coordinate of the centroid of a given Model edge path in the Model contour scene, expressed in calibrated units. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image.

Syntax

*control.***ModelEdgePathCentroidY**(*ModelEdgePathIndex* As Long)

Type

Single

Range

Any Single value.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

```
Dim CenterX As Single
Dim CenterY As Single
Dim i As Long
For i = 0 To Scene.ModelEdgePathCount - 1
```

```

CenterX = Scene.ModelEdgePathCentroidX(i)
CenterY = Scene.ModelEdgePathCentroidY(i)
...
Next i

```

ModelEdgePathCount

Description

Number of Model edge paths in the Model contour scene. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image. The model edge path selection structure is used to implement interactive Model edition functionalities. As opposed to a regular edge path, the edge elements that form a model edge path are not necessarily stored sequentially in the scene's edge elements array. A given model edge path *i* always starts at index **ModelEdgePathFirstEdgeIndex**(*i*) and ends at index **ModelEdgePathLastEdgeIndex**(*i*), but its edge elements are not always stored sequentially between these two indexes in the edge element array. To scan the edge elements of a model edge path, the **ModelEdgePathNextEdgeIndex** and **ModelEdgePathPreviousEdgeIndex** iterators must be used.

Syntax

control.**ModelEdgePathCount**

Type

Long

Range

Greater than or equal to 0.

Example

See example in **ModelEdgePathFirstEdgeIndex**.

ModelEdgePathFirstEdgeIndex

Description

Index of the first edge element of a given model edge path in a Model contour scene. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image. The model edge path selection structure is used to implement interactive Model edition functionalities. As opposed to a regular edge path, the edge elements that form a model edge path are not necessarily stored sequentially in the scene's edge elements array. A given model edge path *i* always starts at index **ModelEdgePathFirstEdgeIndex**(*i*) and ends at index **ModelEdgePathLastEdgeIndex**(*i*), but its edge elements are not always stored sequentially between these two indexes in the edge element array. To scan the edge elements of a model edge path, the **ModelEdgePathNextEdgeIndex** and **ModelEdgePathPreviousEdgeIndex** iterators must be used.

Syntax

control.**ModelEdgePathFirstEdgeIndex**(*ModelEdgePathIndex* As Long)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

```

Dim i As Long
Dim j As Long

' Scan the edge paths of the Model contour scene
' to draw all the detected contours in blue

For i = 0 To Scene.EdgePathCount - 1

    ' Scan the edge elements of the current edge path

    For j = Scene.EdgePathFirstEdgeIndex(i) To _
        Scene.EdgePathLastEdgeIndex(i) - 1

        ' Draw the edge element of the contour in blue

    Next j

Next i

' Scan the model edge paths of the Model contour scene
' to draw the selected features in green

For i = 0 To Scene.ModelEdgePathCount - 1

    ' Scan the edge elements of the current model edge path
    ' The edge elements are scanned from last to first

    j = Scene.ModelEdgePathFirstEdgeIndex(i)

    Do While j <> -1

        ' Draw the edge element of the feature in green

        j = Scene.ModelEdgePathPreviousEdgeIndex(i)

```

Loop

Next *i*

ModelEdgePathLastEdgeIndex

Description

Index of the last edge element of a given model edge path in a Model contour scene. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image. The model edge path selection structure is used to implement interactive Model edition functionalities. As opposed to a regular edge path, the edge elements that form a model edge path are not necessarily stored sequentially in the scene's edge elements array. A given model edge path *i* always starts at index **ModelEdgePathFirstEdgeIndex**(*i*) and ends at index **ModelEdgePathLastEdgeIndex**(*i*), but its edge elements are not always stored sequentially between these two indexes in the edge element array. To scan the edge elements of a model edge path, the **ModelEdgePathNextEdgeIndex** and **ModelEdgePathPreviousEdgeIndex** iterators must be used.

Syntax

control.**ModelEdgePathLastEdgeIndex**(*ModelEdgePathIndex* As Long)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

See example in **ModelEdgePathFirstEdgeIndex**.

ModelEdgePathNextEdgeIndex

Description

Moves the internal iterator to the next edge element of a given model edge path and returns its index. Read only. Returns -1 after the last edge element of the model edge path is reached. This internal iterator is used to scan the edge elements of a model edge path. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image. The model edge path selection structure is used to implement interactive Model edition functionalities. As opposed to a regular edge path, the edge elements that form a model edge path are not necessarily stored sequentially in the scene's edge elements array. A given model edge path *i* always starts at index **ModelEdgePathFirstEdgeIndex**(*i*) and ends at index **ModelEdgePathLastEdgeIndex**(*i*), but its edge elements are not always stored sequentially between these two indexes in the edge element array. To scan the edge elements of a model

edge path, the **ModelEdgePathNextEdgeIndex** and **ModelEdgePathPreviousEdgeIndex** iterators must be used.

Syntax

*control.***ModelEdgePathNextEdgeIndex**(*ModelEdgePathIndex* As Long)

Type

Long

Range

Greater than or equal to -1.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

See example in **ModelEdgePathFirstEdgeIndex**.

ModelEdgePathPerimeter

Description

Perimeter of a given Model edge path in the Model contour scene, expressed in calibrated units. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image.

Syntax

*control.***ModelEdgePathPerimeter**(*ModelEdgePathIndex* As Long)

Type

Single

Range

Any positive Single value.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

```
Dim Perimeter As Single
Dim i As Long
Perimeter = 0
For i = 0 To Scene.ModelEdgePathCount - 1
    Perimeter = Perimeter + Scene.ModelEdgePathPerimeter(i)
Next i
```

ModelEdgePathPreviousEdgeIndex

Description

Moves the internal iterator to the previous edge element of a given model edge path and returns its index. Read only. Returns -1 after the last edge element of the model edge path is

reached. This internal iterator is used to scan the edge elements of a model edge path. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image. The model edge path selection structure is used to implement interactive Model edition functionalities. As opposed to a regular edge path, the edge elements that form a model edge path are not necessarily stored sequentially in the scene's edge elements array. A given model edge path *i* always starts at index **ModelEdgePathFirstEdgeIndex**(*i*) and ends at index **ModelEdgePathLastEdgeIndex**(*i*), but its edge elements are not always stored sequentially between these two indexes in the edge element array. To scan the edge elements of a model edge path, the **ModelEdgePathNextEdgeIndex** and **ModelEdgePathPreviousEdgeIndex** iterators must be used.

Syntax

control.**ModelEdgePathPreviousEdgeIndex**(*ModelEdgePathIndex* As Long)

Type

Long

Range

Greater than or equal to -1.

Parameter(s)

Name	Description	Type	Range
<i>ModelEdgePathIndex</i>	Index of the model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount -1

Example

See example in **ModelEdgePathFirstEdgeIndex**.

ModifyCount

Description

Number of modifications that have been made to the Scene. Read only.

Syntax

control.**ModifyCount** [= *Value* As Long]

Type

Long

Range

Greater than or equal to zero.

Example

```
Dim Count As Integer
Count = HSScene1.ModifyCount
```

Name

Description

Name used to access the object.

Syntax

control.**Name**[= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim Name As String
Name = HSScene1.Name
HSScene2.Name = "Scene_00"
```

OriginX

Description

Horizontal (X) origin of the Image from which the Scene was built, expressed in pixel units. Along with **OriginY**, this property can be used to specify the origin of an image cropped externally to HexSight.

Syntax

```
control.OriginX [= Value As Long]
```

Type

Long

Range

Greater than or equal to zero.

Remarks

Along with **OriginY**, this property can be used to specify the origin of an image cropped externally to HexSight.

Example

```
Dim X0 As Long
Dim Y0 As Long
' Use the origin of the cropped image from which the scene
' was built to offset the contours correctly in a Picture
' Box
X0 = HSScene1.OriginX
Y0 = HSScene1.OriginY
```

OriginY

Description

Vertical (Y) origin of the Image from which the Scene was built, expressed in pixel units. Along with **OriginX**, this property can be used to specify the origin of an image cropped externally to HexSight.

Syntax

```
control.OriginY [= Value As Long]
```

Type

Long

Range

Greater than or equal to zero.

Example

```
Dim X0 As Long
```

```

Dim Y0 As Long
' Use the origin of the cropped image from which the scene
' was built to offset the contours correctly in a Picture
' Box
X0 = HSScene1.OriginX
Y0 = HSScene1.OriginY

```

Type

Description

Type of the Scene object. Read only.

Syntax

control.**Type**

Type

hsSceneType

Range

The valid range for this property is as follows:

Constant	Value	Description
hsSceneReserved	0	Reserved internal type.
hsSceneStandard	4	Standard contour scene generated by the edge detection operation.
hsSceneModel	5	Model scene created by the Locator tool. Contains only model-specific information.
hsSceneModelSource	6	Model scene that is used by the Locator tool to create new models.
hsSceneInstances	7	Instance scene that is generated by the Locator tool. Includes found-instance information for the model-based mode.

Example

```

Dim SceneType As hsSceneType
ObjectType = Scene1.Type

```

TypeName

Description

ActiveX type name for the object. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSScene"

Example

```

Dim ObjectType As String
ObjectType = HSScene1.TypeName

```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = HSScene1.UnitsLength
HSScene1.UnitsLength = hsInch
```

Width

Description

Width of the Scene expressed in pixels.

Syntax

control.**Width** [= *Value* As Single]

Type

Single

Range

Greater than or equal to zero.

Example

```
Dim Value As Single
Value = HSScene1.Width
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = HSScene1.Version
```

Methods

AboutBox

Description

Displays the HSScene About box.

Syntax

```
control.AboutBox( )
```

Parameter(s)

None

Return

No return value.

Example

```
HSScene1.AboutBox
```

ComputeProperties

Description

Computes the properties of the scene's edge paths. This method must be called after a new scene is created before accessing any of the following properties: **EdgePathArea**, **EdgePathCentroidX**, **EdgePathCentroidY** and **EdgePathPerimeter**.

Syntax

```
control.ComputeProperties( )
```

Parameter(s)

None

Return

No return value.

Example

```
Dim CenterX As Single
Dim CenterY As Single
Dim i As Long
Scene.ComputeProperties
For i = 0 To Scene.EdgePathCount - 1
    CenterX = Scene.EdgeCentroidX(i)
```

```

        CenterY = Scene.EdgeCentroidY(i)
    Next i

```

CopyFormatFrom

Description

Copies the format of a specified object to the current object.

Syntax

```
control.CopyFormatFrom(SourceObject As Object)
```

Parameter(s)

Name	Description	Type	Range
<i>SourceObject</i>	Object from which the format is copied.	HSScene or HSImage object	Not applicable.

Return

No return value.

Example

```
Scene1.CopyFormatFrom( Image1)
```

FindModelEdgePathFromEdgeI

Description

Returns the Model feature which contains the selected edgel.

Syntax

```
control.FindModelEdgePathFromEdgeI(EdgeIIndex As Long) As Long
```

Parameter(s)

Name	Description	Type	Range
<i>EdgeIIndex</i>	Index of the edge element in the scene.	Long	Not applicable.

Return

Long

Range

Minimum: 0

Maximum: **ModelEdgePathCount** - 1

Example

```

' Find the model feature of the edge
Dim ModelFeature As Long
ModelFeature = FindModelEdgePathFromEdgeI(0)

```

GetDispatchPointer

Description

Reserved for internal use.

ImageToWorld

Description

This method takes two parameters (X and Y) that represent a point in the Image coordinate system. When the method is executed the two entered parameters are modified to express the same point with respect to the World coordinate system.

Syntax

control.**ImageToWorld**(X As Single, Y As Single) As Boolean

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point.	Single	Any Single value.
Y	Y coordinate of the point.	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If the Scene is not calibrated then the method will return False.

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 200
Y1 = 350
Succeed = HSScene1.ImageToWorld(X1,Y1)
```

Load

Description

Loads the content of an scene file (*.hsc).

Syntax

control.**Load**([FileName As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the hsc file to load. Optional.	Variant	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSScene1.Load("..\DatabaseFiles\Scene.hsc")
Succeed = HSScene2.Load( )
```

Modify

Description

Modifies the **ModifyCount** property.

Syntax

control.**Modify**()

Parameter(s)

None

Return

No return value.

Example

```
HSScene1.Modify
```

Reset

Description

Clears the content of the Scene.

Syntax

control.**Reset**()

Parameter(s)

None

Return

No return value.

Example

```
HSScene1.Reset
```

Save

Description

Saves the contents of the Scene in the specified file.

Syntax

control.**Save**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file where the content of the Scene will be saved.	Variant	If parameter is a string: Range not applicable. No other variable types supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSScene1.Save("..\DatabaseFiles\Image1.hdb")
Succeed = HSScene1.Save( )
```

WorldToImage

Description

This method takes two parameters (X and Y) that represent a point in the World coordinate system. When the method is executed, the two entered parameters are modified to express the same point with respect to the Image coordinate system.

Syntax

control.**WorldToImage**(X As Single, Y As Single) As Boolean

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the point	Single	Any Single value.
Y	Y coordinate of the point	Single	Any Single value.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If the Scene is not calibrated then the method will return False.

Example

```
Dim Succeed As Boolean
Dim X1, Y1 As Single
X1 = 120.3
Y1 = 35.0
Succeed = HSScene1.WorldToImage(X1,Y1)
```

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

HSView

Description

An HSView control is a collection of Images, Scenes and Entities. By analogy, a View is a folder that can contain any combination of Images, Scenes and Entities.

Properties

Arc

Description

Arc object in the View.

Syntax

control.Arc (*ArcName* As String) [= *Value* As HSArc]

Parameter(s)

Parameter	Description	Type	Range
<i>ArcName</i>	Name of the arc object.	String	Not applicable.

Type

HSArc

Range

Not applicable.

Example

```
Dim lArc As HSArc
Set lArc = HSView1.Arc("InnerArc")
```

ArcCount

Description

Number of Arc objects in the View. Read only.

Syntax

control.ArcCount

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim lCount as Long
lCount = View.ArcCount
```

ArcExists

Description

Specifies if the arc object exists in the View. Read only.

Syntax

control.**ArcExists**(ArcName As String)

Type

Boolean

Range

Value	Description
True	Arc object exists.
False	Arc object does not exist.

Parameter(s)

Parameter	Description	Type	Range
ArcName	Name of the arc object.	String	Not applicable.

Example

```
Dim ExistingArc As Boolean
ExistingArc = HSView1.ArcExists("Arc_0")
```

ArcIndex

Description

Index of a arc object in the View. Read only.

Syntax

control.**ArcIndex** (ArcName As String)

Type

Long

Range

Minimum: 0

Maximum: **ArcCount** -1

Parameter(s)

Parameter	Description	Type	Range
ArcName	Name of the arc object.	String	Not applicable.

Remark(s)

A value of -1 is returned if the Arc does not exist.

Example

```
Dim lIndex as Long
lIndex = View.ArcIndex("Arc_0")
```

ArcName

Description

Name of the arc object in the View. Read only.

Syntax

control.**ArcName**(*Index* As Long)

Type

String

Range

Not applicable

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the arc object.	String	Minimum: 0 Maximum: ArcCount -1

Example

```
Dim Name as String
Name = View.ArcName(0)
```

Description

Description

String that describes the standard use of the control. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Descrip As String
Descrip = HSView1.Description
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

Image

Description

Image object in the View.

Syntax

```
control.Image (ImageName As String) [= Value As HSIImage]
```

Parameter(s)

Parameter	Description	Type	Range
<i>ImageName</i>	Name of the Image object.	String	Not applicable.

Type

HSImage

Range

Not applicable.

Example

```
Dim ImageObject As HSIImage
Set ImageObject = HSView1.Image("NewImage")
```

ImageCount

Description

Number of Image objects in the View. Read only.

Syntax

```
control.ImageCount
```

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim lCount as Long
lCount = View.ImageCount
```

ImageExists

Description

Specifies if the Image object exists in the View. Read only.

Syntax

```
control.ImageExists(ImageName As String)
```

Type

Boolean

Range

Value	Description
True	Image object exists.
False	Image object does not exist.

Parameter(s)

Parameter	Description	Type	Range
<i>ImageName</i>	Name of the Image object.	String	Not applicable.

Example

```
Dim ExistingImage As Boolean
ExistingImage = HSView1.ImageExists("Image_0")
```

ImageIndex

Description

Index of an Image object in the View. Read only.

Syntax

control.**ImageIndex** (*ImageName* As String)

Type

Long

Range

Minimum: 0

Maximum: **ImageCount** -1

Parameter(s)

Parameter	Description	Type	Range
<i>ImageName</i>	Name of the Image object.	String	Not applicable.

Example

```
Dim lIndex as Long
lIndex = View.ImageIndex("Image_0")
```

ImageName

Description

Name of the Image object in the View. Read only.

Syntax

control.**ImageName**(*Index* As Long)

Type

String

Range

Not applicable

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the Image object.	String	Minimum: 0 Maximum: ImageCount -1

Example

```
Dim Name as String
Name = View.ImageName(0)
```

Information

Description

String that provides information pertaining to the object. For example: Name, Boundary coordinates, Image Coordinate System, Timestamp, etc.

Syntax

control.**Information** [= *Value* As String]

Type

String

Range

Not applicable

Example

```
Dim Info As String
Info = HSView1.Information
```

Line

Description

Line object in the View.

Syntax

control.**Line** (*LineName* As String) [= *Value* As HSLine]

Parameter(s)

Parameter	Description	Type	Range
<i>LineName</i>	Name of the Line object.	String	Not applicable.

Type

HSLine

Range

Not applicable.

Example

```
Dim lLine As HSLine
Set lLine = HSView1.Line("InnerLine")
```

LineCount

Description

Number of Line objects in the View. Read only.

Syntax*control*.**LineCount****Type**

Long

Range

Greater than or equal to 0.

Example

```
Dim lCount as Long
lCount = View.LineCount
```

LineExists

Description

Specifies if the Line object exists in the View. Read only.

Syntax*control*.**LineExists**(*LineName* As String)**Type**

Boolean

Range

Value	Description
True	Line object exists.
False	Line object does not exist.

Parameter(s)

Parameter	Description	Type	Range
<i>LineName</i>	Name of the Line object.	String	Not applicable.

Example

```
Dim ExistingLine As Boolean
ExistingLine = HSView1.LineExists("Line_0")
```

LineIndex

Description

Index of a Line object in the View. Read only.

Syntax*control*.**LineIndex** (*LineName* As String)**Type**

Long

Range

Minimum: 0

Maximum: **LineCount** -1

Parameter(s)

Parameter	Description	Type	Range
<i>LineName</i>	Name of the Line object.	String	Not applicable.

Remark(s)

A value of -1 is returned if the Line does not exist.

Example

```
Dim lIndex as Long
lIndex = View.LineIndex("Line_0")
```

LineName

Description

Name of the Line object in the View. Read only

Syntax

control.**LineName**(*Index* As Long)

Type

String

Range

Not applicable

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the Line object.	String	Minimum: 0 Maximum: LineCount -1

Example

```
Dim Name as String
Name = View.LineName(0)
```

ModifyCount

Description

Number of modifications that have been made to the View. Read only.

Syntax

control.**ModifyCount**

Type

Integer

Range

Greater than or equal to zero.

Example

```
Dim Count As Integer
Count = HSView1.ModifyCount
```

Name

Description

Name used to access the object.

Syntax

control.**Name**[= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = HSView1.Name
HSView2.Name = "View_00"
```

Point

Description

Point object in the View.

Syntax

control.**Point** (*PointName* As String) [= *Value* As HSPoint]

Parameter(s)

Parameter	Description	Type	Range
<i>PointName</i>	Name of the Point object	String	Not applicable.

Type

HSPoint

Range

Not applicable.

Example

```
Dim lPoint As HSPoint
Set lPoint = HSView1.Point("InnerPoint")
```

PointCount

Description

Number of Point objects in the View. Read only.

Syntax

control.**PointCount**

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim lCount as Long
lCount = View.PointCount
```

PointExists

Description

Specifies if the Point object exists in the View. Read only.

Syntax

control.**PointExists**(*PointName* As String)

Type

Boolean

Range

Value	Description
True	Point object exists.
False	Point object does not exist.

Parameter(s)

Parameter	Description	Type	Range
<i>PointName</i>	Name of the Point object.	String	Not applicable.

Example

```
Dim ExistingPoint As Boolean
ExistingPoint = HSVView1.PointExists("Point_0")
```

PointIndex

Description

Index of a Point object in the View. Read only.

Syntax

control.**PointIndex** (*PointName* As String)

Type

Long

Range

Minimum: 0

Maximum: **PointCount** -1

Parameter(s)

Parameter	Description	Type	Range
<i>PointName</i>	Name of the Point object.	String	Not applicable.

Remark(s)

A value of -1 is returned if the Point does not exist.

Example

```
Dim lIndex as Long
lIndex = View.PointIndex("Point_0")
```

PointName

Description

Name of the Point object in the View. Read only.

Syntax

control.**PointName**(*Index* As Long)

Type

String

Range

Not applicable

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the Point object.	String	Minimum: 0 Maximum: PointCount -1

Example

```
Dim Name as String
Name = View.PointName(0)
```

Scene

Description

Scene object in the View.

Syntax

control.**Scene** (*SceneName* As String) [= Value As HSScene]

Parameter(s)

Parameter	Description	Type	Range
<i>SceneName</i>	Name of the Scene object.	String	Not applicable.

Type

HSScene

Range

Not applicable.

Example

```
Dim SceneObject As HSScene
Set SceneObject = HSView1.Scene("NewScene")
```

SceneCount

Description

Number of Scene objects in the View. Read only.

Syntax

control.**SceneCount**

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim lCount as Long
lCount = View.SceneCount
```

SceneExists

Description

Specifies if the Scene object exists. Read only.

Syntax

control.**SceneExists**(SceneName As String)

Type

Boolean

Range

Value	Description
True	Scene object exists.
False	Scene object does not exist.

Parameter(s)

Parameter	Description	Type	Range
<i>SceneName</i>	Name of the Scene object.	String	Not applicable.

Example

```
Dim Exist As Boolean
Exist = View1.SceneExists("Scene_0")
```

SceneIndex

Description

Index of the Scene object. Read only.

Syntax

control.**SceneIndex** ([SceneName As String])

Type

Long

Range

Minimum: 0

Maximum: **SceneCount** -1

Parameter(s)

Parameter	Description	Type	Range
<i>SceneName</i>	Name of the Scene object.	String	Not applicable.

Remark(s)

A value of -1 is returned if the Scene does not exist.

Example

```
Dim lIndex as Long
```

```
lIndex = View.SceneIndex("Scene_0")
```

SceneName

Description

Name of the Scene object in the View. Read only

Syntax

control.**SceneName**(*Index* As Long)

Type

String

Range

Not applicable

Parameter(s)

Parameter	Description	Type	Range
<i>Index</i>	Index of the Scene object.	String	Minimum: 0 Maximum: SceneCount -1

Example

```
Dim Name as String
Name = View.SceneName(0)
```

TypeName

Description

ActiveX type name for the object. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSView"

Example

```
Dim ObjectType As String
ObjectType = View1.TypeName
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = HSView1.Version
```

Methods

AboutBox

Description

Displays the HSView control About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

```
HSView1.AboutBox
```

AddArc

Description

Adds an Arc object to the View.

Syntax

control.**AddArc**(ArcName As String) As HSArc

Parameter(s)

Parameter	Description	Type	Range
<i>ArcName</i>	Name of the Arc object that will be added.	String	Not applicable.

Return

HSArc

Range

Not applicable.

Example

```
Dim NewArcObject As HSArc
Set NewArcObject = _
    HSView1.AddArc("Arc")
```

AddImage

Description

Adds a new Image to the View.

Syntax

control.**AddImage**(*ImageName* As String, *ImageType* As VARIANT) As HSImage

Parameter(s)

Parameter	Description	Type	Range
<i>ImageName</i>	Name of the Image that will be added.	String	Not applicable.
<i>ImageType</i>	Type of the Image that will be added.	VARIANT	hsImageReserved, hsImage8bppGreyScale, hsImage16bppGreyScale, hsImage16bppHeightMap, hsImage32bppRgb

Return

HSImage

Range

Not applicable.

Example

```
Dim NewImage As HSImage
Set NewImage = HSView1.AddImage("Modified Image",hsImage8bppGreyScale)
```

AddLine

Description

Adds a Line object to the View

Syntax

control.**AddLine**(*LineName* As String) As HSLine

Parameter(s)

Parameter	Description	Type	Range
<i>LineName</i>	Name of the Line object that will be added.	String	Not applicable.

Return

HSLine

Range

Not applicable.

Example

```
'Build a Line in the current View
Dim lLine As HSLine
Set lLine = myView.AddLine ("Line")
    lLine.Calibration = lImageCalibration
    lLine.Calibrated = False
    lLine.StartPoint.ImageX = 100
    lLine.StartPoint.ImageY = 100
    lLine.EndPoint.ImageX = 300
    lLine.EndPoint.ImageY = 300
```

AddPoint

Description

Adds a Point object to the View.

Syntax

control.**AddPoint**(*PointName* As String) As HSPoint

Parameter(s)

Parameter	Description	Type	Range
<i>PointName</i>	Name of the Point object that will be added.	String	Not applicable.

Return

HSPoint

Range

Not applicable.

Example

```
'Build a Point in the current View
Dim lPoint as HSPoint
Set lPoint = myView.AddPoint ("Point")
    lPoint.Calibration = lImageCalibration
    lPoint.ImageX = 320
    lPoint.ImageY = 240
```

AddScene

Description

Adds a new Scene to the View.

Syntax

control.**AddView**(*SceneName* As String) As HSScene

Parameter(s)

Parameter	Description	Type	Range
<i>PointName</i>	Name of the Scene that will be added.	String	Not applicable.

Return

HSScene

Range

Not applicable.

Example

```
Dim MyScene As HSScene
Set MyScene = HSVIEW1.AddScene ("NewScene")
```

GetDispatchPointer

Description

Reserved for internal use.

Load

Description

Loads a View from a file.

Syntax

control.**Load**([*FileName* As Variant]) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file that will be loaded.	Variant	If parameter is a string: Range not applicable. No other variable types supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSView1.Load("..\DatabaseFiles\View.hdb")
Succeed = HSView2.Load( )
```

Modify

Description

Modifies the **ModifyCount** property.

Syntax

control.**Modify**()

Parameter(s)

None

Return

No return value.

Example

```
HSView1.Modify
```

RemoveArc

Description

Removes selected Arc object from the View.

Syntax

control.**RemoveArc**(*ArcName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ArcName</i>	Name of the Arc object that will be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RemoveArc("ArcEntity")
```

RemoveImage

Description

Removes selected Image object from the View.

Syntax

```
control.RemoveImage(ImageName As String) As HSImage
```

Parameter(s)

Parameter	Description	Type	Range
<i>ImageName</i>	Name of the Image that will be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RemoveImage("Image_15")
```

RemoveLine

Description

Removes selected Line object from the View.

Syntax

```
control.RemoveLine(LineName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>ArcName</i>	Name of the Line object that will be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RemoveLine("Perpendicular")
```

RemovePoint

Description

Removes the selected Point object from the View.

Syntax

control.**RemovePoint**(*PointName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ArcName</i>	Name of the Point object that will be removed.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RemovePoint("PointEntity")
```

RemoveScene

Description

Removes the selected Scene object from the View.

Syntax

control.**RemoveScene**(*SceneName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>SceneName</i>	Name of the Scene that will be removed	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RemoveScene("Scene_00")
```

RenameArc

Description

Renames the selected Arc object in the View.

Syntax

control.**RenameArc**(*ArcName* As String, *NewArcName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ArcName</i>	Name of the Arc object that will be renamed.	String	Not applicable.
<i>NewArcName</i>	New name that will be given to the Arc.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RenameArc("ArcEntity","Arc")
```

RenameImage

Description

Renames the selected Image in the View.

Syntax

control.**RenameImage**(*ImageName* As String, *NewImageName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>ImageName</i>	Name of the Image that will be renamed.	String	Not applicable.
<i>NewImageName</i>	New name that will be given to the Image.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RenameImage("Image_00","Image_15")
```

RenameLine

Description

Renames the selected Line object in the View.

Syntax

control.**RenameLine**(*LineName* As String, *NewLineName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>LineName</i>	Name of the Line object that will be renamed.	String	Not applicable.
<i>NewLineName</i>	New name that will be given to the Line.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RenameLine("InnerLine_00", "OuterLine_00")
```

RenamePoint

Description

Renames the selected Point object in the View.

Syntax

control.**RenamePoint**(*PointName* As String, *NewPointName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>PointName</i>	Name of the Point object that will be renamed.	String	Not applicable.
<i>NewPointName</i>	New name that will be given to the Point.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RenamePoint("MidPoint_00", "EndPoint_00")
```

RenameScene

Description

Renames the selected Scene object in the View.

Syntax

control.**RenameScene**(*SceneName* As String, *NewSceneName* As String) As Boolean

Parameter(s)

Parameter	Description	Type	Range
<i>SceneName</i>	Name of the Scene that will be renamed.	String	Not applicable.
<i>NewSceneName</i>	New name that will be given to the Scene.	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = HSView1.RenameScene("Scene_00", "Scene_Main")
```

Reset

Description

Clears the content of the View.

Syntax

```
control.Reset( )
```

Parameter(s)

None

Return

No return value.

Example

```
HSView1.Reset
```

Save

Description

Saves the contents of the View in the specified file.

Syntax

```
control.Save([FileName As Variant]) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>FileName</i>	Path and name of the file where the content of the View will be saved.	Variant	If parameter is a string: Range not applicable. No other variable types supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

When no parameter is provided with the function call, a file browser dialog is displayed to allow the user to select or to enter the name of the file.

Example

```
Dim Succeed As Boolean
Succeed = HSView1.Save("..\DatabaseFiles\View1.hdb")
```

Events

DatabaseChange

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

Color Finder

Description

The Color Finder Tool processes the input image using user-defined filters. Processing statistics are outputted for each defined filter. Best filter match information and global image statistics are also available as results.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the Image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
```

```
Result = ColorTool.BilinearInterpolationEnabled
```

```
ColorTool.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = ColorTool.CalibratedUnitsEnabled
ColorTool.CalibratedUnitsEnabled = False
```

ClearOutputFilterImageEnabled

Description

Specifies if the image output by the tool will be cleared in the next execution of the tool.

Syntax

control.ClearOutputFilterImageEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
False	The output image of the color finder will not be cleared.
True	The output image of the color finder will be cleared.

Example

```
Dim Enabled As Boolean
Enabled = ColorTool.ClearOutputBlobImageEnabled
ColorTool.ClearOutputBlobImageEnabled = False
```

ColorFilterBestMatchIndex

Description

The index number of the filter having the best match quality. Read only.

Syntax

control.ColorFilterBestMatchIndex

Type

Integer

Range

Minimum: 0

Maximum: **ColorFilterCount - 1**

Example

```
Dim Index As Integer
Index = ColorTool.ColorFilterBestMatchIndex
```

ColorFilterBestMatchName

Description

The name of the filter having the best match quality. Read only.

Syntax

control.ColorFilterBestMatchName

Type

String

Range

Not applicable.

Example

```
Dim FilterName As String
FilterName = ColorTool.ColorFilterBestMatchName
```

ColorFilterCount

Description

Number of color filters defined. Read only.

Syntax*control*.ColorFilterCount**Type**

Integer

Range

Greater than or equal to zero.

Example

```
Dim Number As Integer
Number = ColorTool.ColorFilterCount
```

ColorFilterEnabled

Description

Specifies if the selected color filter is enabled, which means that the Color Finder will use this filter for statistical analysis when executed.

Syntax*control*.ColorFilterEnabled(*Filter* As Variant) [= *NewValue* As Boolean]**Type**

Boolean

Range

Value	Description
True	Filter is enabled.
False	Filter is disabled.

Parameter(s)

Name	Description	Type	Range
<i>Filter</i>	Index or name of the selected filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Enabled As Boolean
```

Enabled = ColorTool.**ColorFilterEnabled**(0)

ColorFilterMatchPixelCount

Description

Number of pixels in search area that match the conditions set by the filter. Read only.

Syntax

control.**ColorFilterMatchPixelCount**(*Filter* As Variant)

Type

Integer

Range

Minimum: 0.

Maximum: **ImagePixelCount**

Parameter(s)

Name	Description	Type	Range
<i>Filter</i>	Index or name of the selected filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim FilterPixelCount As Integer
FilterPixelCount = ColorTool.ColorFilterMatchPixelCount(1)
```

ColorFilterMatchQuality

Description

Amount of search area that match the conditions set by the filter. Match quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that 100% of the search area were successfully matched to the specified filter. Read only.

Syntax

control.**ColorFilterMatchQuality**(*Filter* As Variant)

Type

Single

Range

Minimum: 0.0.

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Filter</i>	Index or name of the selected filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim MatchQuality As Single
MatchQuality = ColorTool.ColorFilterMatchQuality("RED")
```

ColorFilterName

Description

Name of the selected color filter.

Syntax

control.**ColorFilterName**(*FilterIndex* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>FilterIndex</i>	Index of the selected filter.	Integer	Minimum: 0 Maximum: ColorFilterCount - 1

Example

```
Dim Name As String
Name = ColorTool.ColorFilterName(0)
ColorTool.ColorFilterName(1) = "RED"
```

CoordinateSystem

Description

Coordinate system used to express the results.

Syntax

control.**CoordinateSystem** [= *Value* As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.

Constant	Value	Description
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim CSystem As hsCoordinateSystem
CSystem = ColorTool.CoordinateSystem
ColorTool.CoordinateSystem = hsCoordinateSystemTool
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard usage of the Color Finder. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = ColorTool.Description
```

ElapsedTime

Description

Total time elapsed (in milliseconds) during the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = ColorTool.ElapsedTime
```

ErrorObject

DescriptionReturns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.**Syntax***control*.**ErrorObject****Type**

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & ColorTool.ErrorObject.Description
```

ImageHeight

Description

Height of the tool's region of interest expressed in pixels. Read only.

Syntax*control*.**ImageHeight****Type**

Long

Range

Greater than or equal to 0.

Example

```
Dim Height As Long
Dim Width As Long
Dim Count As Long
Height = ColorTool.ImageHeight
Width = ColorTool.ImageWidth
Count = Height * Width
```

ImagePixelCount

Description

Number of pixels in the tool's region of interest. Read only.

Syntax

control.**ImagePixelCount**

Type

Long

Range

Greater than or equal to 0.

Example

Dim Count As Long

Count = ColorTool.**ImagePixelCount**

ImageSubsampling

Description

Factor used to subsample the image in the tool's region of interest. With a subsampling factor of 1, the image is not subsampled. With a subsampling factor of 2, the image is subsampled in tiles of 2x2 pixels. With a subsampling factor of 3 the image is subsampled in tiles of 3x3 pixels and so forth.

Syntax

control.**ImageSubsampling** [= Value As Integer]

Type

Integer

Range

[1,2,3,4,5,6,7,8]

Remark(s)

Using a higher subsampling factor may speeds up the processing but slightly reduces the accuracy of the statistics computed. The pixel properties computed by the Color Finder tool are normalized with respect to the subsampling factor (**ImageHeight**, **ImagePixelCount** and **ImageWidth**). So for instance, the total number of pixels in the image should remain the same at any subsampling factor. Note that there might be slight differences in the values of these properties when either of the width or the height of the region of interest is not a multiple of the subsampling factor used.

Example

Dim Subsampling As Integer

Subsampling = ColorTool.**ImageSubsampling**

ColorTool.**ImageSubsampling** = 8

ImageWidth

Description

Width of the tool's region of interest expressed in pixels. Read only.

Syntax

control.**ImageWidth**

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim Height As Long
Dim Width As Long
Dim Count As Long
Height = ColorTool.ImageHeight
Width = ColorTool.ImageWidth
Count = Height * Width
```

InputImage

Description

Name of the Image on which the tool is applied.

Syntax

control.**InputImage** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim InputImage As String
InputImage = ColorTool.InputImage
ColorTool.InputImage = "Color Image"
```

InputGreyScaleImage

Description

Name of the Image on which the tool is applied.

Syntax

control.**InputImage** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim InputImage As String
InputImage = ColorTool.InputImage
ColorTool.InputImage = "Color Image"
```

InputView

Description

Name of the View from which the Image is retrieved. See **InputImage** for more details.

Syntax

control.**InputView** [= Value As String]

Type

String

Range

Not applicable.

Remark(s)See also **InputImage**.**Example**

```
Dim InputImageView As String
InputImageView = ColorTool.InputView
ColorTool.InputView = "Acquisition View"
```

InputGreyScaleView

DescriptionName of the View from which the Image is retrieved. See **InputImage** for more details.**Syntax***control*.**InputView** [= Value As String]**Type**

String

Range

Not applicable.

Remark(s)See also **InputImage**.**Example**

```
Dim InputImageView As String
InputImageView = ColorTool.InputView
ColorTool.InputView = "Acquisition View"
```

InputInstanceScene

Description

Name of the Scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax*control*.**InputInstanceScene** [=Name As String]**Type**

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = ColorTool.InputInstanceScene
ColorTool.InputInstanceScene = "scene 0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.InputInstanceView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = ColorTool.InputInstanceView
ColorTool.InputInstanceView = "view 0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.InputModelScene [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = ColorTool.InputModelScene
ColorTool.InputModelScene = "model scene 0"
```

InputModelView

Description

Name of the View that provides the Model Scene. See **InputModelScene** for more details.

Syntax

control.InputModelView [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = ColorTool.InputModelView
ColorTool.InputModelView = "view 0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

control.InstanceSelectionEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Enabled As Boolean
Enabled = ColorTool.InstanceSelectionEnabled
ColorTool.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** properties are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.InstanceSelectionIndex [= Value As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the referenced Locator minus 1

Example

```
Dim Index As Integer
Index = ColorTool.InstanceSelectionIndex
ColorTool.InstanceSelectionIndex = 0
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input Image. This

transform can be supplied manually using the **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.ModelTransformEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = ColorTool.ModelTransformEnabled
ColorTool.ModelTransformEnabled = True
```

Name

Description

Name used to access the Color Finder tool through the Process Manager.

Syntax

control.Name

Type

String

Range

Not applicable.

Example

```
Dim ToolName As String
ToolName = ColorTool.Name
ColorTool.Name = "Foreground Color Finder"
```

OutputFilterImage

Description

Name of the filter image created by the tool. This image is optionally output according to the value of the **OutputFilterImageEnabled** property. It contains the result of the color filter finding process.

Syntax

control.OutputFilterImage [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
```

```

ImageName = ColorTool.OutputFilterImage
ColorTool.OutputFilterImage = "Color Filtered Image"

```

OutputFilterImageEnabled

Description

Specifies if a filter image will be output after the color filter finding process.

Syntax

```
control.OutputFilterImageEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
False	The image will not be output.
True	The image will be output.

Remark(s)

Generating a filter image considerably increases the tool execution time.

Example

```

Dim Enable As Boolean
Enable = ColorTool.OutputFilterImageEnabled
ColorTool.OutputFilterImageEnabled = False

```

OutputFilterView

Description

Name of the View to which the Output Filter image is output. See **OutputFilterImage** for more details.

Syntax

```
control.OutputFilterView [= Name As String]
```

Type

String

Range

Not applicable.

Remark(s)

Example

```

Dim ViewName As String
ViewName = ColorTool.OutputBlobView
ColorTool.OutputBlobView = "BlobView0"

```

SamplingStep

Description

Step used in the tool's last execution to sample the box from the input Image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the

ModelTransformEnabled properties. All pixels in the sampled box are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input Image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

Dim Step As Single

Step = ColorFinder.**SamplingStep**

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input Image.

Remark(s)

A custom sampling step is usually not recommended.

Example

Dim Step As Single

Step = ColorTool.**SamplingStepCustom**

ColorTool.**SamplingStepCustom** = 2.1

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom .

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Enabled As Boolean
Enabled = ColorTool.SamplingStepCustomEnabled
ColorTool.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

control.**SamplingStepDefault**

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input Image.

Example

```
Dim Sampling As Single
Sampling = ColorTool.SamplingStepDefault
```

ToolHeight

Description

Height of the tool rectangle.

SyntaxF

control.**ToolHeight** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height as Single
Height = ColorFinder.ToolHeight
```

`ColorFinder.ToolHeight = 15.2`

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

`control.ToolPositionX [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units, and with respect to a coordinate system, as specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim X_Pos As Single
X_Pos = ColorTool.ToolPositionX
ColorTool.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

`control.ToolPositionY [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units, and with respect to a coordinate system, as specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World

Mode	Length unit	Coordinate system
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = ColorTool.ToolPositionY
ColorTool.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system, as specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = ColorTool.ToolRotation
ColorTool.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= Value as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = ColorTool.ToolWidth
ColorTool.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = ColorTool.TransformRotation
ColorTool.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation

and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

`control.TransformScale [= Value As Single]`

Type

Single

Range

Minimum: Greater than zero.

Maximum: Unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = ColorTool.TransformScale
ColorTool.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

`control.TransformTranslationX [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = ColorTool.TransformTranslationX
ColorTool.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = ColorTool.TransformTranslationY
ColorTool.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the Color Finder tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSColorFinder".

Example

```
Dim TypeName As String
TypeName = ColorTool.TypeName
```

UnitsLength

Description

Unit used to express calibrated lengths and positions.

Syntax

control.UnitsLength [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter

Constant	Value	Description
hsMicron	3	micron
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnits	8	generic calibration units

Example

```
Dim UnitsLength As hsUnitsLength
UnitsLength = ColorTool.UnitsLength
ColorTool.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	Radian
hsDegree	1	Degree

Example

```
Dim UnitsOrientation As hsUnitsOrientation
UnitsOrientation = ColorTool.UnitsOrientation
ColorTool.UnitsOrientation = hsRadian
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "4.0.000 R 4405 (release)".

Example

```
Dim Version As String
Version = ColorTool.Version
```

Methods

AboutBox

Description

Displays the Color Finder tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

```
ColorTool.AboutBox
```

AddColorFilter

Description

Adds a new color filter to the list of filters configured for the Color Finder tool. Note that the newly created color filter is initialized with default internal values not related to selected input image. Modifying newly created color filter values can be done either interactively or programmatically. The **SetColorFilterRGBValues**, **SetColorfilterHSLValues** and **SetColorFilterHSLTolerances** methods can be used, to programmatically set properties of a color filter.

Syntax

control.**AddColorFilter**(*FilterName* As String) As Integer

Parameter(s)

Name	Description	Type	Range
<i>FilterName</i>	Name of the Filter to be added.	String	Not applicable.

Return

Type	Description	Range
Integer	Index of the newly created filter.	Successful: Greater than or equal to 0 Failed: -1

Example

```
Dim NewIndex As Integer
NewIndex = ColorTool.AddColorFilter("BLUE")
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Color Finder process.

Syntax

control.**Execute**() As Single

Parameter(s)

None

Return

Type	Description	Range
Single	Execution time in milliseconds.	Greater than 0.0

Example

Time As Single

Time = ColorTool.**Execute**

ExportColorFilter

Description

Exports a color filter to the current list of filters to a specific color filter file (*.clf).

Syntax

control.**ExportColorFilter**(*FileName* As String, *Filter* As Variant) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of configuration file to export.	String	Not applicable.
<i>Filter</i>	Index or name of the filter to be removed.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.

Return

Boolean

Example

ColorTool.**ExportColorFilter** "red.clf", "RED"

GetColorFilterHSLTolerances

Description

This method fills three parameters (*Hue*, *Saturation* and *Luminance*) values when executed. Those parameters returns respectively the *Hue*, *Saturation* and *Luminance* tolerances defined for the specified color filter.

Syntax

control.**GetColorFilterHSLTolerances**(*Filter* As Variant, *Hue* As Integer, *Saturation* As Integer, *Luminance* As Integer)

Parameter(s)

Parameter	Description	Type	Range
<i>Filter</i>	Index or name of the filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.
<i>Hue</i>	Hue tolerance value of the filter.	Integer	Minimum: 0 Maximum: 128
<i>Saturation</i>	Saturation tolerance value of the filter.	Integer	Minimum: 0 Maximum: 128
<i>Luminance</i>	Luminance tolerance value of the filter.	Integer	Minimum: 0 Maximum: 128

Return

No return value..

Example

```
Dim H, S, L As Integer
ColorTool.GetColorFilterHSLTolerances "RED", H, S, L
```

GetColorFilterHSLValues

Description

This method fills three parameters (*Hue*, *Saturation* and *Luminance*) values when executed. Those parameters returns respectively the *Hue*, *Saturation* and *Luminance* values defined for the specified color filter.

Syntax

control.**GetColorFilterHSLValues**(*Filter* As Variant, *Hue* As Integer, *Saturation* As Integer, *Luminance* As Integer)

Parameter(s)

Parameter	Description	Type	Range
<i>Filter</i>	Index or name of the filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.
<i>Hue</i>	Hue value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Saturation</i>	Saturation value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Luminance</i>	Luminance value of the filter.	Integer	Minimum: 0 Maximum: 255

Return

No return value..

Example

```
Dim H, S, L As Integer
ColorTool.GetColorFilterHSLValues "RED", H, S, L
```

GetColorFilterRGBValues

Description

This method fills three parameters (*Red*, *Green* and *Blue*) values when executed. Those parameters returns respectively the *Red*, *Green* and *Blue* values defined for the specified color filter.

Syntax

control.**GetColorFilterRGBValues**(*Filter* As Variant, *Red* As Integer, *Green* As Integer, *Blue* As Integer)

Parameter(s)

Parameter	Description	Type	Range
<i>Filter</i>	Index or name of the filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.
<i>Red</i>	Red value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Green</i>	Green value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Blue</i>	Blue value of the filter.	Integer	Minimum: 0 Maximum: 255

Return

No return value..

Example

```
Dim R, G, B As Integer
ColorTool.GetColorFilterRGBValues "RED", R, G, B
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the Color Finder properties interface.

Syntax*control.HideProperties***Parameter(s)**

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = ColorTool.HideProperties
```

ImportColorFilter

Description

Imports a color filter to the current list of filters configured for the Color Finder tool.

Syntax*control.ImportColorFilter(FileName As String)***Parameter(s)**

Name	Description	Type	Range
<i>FileName</i>	Name of the file to load.	String	Not applicable.

Return

Type	Description	Range
Integer	Index of the newly imported filter.	Successful: Greater than or equal to 0 Failed: -1

Example

```
ColorTool.ImportModel "green.clf"
```

LoadConfiguration

Description

Reserved for internal use.

RemoveColorFilter

Description

Removes a color filter from the current list of filters configured for the Color Finder tool.

Syntax*control.RemoveColorFilter(Filter As Variant) As Boolean*

Parameter(s)

Name	Description	Type	Range
<i>Filter</i>	Index or name of the filter to be removed.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
ColorTool.RemoveColorFilter "RED"
ColorTool.RemoveColorFilter 0
```

SaveConfiguration

Description

Reserved for internal use.

SetColorFilterHSLTolerances

Description

This method sets respectively the *Hue*, *Saturation* and *Luminance* tolerances defined for the specified color filter.

Syntax

```
control.SetColorFilterHSLTolerances(Filter As Variant, Hue As Integer, Saturation As Integer, Luminance As Integer)
```

Parameter(s)

Parameter	Description	Type	Range
<i>Filter</i>	Index or name of the filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.
<i>Hue</i>	Hue tolerance value of the filter.	Integer	Minimum: 0 Maximum: 128
<i>Saturation</i>	Saturation tolerance value of the filter.	Integer	Minimum: 0 Maximum: 128
<i>Luminance</i>	Luminance tolerance value of the filter.	Integer	Minimum: 0 Maximum: 128

Return

No return value..

Example

```
ColorTool.SetColorFilterHSLTolerances "RED", 10, 10, 10
```

SetColorFilterHSLValues

Description

This method sets respectively the *Hue*, *Saturation* and *Luminance* values defined for the specified color filter.

Syntax

control.**SetColorFilterHSLValues**(*Filter* As Variant, *Hue* As Integer, *Saturation* As Integer, *Luminance* As Integer)

Parameter(s)

Parameter	Description	Type	Range
<i>Filter</i>	Index or name of the filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.
<i>Hue</i>	Hue value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Saturation</i>	Saturation value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Luminance</i>	Luminance value of the filter.	Integer	Minimum: 0 Maximum: 255

Return

No return value..

Example

ColorTool.**SetColorFilterHSLValues** "RED", 85, 255, 128

SetColorFilterRGBValues

Description

This method sets respectively the *Red*, *Green* and *Blue* values defined for the specified color filter.

Syntax

control.**SetColorFilterRGBValues**(*Filter* As Variant, *Red* As Integer, *Green* As Integer, *Blue* As Integer)

Parameter(s)

Parameter	Description	Type	Range
<i>Filter</i>	Index or name of the filter.	Variant	Integer: Minimum: 0 Maximum: ColorFilterCount - 1 String: Range not applicable. Other types not supported.
<i>Red</i>	Red value of the filter.	Integer	Minimum: 0 Maximum: 255
<i>Green</i>	Green value of the filter.	Integer	Minimum: 0 Maximum: 255

Parameter	Description	Type	Range
<i>Blue</i>	Blue value of the filter.	Integer	Minimum: 0 Maximum: 255

Return

No return value..

Example

```
ColorTool.SetColorFilterRGBValues "RED", 255, 0, 0
```

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the Color Finder properties interface.

Syntax

```
control.ShowProperties(Modal As Boolean, [Option As Variant]) As Boolean
```

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	True, False
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface, if the window should be displayed topmost, and what the behavior of the control is when the OK button is clicked. When this parameter is not provided, all the elements of the bitfield are set except <code>hsExecuteOnOK</code> , <code>hsExecuteRemaining</code> and <code>hsDisplayTopMost</code> .	Variant	Bitfield elements shown in table below.

<i>Option</i> Field	Value	Description
<code>hsShowApply</code>	0x01	Displays APPLY button
<code>hsShowReset</code>	0x02	Displays RESET button
<code>hsShowDefault</code>	0x04	Displays DEFAULT button
<code>hsShowExecute</code>	0x08	Displays EXECUTE button
<code>hsShowOK</code>	0x10	Displays OK button

Option Field	Value	Description
hsExecuteOnOK	0x20	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	0x40	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the bitfields hsExecuteOnOK and hsExecuteRemaining are both set.
hsDisplayTopMost	0x80	Displays the dialog as topmost.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = ColorFinder.ShowProperties(True)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

DataMatrix Reader

Description

The DataMatrix Reader reads and extracts information from data matrix symbology.

Properties

AdaptiveThresholdEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **AdaptiveThresholdEnabled** is set to True, the tool activates the adaptive threshold method. Three threshold methods are available: global (**GlobalThresholdEnabled**), local (**LocalThresholdEnabled**), and adaptive (**AdaptiveThresholdEnabled**). These three methods are not exclusive and can be enabled simultaneously. When reading high quality marks, the global method is sufficient. With codes where the spacing between dots is uneven, the local method is recommended. When the background of the matrix is uneven, the adaptive method may produce better results. The global method is usually faster than both the local and adaptive methods. For the local and adaptive methods to work efficiently, **AverageElementSize** and **SamplesPerCell** should both be set to a value higher than 5 and **WarpMethod** should be set to **hsMatrixWarpSlow**. This property is auto-configured after a successful **Learn**.

Syntax

control.AdaptiveThresholdEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The adaptive threshold method is enabled.
False	The adaptive threshold method is disabled.

Example

```
' Read the matrix using the adaptive and global threshold
' methods
DataMatrix1.AdaptiveThresholdEnabled = True
DataMatrix1.GlobalThresholdEnabled = True
DataMatrix1.LocalThresholdEnabled = False
DataMatrix1.Execute
```

AngleVariation

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The maximum allowable variation, in degrees, of the angle of the data matrix to read, with respect to the angle set by the **MatrixOrientation** property. By default, the maximum allowable variation is 23 degrees. The **AngleVariation** and **MatrixOrientation** properties are ignored if **MatrixOrientationEnabled** is set to False.

Syntax

control.**AngleVariation** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 180

Example

```
' Search for a data matrix at 45 degrees and allow a
' maximum variation of +/- 10 degrees from this angle
DataMatrix1.MatrixOrientation = hsMatrixOrientation45
DataMatrix1.MatrixOrientationEnabled = True
DataMatrix1.AngleVariation = 10
DataMatrix1.Execute
```

AverageElementSize

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Average size, in pixels, of the cells in the data matrix. For optimal performance, the field of view should be set so that the average cell size falls between 5 and 10 pixels. With relatively large and well contrasted matrixes, **AverageElementSize** can be lowered slightly to speed up the read cycle. Setting **AverageElementSize** to a value lower than 4 pixels will increase the processing time significantly. This property is auto-configured after a successful **Learn**.

Syntax

control.*control*.**AverageElementSize** [= *Value* As Integer]

Type

Integer

Range

Minimum: 2

Maximum: 20

Example

```
' Reduce the average cell size to speed up the process
DataMatrix1.AverageElementSize = 5
DataMatrix1.Execute
```

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
Result = DataMatrix1.BilinearInterpolationEnabled
DataMatrix2.BilinearInterpolationEnabled = True
```

BorderIntegrity

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The minimum percentage of the border that needs to be found in order to correctly decode a data matrix. The default is 80%, which means a 20% border mismatch or damage is tolerated.

Syntax

control.**BorderIntegrity** [= *Value* As Integer]

Type

Integer

Range

Minimum: 75
Maximum: 100

Example

```
' Decrease border integrity to read a degraded code
DataMatrix1.BorderIntegrity = 75
DataMatrix1.Execute
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = DataMatrix1.CalibratedUnitsEnabled
DataMatrix2.CalibratedUnitsEnabled = False
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

hsCoordinateSystem	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim CurrentSystem As hsCoordinateSystem
' Get the current coordinate system
CurrentSystem = DataMatrix1.CoordinateSystem
```

' Express the results in the Object coordinate system
 DataMatrix1.**CoordinateSystem** = hsCoordinateSystemObject

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Info As String
Info = DataMatrix1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = DataMatrix1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax*control*.**ErrorObject****Type**

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & _
    DataMatrix1.ErrorObject.Description
```

FineTuneMethodEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Property used to configure methods to fine-tune the performance of the data matrix decoding algorithm. This property is auto-configured after a successful **Learn**.

Syntax*control*.**FineTuneMethodEnabled** (*Method* As hsMatrixFineTuneMethod) [= *Value* As Boolean]**Type**

Boolean

Range

Value	Description
True	The fine-tuning method is enabled
False	The fine-tuning method is disabled

Parameter(s)

Name	Description	Type	Range
<i>Method</i>	Fine-tuning method to enable/disable	hsMatrixFineTuneMethod	Enum. See table below.

hsMatrixFineTuneMethod	Value	Description
hsMatrixPositionEnhance	0	Enables an extra enhancement step when reading a high density data matrix with low resolution.
hsMatrixIntensityEnhance	1	Increases capacity to read a data matrix in which borders contain a high variation in light intensity.
hsMatrixNoQuietZoneClutter	2	Speeds the reading of a data matrix that has a sufficient, uncluttered quiet zone.

hsMatrixFineTuneMethod	Value	Description
hsMatrixNoTopTimingBorder	3	Enables reading of a data matrix having no top timing border.
hsMatrixNoRightTimingBorder	4	Enables reading of a data matrix having no right timing border.
hsMatrixAllowSteepAngle	5	Increases capacity to read a data matrix with severe geometrical distortion
hsMatrixAllowSeverDamage	6	Increases capacity to read a data matrix with severe border damage.

Example

```
' Decode a data matrix without timing borders
DataMatrix1.FineTuneMethodEnabled _
    (hsMatrixNoTopTimingBorder) = True
DataMatrix1.FineTuneMethodEnabled _
    (hsMatrixNoRightTimingBorder) = True
DataMatrix1.Execute
```

GlobalThresholdEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **GlobalThresholdEnabled** is set to True, the tool activates the global threshold method. Three threshold methods are available: global (**GlobalThresholdEnabled**), local (**LocalThresholdEnabled**), and adaptive (**AdaptiveThresholdEnabled**). These three methods are not exclusive and can be enabled simultaneously. When reading high quality marks, the global method is sufficient. With codes where the spacing between dots is uneven, the local method is recommended. When the background of the matrix is uneven, the adaptive method may produce better results. The global method is usually faster than both the local and adaptive methods. For the local and adaptive methods to work efficiently, **AverageElementSize** and **SamplesPerCell** should both be set to a value higher than 5 and **WarpMethod** should be set to **hsMatrixWarpSlow**. This property is auto-configured after a successful **Learn**.

Syntax

control.GlobalThresholdEnabled [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The global threshold method is enabled.
False	The global threshold method is disabled.

Example

```
' Read the matrix using the global threshold method only
DataMatrix1.AdaptiveThresholdEnabled = False
```

```
DataMatrix1.GlobalThresholdEnabled = True
DataMatrix1.LocalThresholdEnabled = False
DataMatrix1.Execute
```

InputImage

Description

Name of the image on which the tool is applied.

Syntax

```
control.InputImage [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = DataMatrix1.InputImage
DataMatrix2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the image on which the tool is applied.

Syntax

```
control.InputImage [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = DataMatrix1.InputImage
DataMatrix2.InputImage = "image1"
```

InputView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

```
control.InputView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = DataMatrix1.InputView
DataMatrix2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.**InputView** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = DataMatrix1.InputView
DataMatrix2.InputView = "view0"
```

InputInstanceScene

Description

Name of the scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = DataMatrix1.InputInstanceScene
DataMatrix2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
```

```

ViewName = DataMatrix1.InputInstanceView
DataMatrix2.InputInstanceView = "view0"

```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

```
control.InputModelScene [= Name As String]
```

Type

String

Range

Not applicable.

Example

```

Dim ModelSceneName As String
ModelSceneName = DataMatrix1.InputModelScene
DataMatrix2.InputModelScene = "model_scene0"

```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details

Syntax

```
control.InputModelView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```

Dim ModelViewName As String
ModelViewName = DataMatrix1.InputModelView
DataMatrix2.InputModelView = "view0"

```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = DataMatrix1.InstanceSelectionEnabled
DataMatrix2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = DataMatrix1.InstanceSelectionIndex
DataMatrix2.InstanceSelectionIndex = 0
```

LocalThresholdEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **LocalThresholdEnabled** is set to True, the tool activates the local threshold method. Three threshold methods are available: global (**GlobalThresholdEnabled**), local (**LocalThresholdEnabled**), and adaptive (**AdaptiveThresholdEnabled**). These three methods are not exclusive and can be enabled simultaneously. When reading high quality marks, the global method is sufficient. With codes where the spacing between dots is uneven, the local method is recommended. When the background of the matrix is uneven, the adaptive method may produce better results. The global method is usually faster than both the local and adaptive methods. For the local and adaptive methods to work efficiently, **AverageElementSize** and **SamplesPerCell** should both be set to a value higher than 5 and **WarpMethod** should be set to **hsMatrixWarpSlow**. This property is auto-configured after a successful **Learn**.

Syntax

```
control.LocalThresholdEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	The local threshold method is enabled.
False	The local threshold method is disabled.

Example

```
' Read the matrix using the local and global threshold
' methods
DataMatrix1.AdaptiveThresholdEnabled = False
DataMatrix1.GlobalThresholdEnabled = True
DataMatrix1.LocalThresholdEnabled = True
DataMatrix1.Execute
```

MatrixAngle

Description

Angle of rotation of the data matrix instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the orientation units specified by the **UnitsOrientation** property. Read only.

Syntax

```
control.MatrixAngle(Index As Integer)
```

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	BarcodeAngle property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```
Dim MatrixAngle As Single
' Get the angle of the first data matrix in the calibrated
' World coordinate system
DataMatrix1.CoordinateSystem = hsCoordinateSystemWorld
MatrixAngle = DataMatrix1.MatrixAngle(0)
```

MatrixBorderMatchPercentage

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The percentage of border cells that are detected in a data matrix instance. The border match is 100% for a data matrix with perfect borders. Read only.

Syntax

control.**MatrixBorderMatchPercentage**(*Index* As Integer)

Type

Single

Range

Minimum: 75

Maximum: 100

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```
Dim BorderPercentage As Single
' Get the border match percentage of the first data matrix
BorderPercentage = _
    DataMatrix1.MatrixBorderMatchPercentage(0)
```

MatrixColumns

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required number of columns in the data matrix. The **MatrixColumns** property is ignored if **MatrixColumnsEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.**MatrixColumns** [= *Value* As Integer]

Type

Integer

Range

Minimum: 8

Maximum: 144

Example

```
' Decode a 18 x 18 cells data matrix
DataMatrix1.MatrixColumns = 18
```

```
DataMatrix1.MatrixColumnsEnabled = True
DataMatrix1.MatrixRows = 18
DataMatrix1.MatrixRowsEnabled = True
DataMatrix1.Execute
```

MatrixColumnsEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixColumnsEnabled** is set to True, a data matrix must have a number of columns equal to the **MatrixColumns** property to be correctly decoded. When set to false, a data matrix with any number of columns can be decoded.

Syntax

control.**MatrixColumnsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the data matrix must have a number of columns equal to the MatrixColumns property.
False	Data matrixes with any number of columns can be decoded.

Example

```
' Decode a 18 x 18 cells data matrix
DataMatrix1.MatrixColumns = 18
DataMatrix1.MatrixColumnsEnabled = True
DataMatrix1.MatrixRows = 18
DataMatrix1.MatrixRowsEnabled = True
DataMatrix1.Execute
```

MatrixContrast

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The percentage of contrast in a data matrix instance. A value of 100% indicates a contrast of 255. Read only.

Syntax

control.**MatrixContrast** (*Index* As Integer)

Type

Integer

Range

Minimum: 0

Maximum: 100

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```
Dim MatrixContrast As Integer
' Get the contrast of the first data matrix
MatrixContrast = DataMatrix1.MatrixContrast(0)
```

MatrixCornerX

Description

X coordinate of one of the 4 indexed corners of a given data matrix instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the length units specified by the **UnitsLength** property when the World, the Object or the Tool coordinate system is selected. Expressed in pixels when the Image coordinate system is selected. In the exceptional case where the tool is not configured to work with calibrated units (**CalibratedUnitsEnabled** and **ModelTransformEnabled** properties are both set to False), values returned in the Tool coordinate system are expressed in pixels, and values cannot be returned in the World or Object coordinate systems. Read only.

Syntax

control.**MatrixCornerX**(*Index* As Integer, *Corner* As Integer)

Type

Single

Range

Any value inside the boundaries of the selected coordinate system.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1
<i>Corner</i>	Index of the corner.	Integer	0,1,2,3

Example

```
' Draw a green box in the image around the first matrix
DataMatrix1.CoordinateSystem = hsCoordinateSystemImage
DrawLine DataMatrix1.MatrixCornerX(0,0), _
          DataMatrix1.MatrixCornerY(0,0), _
          DataMatrix1.MatrixCornerX(0,1), _
          DataMatrix1.MatrixCornerY(0,1), _
          Green
DrawLine DataMatrix1.MatrixCornerX(0,1), _
          DataMatrix1.MatrixCornerY(0,1), _
```

```

DataMatrix1.MatrixCornerX(0,2), _
DataMatrix1.MatrixCornerY(0,2), _
Green
DrawLine DataMatrix1.MatrixCornerX(0,2), _
DataMatrix1.MatrixCornerY(0,2), _
DataMatrix1.MatrixCornerX(0,3), _
DataMatrix1.MatrixCornerY(0,3), _
Green
DrawLine DataMatrix1.MatrixCornerX(0,3), _
DataMatrix1.MatrixCornerY(0,3), _
DataMatrix1.MatrixCornerX(0,0), _
DataMatrix1.MatrixCornerY(0,0), _
Green

```

MatrixCornerY

Description

Y coordinate of one of the 4 indexed corners of a given data matrix instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the length units specified by the **UnitsLength** property when the World, the Object or the Tool coordinate system is selected. Expressed in pixels when the Image coordinate system is selected. In the exceptional case where the tool is not configured to work with calibrated units (**CalibratedUnitsEnabled** and **ModelTransformEnabled** properties are both set to False), values returned in the Tool coordinate system are expressed in pixels, and values cannot be returned in the World or Object coordinate systems. Read only.

Syntax

control.**MatrixCornerY**(*Index* As Integer, *Corner* As Integer)

Type

Single

Range

Any value inside the boundaries of the selected coordinate system.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1
<i>Corner</i>	Index of the corner.	Integer	0,1,2,3

Example

```

' Draw a green box in the image around the first matrix
DataMatrix1.CoordinateSystem = hsCoordinateSystemImage
DrawLine DataMatrix1.MatrixCornerX(0,0), _
DataMatrix1.MatrixCornerY(0,0), _
DataMatrix1.MatrixCornerX(0,1), _
DataMatrix1.MatrixCornerY(0,1), _

```

```

        Green
DrawLine DataMatrix1.MatrixCornerX(0,1), _
        DataMatrix1.MatrixCornerY(0,1), _
        DataMatrix1.MatrixCornerX(0,2), _
        DataMatrix1.MatrixCornerY(0,2), _
        Green
DrawLine DataMatrix1.MatrixCornerX(0,2), _
        DataMatrix1.MatrixCornerY(0,2), _
        DataMatrix1.MatrixCornerX(0,3), _
        DataMatrix1.MatrixCornerY(0,3), _
        Green
DrawLine DataMatrix1.MatrixCornerX(0,3), _
        DataMatrix1.MatrixCornerY(0,3), _
        DataMatrix1.MatrixCornerX(0,0), _
        DataMatrix1.MatrixCornerY(0,0), _
        Green

```

MatrixDamagePercentage

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The percentage of damage in a data matrix instance. Damage percentage is calculated as the total number of error bits divided by the total number of bits in the matrix. Read only.

Syntax

control.**MatrixDamagePercentage**(*Index* As Integer)

Type

Single

Range

Minimum: 0

Maximum: 100

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```

Dim MatrixDamage As Single
' Get the damage of the first data matrix
MatrixDamage = DataMatrix1.MatrixDamagePercentage(0)

```

MatrixDecodeTime

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Total time of the decoding process. Read only.

Syntax

control.**MatrixDecodeTime**(*Index* As Integer)

Type

Single

Range

Greater than or equal to zero.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```
Dim DecodeTime As Single
' Get the time spent decoding the first data matrix
DecodeTime = DataMatrix1.MatrixDecodeTime(0)
```

MatrixECCLevel

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required ECC level of the data matrix to be decoded. The **MatrixECCLevel** property is ignored if **MatrixECCLevelEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.**MatrixECCLevel** [= *Value* As hsMatrixECCLevel]

Type

hsMatrixECCLevel

Range

The valid range for this property is as follows:

hsMatrixECCLevel	Value	Description
hsMatrixECC200	0	ECC 200 data matrix.
hsMatrixECC140	1	ECC 140 data matrix.
hsMatrixECC100	2	ECC 100 data matrix.
hsMatrixECC080	3	ECC 080 data matrix.
hsMatrixECC050	4	ECC 050 data matrix.

hsMatrixECCLevel	Value	Description
hsMatrixECC000	5	ECC 000 data matrix.

Example

```
' Decode a ECC 200 data matrix
DataMatrix1.MatrixECCLevel = hsMatrixECC200
DataMatrix1.MatrixECCLevelEnabled = True
DataMatrix1.Execute
```

MatrixECCLevelEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixECCLevelEnabled** is set to True, a data matrix must be of the ECC level specified by the **MatrixECCLevel** property to be correctly decoded. When set to false, a data matrix of any ECC level can be decoded.

Syntax

control.**MatrixECCLevelEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the ECC level configured by the MatrixECCLevel property.
False	Data matrixes of any ECC levels can be decoded.

Example

```
' Decode a ECC 200 data matrix
DataMatrix1.MatrixECCLevel = hsMatrixECC200
DataMatrix1.MatrixECCLevelEnabled = True
DataMatrix1.Execute
```

MatrixFoundCount

Description

Number of data matrixes actually found and decoded in the image. Read only.

Syntax

control.**MatrixFoundCount**

Type

Integer

Range

Minimum: 0

Maximum: **MatrixToDecode** -1

Example

```

Dim MatrixesToDecode As Integer
Dim MatrixesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim i As Integer
' Configure the tool to read 4 data matrixes in the image
DataMatrix1.MatrixToDecode = 4
DataMatrix1.Execute
...
' Get the results
MatrixesToDecode = DataMatrix1.MatrixToDecode
MatrixesFound = DataMatrix1.MatrixFoundCount
' Get the strings of the decoded data matrixes
For i = 0 To MatrixesFound - 1
    ResultString = DataMatrix1.MatrixString(i)
Next i
' Get the error code of the undecoded data matrixes
For i = MatrixesFound To MatrixesToDecode - 1
    ErrorCode = DataMatrix1.MatrixStatusCode(i)
Next i

```

MatrixHeight

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required height, in pixels, of the data matrix to be decoded. This height can vary by the percentage set by the **SizeVariation** property. The **MatrixHeight** property is ignored if **MatrixHeightEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.**MatrixHeight** [= *Value* As Integer]

Type

Integer

Range

Minimum: 20
Maximum: 1024

Example

```

' Decode a 100 x 100 pixels data matrix
DataMatrix1.MatrixHeight = 100
DataMatrix1.MatrixHeightEnabled = True
DataMatrix1.MatrixWidth = 100
DataMatrix1.MatrixWidthEnabled = True

```

```
DataMatrix1.SizeVariation = 10
DataMatrix1.Execute
```

MatrixHeightEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixHeightEnabled** is set to True, a data matrix must have a height that matches the **MatrixHeight** property to be correctly decoded. When set to false, a data matrix of any height can be decoded.

Syntax

```
control.MatrixHeightEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the height of the data matrix must match the MatrixHeight property.
False	Data matrixes of any height can be decoded.

Example

```
' Decode a 100 x 100 pixels data matrix
DataMatrix1.MatrixHeight = 100
DataMatrix1.MatrixHeightEnabled = True
DataMatrix1.MatrixWidth = 100
DataMatrix1.MatrixWidthEnabled = True
DataMatrix1.SizeVariation = 10
DataMatrix1.Execute
```

MatrixImageStyle

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required image style (normal or mirror) of the data matrix to be decoded. The **MatrixImageStyle** property is ignored if **MatrixImageStyleEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

```
control.MatrixImageStyle [= Value As hsMatrixImageStyle]
```

Type

hsMatrixImageStyle

Range

The valid range for this property is as follows:

hsMatrixImageStyle	Value	Description
hsMatrixImageStyleNormal	0	The image is not mirrored
hsMatrixImageStyleMirror	1	The image is mirrored

Example

```
' Decode a data matrix in a mirrored image
DataMatrix1.MatrixImageStyle = hsMatrixImageStyleMirror
DataMatrix1.MatrixImageStyleEnabled = True
DataMatrix1.Execute
```

MatrixImageStyleEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixImageStyleEnabled** is set to True, the image style of a data matrix must match the style specified by the **MatrixImageStyle** property to be correctly decoded. When set to false, a data matrix of any image style can be decoded.

Syntax

control.**MatrixImageStyleEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the image style of the data matrix must match the style specified by the MatrixImageStyleEnabled property.
False	Data matrixes of any image style can be decoded.

Example

```
' Decode a data matrix in a mirrored image
DataMatrix1.MatrixImageStyle = hsMatrixImageStyleMirror
DataMatrix1.MatrixImageStyleEnabled = True
DataMatrix1.Execute
```

MatrixOrientation

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required orientation, in the tool coordinate system, of the data matrix to be decoded. This orientation can vary by the percentage set by the **AngleVariation** property. The **MatrixOrientation** property is ignored if **MatrixOrientationEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.**MatrixOrientation** [= *Value* As hsMatrixOrientation]

Type

hsMatrixOrientation

Range

The valid range for this property is as follows:

hsMatrixImageStyle	Value	Description
hsMatrixOrientation0	0	The data matrix must be at 0 degrees
hsMatrixOrientation90	1	The data matrix must be at 90 degrees
hsMatrixOrientation180	2	The data matrix must be at 180 degrees
hsMatrixOrientation270	3	The data matrix must be at 270 degrees
hsMatrixOrientation45	4	The data matrix must be at 45 degrees
hsMatrixOrientation135	5	The data matrix must be at 135 degrees
hsMatrixOrientation225	6	The data matrix must be at 225 degrees
hsMatrixOrientation315	7	The data matrix must be at 315 degrees

Example

```
' Search for a data matrix at 45 degrees and allow a
' maximum variation of +/- 10 degrees from this angle
DataMatrix1.MatrixOrientation = hsMatrixOrientation45
DataMatrix1.MatrixOrientationEnabled = True
DataMatrix1.AngleVariation = 10
DataMatrix1.Execute
```

MatrixOrientationEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixOrientationEnabled** is set to True, the orientation of a data matrix must match the orientation specified by the **MatrixOrientation** property to be correctly decoded. When set to false, a data matrix at any orientation can be decoded.

Syntax

control.**MatrixOrientationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the orientation of the data matrix must match the orientation specified by the MatrixOrientation property.
False	Data matrixes at any orientation can be decoded.

Example

```
' Search for a data matrix at 45 degrees and allow a
' maximum variation of +/- 10 degrees from this angle
DataMatrix1.MatrixOrientation = hsMatrixOrientation45
DataMatrix1.MatrixOrientationEnabled = True
DataMatrix1.AngleVariation = 10
DataMatrix1.Execute
```

MatrixPolarity

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required polarity of the data matrix to be decoded. The **MatrixPolarity** property is ignored if **MatrixPolarityEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.**MatrixPolarity** [= *Value* As hsMatrixPolarity]

Type

hsMatrixPolarity

Range

The valid range for this property is as follows:

hsMatrixPolarity	Value	Description
hsMatrixPolarityLightOnDark	0	Light matrix on a dark background
hsMatrixPolarityDarkOnLight	1	Dark matrix on a light background.

Example

```
' Decode a light matrix on a dark background
DataMatrix1.MatrixPolarity = hsMatrixPolarityLightOnDark
DataMatrix1.MatrixPolarityEnabled = True
DataMatrix1.Execute
```

MatrixPolarityEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixPolarityEnabled** is set to True, the polarity of a data matrix must match the polarity specified by the **MatrixPolarity** property to be correctly decoded. When set to false, a data matrix of any polarity can be decoded.

Syntax

control.**MatrixPolarityEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the polarity of the data matrix must match the polarity specified by the MatrixPolarity property.
False	Data matrixes of any polarity can be decoded.

Example

```
' Decode a light matrix on a dark background
DataMatrix1.MatrixPolarity = hsMatrixPolarityLightOnDark
DataMatrix1.MatrixPolarityEnabled = True
DataMatrix1.Execute
```

MatrixRows

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required number of columns in the data matrix. The **MatrixRows** property is ignored if **MatrixRowsEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.**MatrixRows** [= *Value* As Integer]

Type

Integer

Range

Minimum: 8

Maximum: 144

Example

```
' Decode a 18 x 18 cells data matrix
DataMatrix1.MatrixColumns = 18
DataMatrix1.MatrixColumnsEnabled = True
DataMatrix1.MatrixRows = 18
DataMatrix1.MatrixRowsEnabled = True
DataMatrix1.Execute
```

MatrixRowsEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixRowsEnabled** is set to True, a data matrix must have a number of rows equal to the **MatrixRows** property to be correctly decoded. When set to false, a data matrix with any number of rows can be decoded.

Syntax

control.**MatrixRowsEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the data matrix must have a number of rows equal to the MatrixRows property.
False	Data matrixes with any number of rows can be decoded.

Example

```
' Decode a 18 x 18 cells data matrix
DataMatrix1.MatrixColumns = 18
DataMatrix1.MatrixColumnsEnabled = True
DataMatrix1.MatrixRows = 18
DataMatrix1.MatrixRowsEnabled = True
DataMatrix1.Execute
```

MatrixStatusCode

Description

Status code which reports on the success or failure of the decoding process on a given data matrix instance. Read only.

Syntax

control.**MatrixStatusCode**(Index As Integer)

Type

Long

Range

The valid range for this property is as follows:

Value	Description
0000	Success
1000	No Execute Performed
3048	Symbology Decode Failed

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```
Dim MatrixesToDecode As Integer
Dim MatrixesFound As Integer
Dim ResultString As String
```

```

Dim ErrorCode As Long
Dim i As Integer
' Configure the tool to read 4 matrixes in the image
DataMatrix1.MatrixToDecode = 4
DataMatrix1.Execute
...
' Get the results
MatrixesToDecode = DataMatrix1.MatrixToDecode
MatrixesFound = DataMatrix1.MatrixFoundCount
' Get the strings of the decoded matrixes
For i = 0 To MatrixesFound - 1
    ResultString = MatrixReader1.MatrixString(i)
Next i
' Get the error code of the undecoded matrixes
For i = MatrixesFound To MatrixesToDecode - 1
    ErrorCode = DataMatrix1.MatrixStatusCode(i)
    ErrorString = DataMatrix1.MatrixStatusCodeString(i)
Next i

```

MatrixStatusCodeString

Description

Status code message associated to the status code returned by the **MatrixStatusCode** property. Read only.

Syntax

control.**MatrixStatusCodeString**(*Index* As Integer)

Type

String

Range

Not applicable

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```

Dim MatrixesToDecode As Integer
Dim MatrixesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim i As Integer
' Configure the tool to read 4 matrixes in the image
DataMatrix1.MatrixToDecode = 4

```

```

DataMatrix1.Execute
...
' Get the results
MatrixesToDecode = DataMatrix1.MatrixToDecode
MatrixesFound = DataMatrix1.MatrixFoundCount
' Get the strings of the decoded matrixes
For i = 0 To MatrixesFound - 1
    ResultString = MatrixReader1.MatrixString(i)
Next i
' Get the error code of the undecoded matrixes
For i = MatrixesFound To MatrixesToDecode - 1
    ErrorCode = DataMatrix1.MatrixStatusCode(i)
    ErrorString = DataMatrix1.MatrixStatusCodeString(i)
Next i

```

MatrixString

Description

Data string decoded from a given data matrix instance. Read only.

Syntax

control.**MatrixString**(*Index* As Integer)

Type

String

Range

Not applicable

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```

Dim MatrixesToDecode As Integer
Dim MatrixesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim i As Integer
' Configure the tool to read 4 matrixes in the image
DataMatrix1.MatrixToDecode = 4
DataMatrix1.Execute
...
' Get the results
MatrixesToDecode = DataMatrix1.MatrixToDecode
MatrixesFound = DataMatrix1.MatrixFoundCount

```

```

' Get the strings of the decoded matrixes
For i = 0 To MatrixesFound - 1
    ResultString = MatrixReader1.MatrixString(i)
Next i
' Get the error code of the undecoded matrixes
For i = MatrixesFound To MatrixesToDecode - 1
    ErrorCode = DataMatrix1.MatrixStatusCode(i)
    ErrorString = DataMatrix1.MatrixStatusCodeString(i)

Next i

```

MatrixThreshold

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Threshold used to detect the cells in a given data matrix instance. Read only.

Syntax

control.**MatrixThreshold**(*Index* As Integer)

Type

Integer

Range

Minimum: 1

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```

Dim Threshold As Integer
' Get the threshold used to detect cells in the first data
' matrix
Threshold = DataMatrix1.MatrixThreshold(0)

```

MatrixToDecode

Description

Number of data matrix instances to decode in the image.

Syntax

control.**MatrixToDecode** [= *Value* As Integer]

Type

Integer

Range

Minimum:1

Maximum:5

Example

```

Dim MatrixesToDecode As Integer
Dim MatrixesFound As Integer
Dim ResultString As String
Dim ErrorCode As Long
Dim i As Integer
' Configure the tool to read 4 data matrixes in the image
DataMatrix1.MatrixToDecode = 4
DataMatrix1.Execute
...
' Get the results
MatrixesToDecode = DataMatrix1.MatrixToDecode
MatrixesFound = DataMatrix1.MatrixFoundCount
' Get the strings of the decoded data matrixes
For i = 0 To MatrixesFound - 1
    ResultString = DataMatrix1.MatrixString(i)
Next i
' Get the error code of the undecoded data matrixes
For i = MatrixesFound To MatrixesToDecode - 1
    ErrorCode = DataMatrix1.MatrixStatusCode(i)
Next i

```

MatrixTotalErrorBits

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Number of error bits found in a given data matrix instance. Read only

Syntax

control.**MatrixTotalErrorBits**(*Index* As Integer)

Type

Integer

Range

Minimum: 0

Maximum: Number of bits in the data matrix

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the data matrix instance.	Integer	Minimum: 0 Maximum: MatrixToDecode - 1

Example

```
Dim ErrorBits As Integer
' Get the number of error bits in the first data matrix
ErrorBits = DataMatrix1.MatrixErrorBits(0)
```

MatrixWidth

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required width, in pixels, of the data matrix to be decoded. This width can vary by the percentage set by the **SizeVariation** property. The **MatrixWidth** property is ignored if **MatrixWidthEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

```
control.MatrixWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: 20

Maximum: 1024

Example

```
' Decode a 100 x 100 pixels data matrix
DataMatrix1.MatrixHeight = 100
DataMatrix1.MatrixHeightEnabled = True
DataMatrix1.MatrixWidth = 100
DataMatrix1.MatrixWidthEnabled = True
DataMatrix1.SizeVariation = 10
DataMatrix1.Execute
```

MatrixWidthEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **MatrixWidthEnabled** is set to True, a data matrix must have a width that matches the **MatrixWidth** property to be correctly decoded. When set to false, a data matrix of any width can be decoded.

Syntax

```
control.MatrixWidthEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	To be correctly decoded, the width of the data matrix must match the MatrixWidth .
False	Data matrixes of any height can be decoded.

Example

```
' Decode a 100 x 100 pixels data matrix
DataMatrix1.MatrixHeight = 100
DataMatrix1.MatrixHeightEnabled = True
DataMatrix1.MatrixWidth = 100
DataMatrix1.MatrixWidthEnabled = True
DataMatrix1.SizeVariation = 10
DataMatrix1.Execute
```

MinimumEdgeStrength

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The minimum contrast strength required for a matrix edge to be detected in the image. This property is auto-configured after a successful **Learn**.

Syntax

control.**MinimumEdgeStrength** [= *Value* As Integer]

Type

Integer

Range

Minimum: 5

Maximum: 100

Example

```
Dim MinEdgeStrength As Integer
' Get the current minimum edge strength
MinEdgeStrength = DataMatrix1.MinimumEdgeStrength
' Set the minimum edge strength
DataMatrix1.MinimumEdgeStrength = 20
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = DataMatrix1.ModelTransformEnabled
DataMatrix2.ModelTransformEnabled = True
```

Name**Description**

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = DataMatrix1.Name
DataMatrix2.Name = "EdgeLocA"
```

ProcessingAlgorithm**Description**

Activates one of the optional enhancement preprocessing algorithms that can be applied to the input image prior to the decoding process. The selected preprocessing algorithm is also applied prior to a **Learn**.

Syntax

control.**ProcessingAlgorithm** [= Value As hsMatrixProcessing]

Type

hsMatrixProcessing

Range

The valid range for this property is as follows:

hsMatrixProcessing	Value	Description
hsMatrixProcessingNone	0	No pre-processing is applied
hsMatrixProcessingInvert	1	Reverses the polarity of the image
hsMatrixProcessingErode	2	Erodes light objects
hsMatrixProcessingDilate	3	Dilates light objects

hsMatrixProcessing	Value	Description
hsMatrixProcessingOpen	4	Removes small light defects
hsMatrixProcessingClose	5	Removes small dark defects
hsMatrixProcessingGaussian	6	Smooths the edges in the image
hsMatrixProcessingSharpen	7	Enhances the edges in the image
hsMatrixProcessingSharpenLow	8	Enhances the edges in the image after a low-pass filter is applied
hsMatrixProcessingMedian	9	Removes noise with a median filter
hsMatrixProcessingHistogramStretching	10	Stretches the histogram to increase contrast

Example

```
' Filter out high frequency noise in the image prior to
' reading the code
DataMatrix1.ProcessingAlgorithm = _
    hsMatrixProcessingMedian
DataMatrix1.ProcessingSize = 5
DataMatrix1.Execute
```

ProcessingSize

Description

Sets the **Size** parameter that is used by the preprocessing algorithm selected by the **ProcessingAlgorithm** property.

Syntax

control.**ProcessingSize** [= Value As Integer]

Type

Integer

Range

The range depends on the value of **ProcessingAlgorithm**:

hsMatrixProcessing	Range	Description
hsMatrixProcessingNone	Not applicable.	Not applicable.
hsMatrixProcessingInvert	Not applicable.	Not applicable.
hsMatrixProcessingErode	Greater than 0	Number of iterations
hsMatrixProcessingDilate	Greater than 0	Number of iterations
hsMatrixProcessingOpen	Greater than 0	Number of iterations
hsMatrixProcessingClose	Greater than 0	Number of iterations
hsMatrixProcessingGaussian	3,5,7	Kernel size
hsMatrixProcessingSharpen	3,5,7	Kernel size
hsMatrixProcessingSharpenLow	3,5,7	Kernel size
hsMatrixProcessingMedian	3,5,7	Kernel size
hsMatrixProcessingHistogramStretching	Not applicable.	Not applicable.

Example

```
' Filter out high frequency noise in the image prior to
' reading the code
DataMatrix1.ProcessingAlgorithm = _
    hsMatrixProcessingMedian
DataMatrix1.ProcessingSize = 5
DataMatrix1.Execute
```

SamplesPerCell

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The number of samples taken for each matrix cell. The tool samples and averages portions of each cell to determine whether the cell is light or dark. This property is auto-configured after a successful **Learn**.

Syntax

control.control.SamplesPerCell [= *Value* As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 7

Example

```
' Decode the data matrix using 7 samples per cell
DataMatrix1.SamplesPerCell = 7
DataMatrix1.Execute
```

SamplingStep

Description

Step used in the tool's last execution to sample the region of interest from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.SamplingStep

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = DataMatrix1.SamplingStep
```

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = DataMatrix1.SamplingStepCustom
DataMatrix2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Example

```
Dim Enabled As Boolean
Enabled = DataMatrix1.SamplingStepCustomEnabled
DataMatrix2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

control.**SamplingStepDefault**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = DataMatrix1.SamplingStepDefault
```

SearchDirection

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Sets the direction used by the tool to scan the image for data matrix instances.

Syntax

control.**SearchDirection** [= *Value* As hsMatrixSearchDirection]

Type

hsMatrixSearchDirection

Range

The valid range for this property is as follows:

hsMatrixSearchDirection	Value	Description
hsMatrixDirectionHorizontal	0	Data matrix scanned horizontally only.
hsMatrixDirectionVertical	1	Data matrix scanned vertically only.
hsMatrixDirectionHorizontalVertical	2	Data matrix scanned horizontally then vertically.
hsMatrixDirectionVerticalHorizontal	3	Data matrix scanned vertically then horizontally.

Example

```
' Search for horizontal and vertical data matrix instances
DataMatrix1.SearchDirection = _
    hsMatrixDirectionHorizontalVertical
DataMatrix1.Execute
```

SearchSpeed

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Search speed used for locating data matrix instances in the image.

Syntax

control.**SearchSpeed** [= *Value* As hsMatrixSearchSpeed]

Type

hsMatrixSearchSpeed

Range

The valid range for this property is as follows:

hsSearchSpeed	Value	Description
hsMatrixSpeedNormal	0	Normal search speed. The most robust.
hsMatrixSpeedOverdrive	1	Faster than normal speed.
hsMatrixSpeedTurbo	2	Fastest search speed, Use only with high contrast high quality data matrix images.

Example

```
' Use the turbo search mode for finding the data matrix
DataMatrix1.SearchSpeed = hsMatrixSpeedTurbo
DataMatrix1.Execute
```

SizeVariation

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Percentage of allowable variation of the required data matrix height and width set by the **MatrixHeight** and **MatrixWidth** properties.

Syntax

control.**SizeVariation** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 10

Example

```
' Decode a 100 x 100 pixels data matrix
DataMatrix1.MatrixHeight = 100
DataMatrix1.MatrixHeightEnabled = True
DataMatrix1.MatrixWidth = 100
DataMatrix1.MatrixWidthEnabled = True
DataMatrix1.SizeVariation = 10
DataMatrix1.Execute
```


Timeout

Description

Timeout, in milliseconds, after which the tool aborts its process.

Syntax

control.Timeout [= Value As Long]

Type

Long

Range

Greater than zero.

Example

```
Dim Timeout As Integer
Timeout = DataMatrix1.Timeout
DataMatrix2.Timeout = 250
```

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.ToolHeight [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = DataMatrix1.ToolHeight
DataMatrix2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.ToolPositionX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = DataMatrix1.ToolPositionX
DataMatrix2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = DataMatrix1.ToolPositionY
DataMatrix2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = DataMatrix1.ToolRotation
DataMatrix2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

```
control.ToolWidth [= Value As Single]
```

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = DataMatrix1.ToolWidth
DataMatrix2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = DataMatrix1.TransformRotation
DataMatrix2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = DataMatrix1.TransformScale
```

`DataMatrix2.TransformScale = 1.10`

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transform is used to fit the tool to an object in the input image. The components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the

InstanceSelectionEnabled property is set to True.

Syntax

`control.TransformTranslationX [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = DataMatrix1.TransformTranslationX
DataMatrix2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transform is used to fit the tool to an object in the input image. The components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the

InstanceSelectionEnabled property is set to True.

Syntax

`control.TransformTranslationY [= Value As Single]`

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = DataMatrix1.TransformTranslationY
DataMatrix2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSDataMatrixReader".

Example

```
Dim ControlType As String
ControlType = DataMatrix1.TypeName
```

UnitsLength

Description

Unit used to express lengths and coordinates.

Syntax

control.UnitsLength [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = DataMatrix1.UnitsLength
DataMatrix2.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = DataMatrix1.UnitsOrientation
DataMatrix2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.2.000 R 3217 (release)".

Example

```
Dim Version As String
Version = DataMatrix1.Version
```

WarpMethod

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Warp method used to read data matrixes rotated within the tool coordinate system. The slow method increases the robustness on very degraded images, while the fast method usually represents the best speed/robustness compromise on less degraded images.

Syntax

control.**WarpMethod** [= *Value* As hsMatrixWarpMethod]

Type

hsMatrixWarpMethod

Range

The valid range for this property is as follows:

hsMatrixWarpMethod	Value	Description
hsMatrixWarpSlow	0	Matrixes rotated within the tool coordinate system are warped using the slower robust method.
hsMatrixWarpFast	1	Matrixes rotated within the tool coordinate system are warped using the faster method.

Example

```
' Decode a highly degraded data matrix at 45 degrees in
' the tool coordinate system
DataMatrix1.MatrixOrientation = hsMatrixOrientation45
DataMatrix1.MatrixOrientationEnabled = True
DataMatrix1.AngleVariation = 10
DataMatrix1.WarpMethod = hsMatrixWarpSlow
DataMatrix1.Execute
```

WidthHeightRatio

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

The required width/height ratio of the data matrix to be decoded. This property is ignored when both the required height and width are specified using the **MatrixWidth** and **MatrixHeight** properties. **WidthHeightRatio** is ignored if **WidthHeightRatioEnabled** is set to False. This property is auto-configured after a successful **Learn**.

Syntax

control.WidthHeightRatio [= *Value* As Single]

Type

Single

Range

Minimum: 0.3
Maximum: 5.0

Example

```
' Decode a 2:1 aspect ratio data matrix of any size
DataMatrix1.WidthHeightRatio = 2
DataMatrix1.WidthHeightRatioEnabled = True
DataMatrix1.Execute
```

WidthHeightRatioEnabled

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

When **WidthHeightRatioEnabled** is set to True, the width/height ratio of a data matrix must match the ratio specified by the **WidthHeightRatio** property to be correctly decoded. When set to false, a data matrix with any width/height ratio can be decoded.

Syntax

control.WidthHeightRatioEnabled [= *Value* As Boolean]

Type

Boolean

Range:

Value	Description
True	To be correctly decoded, the width/height ratio of the data matrix must match the ratio specified by the WidthHeightRatio property.
False	Data matrixes with any width/height ratio can be decoded.

Example

```
' Decode a 2:1 aspect ratio data matrix of any size
DataMatrix1.WidthHeightRatio = 2
DataMatrix1.WidthHeightRatioEnabled = True
DataMatrix1.Execute
```

Methods

AboutBox

Description

Displays the tool About box.

Syntax

control.AboutBox()

Parameter(s)

None

Range

No return value.

Example

```
DataMatrix1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.Syntax
control.Execute()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example`DataMatrix1.Execute`

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideLearnInterface

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Hides and terminates the Learn dialog.

Syntax`control.HideLearnInterface()`**Parameter(s)**

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example`DataMatrix.HideLearnInterface`

HideProperties

Description

Hides the property pages of the tool if they are displayed.

Syntax`control.HideProperties()`**Parameter(s)**

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = DataMatrix1.HideProperties
```

Learn**Warning**

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Auto-detects the type of data matrix currently present in the image and optimizes the detection parameters accordingly. After a successful call to **Learn** (True), the following detection parameters are optimized: **AdaptiveThresholdEnabled**, **AverageElementSize**, **FineTuneMethodEnabled**, **GlobalThresholdEnabled**, **LocalThresholdEnabled**, **MatrixColumns**, **MatrixRows**, **MatrixECCLevel**, **MatrixHeight**, **MatrixWidth**, **MatrixImageStyle**, **MatrixOrientation**, **MatrixPolarity**, **MinimumEdgeStrength**, **SamplesPerCell**, and **WidthHeightRatio**. After completion of the learn process, the status can be retrieved with the **MatrixStatusCode** property.

Syntax

```
control.Learn([LearnBoundary])
```

Parameter(s)

Name	Description	Type	Range
LearnBoundary	An array of 4 (x,y) coordinates which define the boundary of the data matrix to learn. The boundary is always expressed in pixels, with respect to the image coordinate system. The coordinates should be placed in the array as follows: x0, y0, x1, y1, x2, y2, x3, y3. Optional.	Variant	Array of 8 integers, long, single, or double values

Return

Type	Description	Range
Boolean	Status of the Learn process	True: Successful False: Failed

Example

```
Dim SuccessfulLearn As Boolean
Dim Boundary(8) as Integer
Dim ErrorCode As Long
' Set the matrix boundary to learn to
' (100,200) (100,300) (200,300) (200,200)
Boundary(0)=100
Boundary(1)=200
Boundary(2)=100
Boundary(3)=300
Boundary(4)=200
Boundary(5)=300
```

```

Boundary(6)=200
Boundary(7)=200
SuccessfulLearn = DataMatrix1.Learn(Boundary)
If Not SuccessfulLearn Then
    ErrorCode = DataMatrix1.MatrixStatusCode(0)
EndIf

```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowLearnInterface

Warning

This function has been DEPRECATED. Relying on this feature is highly discouraged and results are unknown.

Description

Shows the Learn dialog. After completion of the learn process, the status can be retrieved with the **MatrixStatusCode** property.

Syntax

control.**ShowLearnInterface**(*Modal* As Boolean)

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```

Dim ErrorCode As Long
Dim ErrorString As String
DataMatrix1.ShowLearnInterface(True)
ErrorCode = DataMatrix1.MatrixStatusCode(0)

```

```
ErrorString = DataMatrix1.MatrixStatusCodeString(0)
```

ShowProperties

Description

Shows the property pages of the tool.

Syntax

```
control.ShowProperties(Modal As Boolean, [Variant Option])
```

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the table below.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = DataMatrix1.ShowProperties(True, _
    hsShowOK + hsShowExecute)
```

Events

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Display

Description

The Display Control allows the developer to visualize database objects used by HexSight applications. These databases are typically made up of multiple Views, each containing a set of Images, Scenes and/or Entities.

Properties

Appearance

Description

Paint style of the control at runtime. Read only at runtime.

Syntax

control.**Appearance**

Type

Integer

Range

Value	Description
0	Flat
1	3D

Example

```
Dim PaintStyle As Integer
PaintStyle = Display1.Appearance
```

AutoPanEnabled

Description

When **AutoPanEnabled** is set to true, the display content is automatically panned when a marker is moved outside the visible area of the display.

Syntax

control.**AutoPanEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	AutoPan is enabled. Default.
False	AutoPan is disabled.

Example

```
' Disable Auto Pan
Display2.AutoPanEnabled = False
```

AxesMarkerConstraints

Description

Constraints that apply to an Axes marker.

Syntax

control.**AxesMarkerConstraints**(*Name* As String) [*Value* As hsAxesMarkerConstraints]

Type

hsAxesMarkerConstraints

Range

Constant	Value	Description
hsAxesNoConstraints	0	No constraint.
hsAxesNoEdit	1	Edition disabled.
hsAxesNoRefresh	2	Automatic refresh disabled. If the marker is modified interactively using the mouse, the Display control is not refreshed automatically.
hsAxesNoMove	16	Move disabled.
hsAxesNoRotation	32	Rotation disabled.
hsAxesNoMidAngle	64	MidAngle modification disabled.
hsAxesFixedSize	128	Fixed size mode enabled.
hsAxesNoLength1	512	Length1 modification disabled.
hsAxesNoLength2	1024	Length2 modification disabled.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker	String	Not applicable.

Remark(s)

By default this property is set to hsNoEdit.

Example

```
Dim Constraint As hsAxesMarkerConstraints
Constraint = Display1.AxesMarkerConstraints
Display1.AxesMarkerConstraints = hsAxeNoMove +_
    hsAxesFixedSize
```

AxesMarkerLabel1

Description

Label of the first axis of an Axes marker.

Syntax

control.**AxesMarkerLabel1**(*Name* As String) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker	String	Not applicable.

Example

```
Dim LabelName As String
LabelName = Display1.AxesMarkerLabel1("AxesMarker1")
Display1.AxesMarkerLabel1("AxesMarker1") = "X"
```

AxesMarkerLabel2

Description

Label of the second axis of an Axes marker.

Syntax

control.AxesMarkerLabel2(*Name* As String) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
Dim LabelName As String
LabelName = Display1.AxesMarkerLabel2("AxesMarker1")
Display1.AxesMarkerLabel2("AxesMarker1") = "Y"
```

AxesMarkerLength1

Description

Length of the first axis of an Axes marker.

Syntax

control.AxesMarkerLength1(*Name* As String) [= *Value* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The length unit used to expressed this property depends on the status of the **MarkerCalibrated** property as shown in the table below:

MarkerCalibrated property	Length unit used
False	Pixel.
True	As defined in UnitsLength property.

Example

```
Dim Radius As Single
Radius = Display1.AxesMarkerLength1("MyMarker1")
Display1.AxesMarkerLength1("MyMarker1") = 50.0
```

AxesMarkerLength2

Description

Length of the second axis of an Axes marker.

Syntax

control.AxesMarkerLength2(*Name* As String) [*Value* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The length unit used to expressed this property depends on the status of the **MarkerCalibrated** property as shown in the following table:

MarkerCalibrated property	Length unit used
False	Pixel.
True	As defined in UnitsLength property.

Example

```
Dim Radius As Single
Radius = Display1.AxesMarkerLength2("MyMarker")
Display1.AxesMarkerLength2("MyMarker") = 50.0
```

AxesMarkerMidAngle

Description

Angle between the two axes of an Axes marker.

Syntax

control.**AxesMarkerMidAngle**(Name As String) [= Value As Single]

Type

Single

Range

This range of this property depends on the **UnitsOrientation** property.

UnitsOrientation property	Range of AxesMarkerMidAngle property
hsDegree	Minimum: -180 Maximum: +180
hsRadian	Minimum: -Pi Maximum: +Pi

Parameter(s)

Parameter(s)	Description	Type	Range
Name	Name of the marker.	String	Not applicable.

Example

```
Dim Angle As Single
```

```
Angle = Display1.AxesMarkerMidAngle("AxesMarker1")
```

```
Display1.AxesMarkerMidAngle("AxesMarker1") = 90.0
```

AxesMarkerRotation

Description

Angle of rotation of an Axes marker.

Syntax

control.**AxesMarkerRotation**(Name As String) [= Value As Single]

Type

Single

Range

This range of this property depends on the **UnitsOrientation** property.

UnitsOrientation property	Range of AxesMarkerMidAngle property
hsDegree	Minimum: -180 Maximum: +180
hsRadian	Minimum: -Pi Maximum: +Pi

Parameter(s)

Parameter(s)	Description	Type	Range
Name	Marker name.	String	Not applicable.

Remark(s)

The angle between the X-axis of a reference coordinate system and the first axis of the marker, going counter clockwise from the former to the latter. The reference coordinate system can be the Image or the World coordinate system depending on the status of the **MarkerCalibrated** property.

MarkerCalibrated property	Reference coordinate system
False	Image
True	World

Example

```
Dim Angle As Single
Angle = Display1.AxesMarkerRotation("MyMarker")
Display1.AxesMarkerRotation("MyMarker") = 0.0
```

BorderStyle

Description

Border style of the control. Read only at runtime.

Syntax

control.**BorderStyle** [= Value As Integer]

Type

Integer

Range

Value	Description
0	No border.
1	Fixed single border.

Example

```
Dim Border As Integer
Border = Display1.BorderStyle
```

Brightness

Description

Adjusts the brightness of images shown in the display. This property only modifies the appearance of the image in the display. It does not modify the actual greylevel values of the image. The **Contrast** property can also be used to adjust the contrast of images shown in the display.

Syntax

control.**Brightness** [= Value As Integer]

Type

Integer

Range

Minimum: -127 (Minimum brightness)

Maximum: 127 (Maximum brightness)

Example

```

If ImageIsTooDark Then
    ' Increase brightness of the display
    Display1.Brightness = 20
ElseIf ImageIsTooBright Then
    ' Decrease brightness of the display
    Display1.Brightness = -20
Else
    ' Use normal brightness
    Display1.Brightness = 0
End If

```

CalibratedUnitsEnabled

Description

Specifies if the control uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Example

```

Dim Result As Boolean
Result = Display1.CalibratedUnitsEnabled
Display2.CalibratedUnitsEnabled = False

```

CircularArcMarkerConstraints

Description

Constraints that apply to a Circular Arc marker.

Syntax

control.**CircularArcMarkerConstraints**(*Name* As String) [= *Value* As hsCircularArcMarkerConstraints]

Type

hsCircularArcMarkerConstraints

Range

Constant	Value	Description
hsCircularArcNoConstraints	0	No constraint.
hsCircularArcNoEdit	1	Edition disabled.

Constant	Value	Description
hsCircularArcNoRefresh	2	Automatic refresh disabled. If the marker is modified interactively using the mouse, the Display control is not refreshed automatically.
hsCircularArcNoMove	16	Move disabled.
hsCircularArcNoRotation	32	Rotation disabled.
hsCircularArcNoOpening	64	Arc Opening disabled.
hsCircularArcNoRadius	512	Arc Radius disabled.
hsCircularArcNoThickness	1024	Arc Thickness disabled.
hsCircularArcNoProjection	2048	Projection disabled.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```

Dim Constraint As hsCircularArcMarkerConstraints
Constraint = _
    Display1.CircularArcMarkerConstraints("MyMarker")
Display1.CircularArcMarkerConstraints("MyMarker") = _
    hsCircularArcNoMove + hsCircularArcNoRefresh

```

CircularArcMarkerOpening

Description

Opening angle of a Circular Arc marker.

Syntax

control.**CircularArcMarkerOpening**(*Name* As String) [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	CircularArcMarkerOpening property range
hsDegree	Minimum: -360.0 deg Maximum: +360.0 deg
hdRadian	Minimum: -2Pi rad Maximum: +2Pi rad

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
Dim Value As Single
Value = Display1.CircularArcMarkerOpening("MyMarker")
Display1.CircularArcMarkerOpening("MyMarker") = 20.0
```

CircularArcMarkerRadius

Description

Radius of a Circular Arc marker.

Syntax

control.**CircularArcMarkerRadius**(Name As String) [= Value As Single]

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The length unit used to expressed this property depends on the status of the **MarkerCalibrated** property as shown in the following table:

MarkerCalibrated property	Length unit used
False	Pixel.
True	As defined in UnitsLength property.

Example

```
Dim Value As Single
Value = Display1.CircularArcMarkerRadius("MyMarker")
Display1.CircularArcMarkerRadius("MyMarker") = 20.0
```

CircularArcMarkerRotation

Description

Angle of rotation of a Circular Arc marker.

Syntax

control.**CircularArcMarkerRotation**(Name As String) [= Value As Single]

Type

Single

Range

This range of this property depends on the **UnitsOrientation** property.

UnitsOrientation property	Range of AxesMarkerMidAngle property
hsDegree	Minimum: -180 Maximum: +180
hsRadian	Minimum: -Pi Maximum: +Pi

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Parameter(s)

The angle between the X-axis of a reference coordinate system and the first axis of the marker, going counter clockwise from the former to the latter. The reference coordinate system can be the Image or the World coordinate system depending on the status of the **MarkerCalibrated** property.

MarkerCalibrated property	Reference coordinate system
False	Image
True	World

Example

```
Dim Angle As Single
Angle = Display1.CircularArcMarkerRotation("MyMarker1")
Display1.CircularArcMarkerRotation("MyMarker1") = 0.0
```

CircularArcMarkerThickness

Description

Radial distance between the two bounding annuli of a Circular Arc marker.

Syntax

```
control.CircularArcMarkerThickness(Name As String) [= Value As Single]
```

Type

Single

Range

Any positive Single value.

Example

```
Dim Value As Single
Value = Display1.CircularArcMarkerThickness("MyMarker")
Display1.CircularArcMarkerThickness("MyMarker") = 35.0
```

ColorBackground

Description

Background color of the display.

Syntax

```
control.ColorBackground [= Value As hsColor]
```


Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Example

```
Dim MyColor As OLE_COLOR
MyColor = Display1.ColorBackground
Display1.ColorBackground = hsGreen
Display1.ColorBackground = RGB(128, 0, 128)
```

ColorForeground

Description

Foreground color of the display.

Syntax

control.**ColorForeground** [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Example

```
Dim MyColor As OLE_COLOR
MyColor = Display1.ColorForeground
Display1.ColorForeground = hsGreen
Display1.ColorForeground = RGB(128,0,128)
```

ColorGrid

Description

Color of the gridlines in the display.

Syntax

control.**ColorGrid** [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Example

```
Dim MyColor As OLE_COLOR
MyColor = Display1.ColorGrid
Display1.ColorGrid = hsGreen
Display1.ColorGrid = RGB(128,0,128)
```

ContextMenuEnabled

Description

Specifies if the context menu is accessible.

Syntax

control.**ContextMenuEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The context menu is not accessible.
True	The context menu is accessible.

Example

```
Dim Enabled As Boolean
Enabled = Display1.ContextMenuEnabled
Display1.ContextMenuEnabled = False
```

ContextMenuMask

Description

Specifies which of the items of the context menu are enabled.

Syntax

```
control.ContextMenuMask [= Value As hsContextMenuMask]
```

Type

hsContextMenuMask

Range

Constant	Value	Description
hsAllEnabled	255	All context menu items enabled.
hsZoom	1	Zoom related menu items enabled.
hsMode	2	Mode related menu items enabled.
hsCalibrate	4	Calibration related menu items enabled.
hsProperties	8	Properties related menu items enabled.

Example

```
Dim Mask As hsContextMenuEnabled
Mask = Display1.ContextMenuMask
Display2.ContextMenuMask = hsZoom + hsMode
```

Contrast

Description

Adjusts the contrast of images shown in the display. This property only modifies the appearance of the image in the display. It does not modify the actual greylevel values of the image. The **Brightness** property can also be used to adjust the brightness of images shown in the display.

Syntax

```
control.Contrast [= Value As Integer]
```

Type

Integer

Range

Minimum: -127 (Minimum contrast)
Maximum: 127 (Maximum contrast)

Example

```

If WeakContrast Then
    ' Increase contrast of the display
    Display1.Contrast = 20
ElseIf StrongContrast Then
    ' Decrease contrast of the display
    Display1.Contrast = -20
Else
    ' Use normal contrast
    Display1.Contrast = 0
End If

```

CrossHairVisible

Description

Specifies if the crosshair cursor is enabled.

Syntax

control.**CrossHairVisible** [= Value As Boolean]

Type

Boolean

Range

Value	Description
False	Crosshair cursor is disabled.
True	Crosshair cursor is enabled.

Example

```

Dim Visible As Boolean
Visible = Display1.CrossHairVisible
Display1.CrossHairVisible = False

```

Description

Description

String that describes the standard usage of the Display control. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```

Dim Description As String
Description = Display1.Description

```

EntityColor

Description

The color used to draw the entity.

Syntax

control.**EntityColor**(*Index* As Integer) [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the entity.	Integer	Minimum: 0 Maximum: EntitySize (default: 50)

Example

```
' Display the detected Outline contours in magenta
Display1.EntityColor(0) = hsMagenta
Display1.EntityPenWidth(0) = hsPenWidthThin
Display1.EntityDatabase(0) = Application1.Database.Handle
Display1.EntityViewName(0) = Locator.OutputOutlineSceneView
Display1.EntityName(0) = Locator.OutputOutlineScene
' Highlight edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of
' the Outline contours in red
Display1.EntityColor(0) = hsRed
Display1.EntityPenWidth(0) = hsPenWidthThin
Display1.EntitySelectionType(0) = hsTypeEdgePath
Display1.EntitySelectionIndex(0) = 0
Display1.EntitySelection(0) = "2:5,7:9,12,15"
```

EntityDatabase

Description

The handle to the database containing the entity to display.

Syntax

control.**EntityDatabase**(*Index* As Integer) [= *Value* As OLE_HANDLE]

Type

OLE_HANDLE

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the entity.	Integer	Minimum: 0 Maximum: EntitySize (default: 50)

Example

```
' Display the detected Outline contours in magenta
Display1.EntityColor(0) = hsMagenta
Display1.EntityPenWidth(0) = hsPenWidthThin
Display1.EntityDatabase(0) = Application1.Database.Handle
Display1.EntityViewName(0) = Locator.OutputOutlineSceneView
Display1.EntityName(0) = Locator.OutputOutlineScene
```

EntityDatabasePassword

Description

Reserved for internal use.

EntityKValues

Description

Reserved for internal use.

EntityName

Description

The name of the entity to display.

Syntax

control.**EntityName**(*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the entity.	Integer	Minimum: 0 Maximum: EntitySize (default: 50)

Example

```
' Display the detected Outline contours in magenta
Display1.EntityColor(0) = hsMagenta
Display1.EntityPenWidth(0) = hsPenWidthThin
Display1.EntityDatabase(0) = Application1.Database.Handle
Display1.EntityViewName(0) = Locator.OutputOutlineSceneView
Display1.EntityName(0) = Locator.OutputOutlineScene
```

EntityPenWidth

Description

The width of the pen used to draw the entity to display.

Syntax

control.**EntityPenWidth**(Index As Integer) [= Value As hsPenWidth]

Type

hsPenWidth

Range

Constant	Value	Description
hsPenWidthNone	0	Not drawn.
hsPenWidthThin	1	Drawn using thin lines.
hsPenWidthThick	2	Drawn using thick lines.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the entity.	Integer	Minimum: 0 Maximum: EntitySize (default: 50)

Example

```
' Display the detected Outline contours in magenta
Display1.EntityColor(0) = hsMagenta
Display1.EntityPenWidth(0) = hsPenWidthThin
Display1.EntityDatabase(0) = Application1.Database.Handle
Display1.EntityViewName(0) = Locator.OutputOutlineSceneView
Display1.EntityName(0) = Locator.OutputOutlineScene
```

EntitySelectionEnabled

Description

Enables Entities selection. See also **EntitySelectionChange** event.

Syntax

control.**EntitySelectionEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Entity Selection is enabled
False	Entity Selection is disabled

Example

```
Dim Result As Boolean
Result = Display1.EntitySelectionEnabled
Display2.EntitySelectionEnabled = False
```

EntitySize

Description

Maximum number of entities to be displayed.

Syntax

control.**EntitySize**[= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 1000, Default: 50

Example

```
Dim EntitiesToDisplay an Integer
EntitiesToDisplay = Display1.EntitySize
Display2.EntitySize = 20
```

EntityViewName

Description

The name of the database view containing the entity to display.

Syntax

control.**EntityViewName**(*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the entity.	Integer	Minimum: 0 Maximum: EntitySize (default: 50)

Example

```
' Set the output of the Acquisition Device as the image to
' show at layer 0 in the display
```

```
Set AcqDev1 = _
    Application1.ProcessManager.Process("Acquisition")
Display1.EntityDatabase(0) = Application1.Database.Handle
Display1.EntityViewName(0) = AcqDev1.OutputView
Display1.EntityName(0) = AcqDev1.OutputImage
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

FocusX

Description

Horizontal position of the focus point in the image.

Syntax

control.**FocusX** [= *Value* As Single]

Type

Single

Range

Boundaries of the image.

Remark(s)

If the **CalibratedUnitsEnabled** property is set to True then the **FocusX** property is expressed in calibrated units. Otherwise, it is expressed in pixels.

Example

```
Dim HorizPos As Single
HorizPos = Display1.FocusX
Display1.FocusX = 100.2
```

FocusY

Description

Vertical position of the focus point in the image.

Syntax

control.**FocusY** [= *Value* As Single]

Type

Single

Range

Boundaries of the image.

Remark(s)

If the **CalibratedUnitsEnabled** property is set to True then the **FocusY** property is expressed in calibrated units. Otherwise, it is expressed in pixels.

Example

```
Dim VertPos As Single
VertPos = Display1.FocusY
Display1.FocusY = 100
```

Font

Description

Selects the font used by the display control when displaying strings.

Syntax

control.**Font** [= *Value* As StdFont]

Type

StdFont

Range

Not applicable.

Example

```
Dim MyFont As StdFont
Set MyFont = Display1.Font
MyFont.Name = "courier"
Set Display1.Font = MyFont
```

GridVisible

Description

Specifies if the gridlines are visible in the display.

Syntax

control.**GridVisible** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The grid is not visible.
True	The grid is visible.

Example

```
Dim Visible As Boolean
Visible = Display1.GridVisible
Display1.GridVisible = False
```

ImageDatabase

Description

The handle to the database containing the image to display at a given layer in the display.

Syntax

control.**ImageDatabase**(*Index* As Integer) [= *Value* As OLE_HANDLE]

Type

OLE_HANDLE

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Set the output of the Acquisition Device as the image to
' show at layer 0 in the display
```

```
Set AcqDev1 = _
    Application1.ProcessManager.Process("Acquisition")
Display1.ImageDatabase(0) = Application1.Database.Handle
Display1.ImageViewName(0) = AcqDev1.OutputView
Display1.ImageName(0) = AcqDev1.OutputImage
```

ImageDatabasePassword

Description

Reserved for internal use.

ImageName

Description

The name of the image to display at a given layer in the display.

Syntax

`control.ImageName(Index As Integer) [= Value As String]`

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Set the output of the Acquisition Device as the image to
' show at layer 0 in the display
```

```
Set AcqDev1 = _
    Application1.ProcessManager.Process("Acquisition")
Display1.ImageDatabase(0) = Application1.Database.Handle
Display1.ImageViewName(0) = AcqDev1.OutputView
Display1.ImageName(0) = AcqDev1.OutputImage
```

ImagePalette

Description

The palette used to draw the image at a given layer in the display.

Syntax

`control.ImagePalette(Index As Integer) [= Value As hsImagePalette]`

Type

hsImagePalette

Range

Constant	Value	Description
hsPaletteColor	-6	The image is drawn in color.
hsPaletteBlue	-5	The image is drawn only to the blue component of the color.
hsPaletteGreen	-4	The image is drawn only to the green component of the color.
hsPaletteRed	-3	The image is drawn only to the red component of the color.

Constant	Value	Description
hsPaletteGreyScale	-2	The image is drawn using the standard grey-scale palette.
hsPaletteNone	-1	The image is not drawn.
hsPalette0	0	The image is drawn using the user-defined palette 0.
hsPalette1	1	The image is drawn using the user-defined palette 1.
hsPalette2	2	The image is drawn using the user-defined palette 2.
hsPalette3	3	The image is drawn using the user-defined palette 3.
hsPalette4	4	The image is drawn using the user-defined palette 4.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Example

```
' Hide the image at layer 0
Display1.ImagePalette(0) = hsPaletteNone
' Show the image at layer 1 as a grey-scale image
Display1.ImagePalette(1) = hsPaletteGreyScale
' Show the image at layer 2 using the user palette 1
Display1.ImagePalette(2) = hsPalette1
```

ImagePaletteColor

Description

The color at a given index in a user-defined palette.

Syntax

control.**ImagePaletteColor**(*PaletteIndex* As Integer, *ColorIndex* As Integer) [= *Value* As OLE_COLOR]

Type

OLE_COLOR

Range

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255).

Parameter(s)

Parameter(s)	Description	Type	Range
<i>PaletteIndex</i>	Index of the user-defined palette.	Integer	Minimum: 0 Maximum: ImageSize - 1
<i>ColorIndex</i>	Index of the color in the user-defined palette.	Integer	Depends on the setting of ImagePaletteType .

Example

```
' Use the user palette 0 to draw the blob image at
' layer 2
```

```

Display1.ImagePaletteType(0) = hsPaletteTypeColor8
Display1.ImagePaletteCycle(0) = True
Display1.ImagePaletteTransparencyIndex(0) = 0
Display1.ImagePaletteColor(0,0) = RGB(255,0,0)
Display1.ImagePaletteColor(0,1) = RGB(255,128,0)
Display1.ImagePaletteColor(0,2) = RGB(255,255,0)
Display1.ImagePaletteColor(0,3) = RGB(0,255,0)
Display1.ImagePaletteColor(0,4) = RGB(0,255,128)
Display1.ImagePaletteColor(0,5) = RGB(0,255,255)
Display1.ImagePaletteColor(0,6) = RGB(0,0,255)
Display1.ImagePaletteColor(0,7) = RGB(255,0,255)
Display1.ImagePalette(2) = hsPalette0
Display1.ImageDatabase(2) = Application1.Database.Handle
Display1.ImageViewName(2) = Blob.OutputView
Display1.ImageName(2) = Blob.OutputBlobImage

```

ImagePaletteCycle

Description

Selects whether the palette is indexed by the low bits or the high bits of the pixel values when the image is drawn. When **ImagePaletteCycle** is set to True, the palette is indexed by the low bits of the pixels and the color palette will cycle as the pixel values increase. For instance, with an 8 bit image and a 3 bit palette (hsPaletteTypeColor8), pixel values 0, 8, 16, 24,... will be drawn using the color at palette index 0; values 1, 9, 17, 25,... will use palette index 1, and so on. When **ImagePaletteCycle** is set to False, the palette is indexed by the high bits of the pixels. For instance, with an 8 bit image and a 3 bit palette (hsPaletteTypeColor8), pixel values between 0 and 31 will be drawn using the color at palette index 0; values between 32 and 63 will use palette index 1, and so on.

Syntax

control.**ImagePaletteCycle**(*PaletteIndex* As Integer) [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The palette is indexed using the low bits of pixel values.
True	The palette is indexed using the high bits of pixel values.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>PaletteIndex</i>	Index of the user-defined palette.	Integer	Minimum: 0 Maximum: ImageSize - 1

Example

' Use the user palette 0 to draw the blob image at

```

' layer 2
Display1.ImagePaletteType(0) = hsPaletteTypeColor8
Display1.ImagePaletteCycle(0) = True
Display1.ImagePaletteTransparencyIndex(0) = 0
Display1.ImagePaletteColor(0,0) = RGB(255,0,0)
Display1.ImagePaletteColor(0,1) = RGB(255,128,0)
Display1.ImagePaletteColor(0,2) = RGB(255,255,0)
Display1.ImagePaletteColor(0,3) = RGB(0,255,0)
Display1.ImagePaletteColor(0,4) = RGB(0,255,128)
Display1.ImagePaletteColor(0,5) = RGB(0,255,255)
Display1.ImagePaletteColor(0,6) = RGB(0,0,255)
Display1.ImagePaletteColor(0,7) = RGB(255,0,255)
Display1.ImagePalette(2) = hsPalette0
Display1.ImageDatabase(2) = Application1.Database.Handle
Display1.ImageViewName(2) = Blob.OutputView
Display1.ImageName(2) = Blob.OutputBlobImage

```

ImagePaletteMaskMode

Description

When **ImagePaletteMaskMode** is enabled the palette colors are drawn using a 50% transparency.

Syntax

control.**ImagePaletteMaskMode**(*PaletteIndex* As Integer) [= *Value* As Boolean]

Range

Value	Description
False	Palette colors are drawn without transparency. Default
True	Palette colors are drawn with 50% transparency.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>PaletteIndex</i>	Index of the user-defined palette.	Integer	Minimum: 0 Maximum: ImageSize - 1

Example

```

' Enable Palette mask Mode
Display1.ImagePaletteMaskMode = True

```

ImagePaletteTransparencyIndex

Description

Defines a pixel value in the image to display which will appear transparent when the image is drawn. **ImagePaletteTransparencyIndex** is useful for displaying images in overlay.

Syntax

control.**ImagePaletteTransparencyIndex**(*PaletteIndex* As Integer) [= *Value* As Integer]

Type

Integer

Range

Any pixel value in the image to display.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>PaletteIndex</i>	Index of the user-defined palette.	Integer	Minimum: 0 Maximum: ImageSize - 1

Example

```
' Use the user palette 0 to draw the blob image at
' layer 2
Display1.ImagePaletteType(0) = hsPaletteTypeColor8
Display1.ImagePaletteCycle(0) = True
Display1.ImagePaletteTransparencyIndex(0) = 0
Display1.ImagePaletteColor(0,0) = RGB(255,0,0)
Display1.ImagePaletteColor(0,1) = RGB(255,128,0)
Display1.ImagePaletteColor(0,2) = RGB(255,255,0)
Display1.ImagePaletteColor(0,3) = RGB(0,255,0)
Display1.ImagePaletteColor(0,4) = RGB(0,255,128)
Display1.ImagePaletteColor(0,5) = RGB(0,255,255)
Display1.ImagePaletteColor(0,6) = RGB(0,0,255)
Display1.ImagePaletteColor(0,7) = RGB(255,0,255)
Display1.ImagePalette(2) = hsPalette0
Display1.ImageDatabase(2) = Application1.Database.Handle
Display1.ImageViewName(2) = Blob.OutputView
Display1.ImageName(2) = Blob.OutputBlobImage
```

ImagePaletteType

Description

The type of a user-defined palette used to draw images.

Syntax

control.**ImagePaletteType**(*PaletteIndex* As Integer) [= *Value* As hsImagePaletteType]

Type

hsImagePaletteType

Range

Constant	Value	Description
hsPaletteTypeUndefined	0	No palette is defined.
hsPaletteTypeColor1	1	Single color palette.
hsPaletteTypeColor2	2	2 colors palette (1 bit index).

Constant	Value	Description
hsPaletteTypeColor8	8	8 colors palette (3 bit index).
hsPaletteTypeColor16	16	16 colors palette (4 bit index).
hsPaletteTypeColor256	256	256 colors palette (8 bit index).

Parameter(s)

Parameter(s)	Description	Type	Range
<i>PaletteIndex</i>	Index of the user-defined palette.	Integer	Minimum: 0 Maximum: ImageSize - 1

Example

```
' Use the user palette 0 to draw the blob image at
' layer 2
Display1.ImagePaletteType(0) = hsPaletteTypeColor8
Display1.ImagePaletteCycle(0) = True
Display1.ImagePaletteTransparencyIndex(0) = 0
Display1.ImagePaletteColor(0,0) = RGB(255,0,0)
Display1.ImagePaletteColor(0,1) = RGB(255,128,0)
Display1.ImagePaletteColor(0,2) = RGB(255,255,0)
Display1.ImagePaletteColor(0,3) = RGB(0,255,0)
Display1.ImagePaletteColor(0,4) = RGB(0,255,128)
Display1.ImagePaletteColor(0,5) = RGB(0,255,255)
Display1.ImagePaletteColor(0,6) = RGB(0,0,255)
Display1.ImagePaletteColor(0,7) = RGB(255,0,255)
Display1.ImagePalette(2) = hsPalette0
Display1.ImageDatabase(2) = Application1.Database.Handle
Display1.ImageViewName(2) = Blob.OutputView
Display1.ImageName(2) = Blob.OutputBlobImage
```

ImageSize

Description

Maximum number of images to be displayed.

Syntax

control.**ImageSize**[= *Value* As Integer]

Type

Integer

Range

Minimum:0

Maximum:1000

Default:5

Example

```
[...}
Display.ImageSize = 10
```

[...]

ImageViewName

Description

The name of the database view containing the image to display at a given layer in the display.

Syntax

control.**ImageViewName**(*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Set the output of the Acquisition Device as the image to
' show at layer 0 in the display
```

```
Set AcqDev1 = _
    Application1.ProcessManager.Process("Acquisition")
Display1.ImageDatabase(0) = Application1.Database.Handle
Display1.ImageViewName(0) = AcqDev1.OutputView
Display1.ImageName(0) = AcqDev1.OutputImage
```

InteractionMode

Description

Interaction mode of the Display control.

Syntax

control.**InteractionMode** [= *Value* As hsInteractionMode]

Type

hsInteractionMode

Range

Constant	Value	Description
hsZoomMode	0	Zoom interaction mode.

Constant	Value	Description
hsPanMode	1	Pan interaction mode.
hsSelectionMode	2	Selection interaction mode.

Example

```
Dim SelMode As hsInteractionMode
SelMode = Display1.InteractionMode
Display1.InteractionMode = hsPanMode
```

InteractiveSelectionEnabled

Description

Enables the interactive selection of Scenes. See also **InteractiveSelectionChange** event.

Syntax

control.**InteractiveSelectionEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Interactive selection is enabled.
False	Interactive selection is disabled.

Example

```
Dim Result As Boolean
Result = Display1.InteractiveSelectionEnabled
Display2.InteractiveSelectionEnabled = False
```

IsMarkerDefined

Description

Note: HexSight supports the **IsMarkerDefined** property for compatibility with applications built in previous versions. To specify whether a marker is defined, use the **MarkerDefined** property.

LabelMarkerText

Description

Text of a Label marker.

Syntax

control.**LabelMarkerText**(*Name* As String) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the Label marker.	String	Not applicable.

Example

```
Dim LabelName As String
LabelName = Display1.LabelMarkerText("MyLabelMarker")
Display1.LabelMarker("MyLabelMarker") = "FAILED"
```

LabelMarkerTextAlign

Description

Alignment of a Label marker with respect to its anchor point.

Syntax

```
control.LabelMarkerTextAlign(Name As String) [= Value As hsLabelMarkerTextAlign]
```

Type

hsLabelMarkerTextAlign

Range

Constant	Value	Description
hsLabelTopLeft	0	Top left alignment.
hsLabelTopCenter	6	Top center alignment.
hsLabelTopRight	2	Top right alignment.
hsLabelBottomLeft	8	Bottom left alignment.
hsLabelBottomCenter	14	Bottom center alignment.
hsLabelBottomRight	10	Bottom right alignment.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the selected label marker.	String	Not applicable.

Example

```
Dim Alignment As hsLabelMarkerTextAlign
Alignment = Display1.LabelMarkerTextAlign("MyLabelMarker")
Display1.LabelMarkerTextAlign("MyLabelMarkr") = hsLabelTopRight
```

LineMarkerConstraints

Description

Constraints that apply to a Line marker.

Syntax

```
control.LineMarkerConstraints(Name As String) [= Value As hsLineMarkerConstraints]
```

Type

hsLineMarkerConstraints

Range

Constant	Value	Description
hsLineNoConstraints	0	No constraint applies.
hsLineNoEdit	1	Edition disabled.
hsLineNoRefresh	2	Automatic refresh disabled. If the marker is modified interactively using the mouse, the Display control is not refreshed automatically.
hsLineNoResizeCorner	4	Corner points resizing disabled.
hsLineNoMove	16	Move disabled.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the Line marker.	String	Not applicable.

Remark(s)

The default constraint for a Line marker is hsLineNoEdit.

Example

```
Dim Const As hsLineMarkerConstraints
Const = Display1.LineMarkerConstraints("MyLineMarker")
Display1.LineMarkerConstraints("MyLineMarker") = (hsLineNoEdit +
hsLineNoRefresh)
```

LineMarkerX2

Description

X coordinate of the second point of a Line marker.

Syntax

control.**LineMarkerX2**(*Name* As String) [= *Value* As Single]

Type

Single

Range

Boundaries of the displayed image.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the Line marker.	String	Not applicable.

Remark(s)

The length unit used to expressed this property depends on the status of the **MarkerCalibrated** property as shown in the following table:

MarkerCalibrated property	Length unit used
False	Pixel.
True	As defined in UnitsLength .

Example

```
Dim XPos As Single
Xpos = Display1.LineMarkerX2("MyLineMarker")
Display1.LineMarkerX2("MyLineMarker") = 120.0
```

LineMarkerY2

Description

Y coordinate of the second point of a Line marker.

Syntax

control.**LineMarkerY2**(*Name* As String) [= *Value* As Single]

Type

Single

Range

Boundaries of the displayed image.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the Line marker.	String	Not applicable.

Remark(s)

The length unit used to expressed this property depends on the status of the **MarkerCalibrated** property as shown in the table below:

MarkerCalibrated property	Length unit used
False	Pixel.
True	As defined in UnitsLength .

Example

```
Dim YPos As Single
Ypos = Display1.LineMarkerY2("MyLineMarker")
Display1.LineMarkerY2("MyLineMarker") = 30.0
```

MarkerAnchorColor

Description

Color of the anchor point of a marker.

Syntax

control.**MarkerAnchorColor**(*Name* As String) [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).

Constant	Value	Description
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

This property applies only to the following types of markers: Point, Line and Rectangle.

Example

```
Dim MyColor As OLE_COLOR
MyColor = Display1.MarkerAnchorColor("MyMarker")
Display1.MarkerAnchorColor("MyMarker") = hsGreen
Display1.MarkerAnchorColor("MyMarker") = RGB(128,0,128)
```

MarkerAnchorStyle

Description

Style of the anchor point of a marker.

Syntax

control.MarkerAnchorStyle(*Name* As String) [= *Value* As hsMarkerAnchorStyle]

Type

hsMarkerAnchorStyle

Range

Constant	Value	Description
hsSquare	1	Square anchor points.
hsSquareFilled	2	Filled square anchor points.
hsDiamond	3	Diamond anchor points.
hsDiamondFilled	4	Filled Diamond anchor points.
hsCircle	5	Circle anchor points.
hsCircleFilled	6	Filled circle anchor points.
hsPlus	7	Plus anchor points.
hsCross	8	Cross anchor points.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

This property applies only to the following types of markers: Point, Line and Rectangle.

Example

```
Dim Style As hsMarkerAnchorStyle
Style = Display1.MarkerAnchorStyle("MyMarker")
Display1.MarkerAnchorStyle("MyMarker") = hsPlus
```

MarkerCalibrated

Description

Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system. This property is set when the marker is created. Read only.

Syntax

control.**MarkerCalibrated**(*Name* As String)

Type

Boolean

Range

False, True

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The methods used to create or modify the configuration of a marker (e.g:

AddRectangleMarker, **ChangeRectangleMarker**) contain a parameter named Calibrated that sets the **MarkerCalibrated** property. This property can be set using only these methods.

Example

```
Dim Calib As Boolean
Calib = Display1.MarkerCalibrated("MyMarker")
```

MarkerColor

Description

Color of the selected marker.

Syntax

control.**MarkerColor**(*Name* As String) [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
Dim MyColor As OLE_COLOR
MyColor = Display1.MarkerColor("MyMarker")
Display1.MarkerColor("MyMarker") = hsGreen
Display1.MarkerColor("MyMarker") = RGB(128,0,128)
```

MarkerDefined

Description

Specifies if the selected marker is defined. Read only.

Syntax

control.**MarkerDefined**(Name As String)

Type

Boolean

Range

True if the selected marker is defined, otherwise False.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

Replaces the **LoadConfiguration** method.

Example

```
Dim IsDefined As Boolean
IsDefined = Display1.MarkerDefined("MyArcMarker")
```

MarkerDisplayName

Description

Specifies if the name of the marker is displayed.

Syntax

control.**MarkerDisplayName**(*Name* As String)

Type

Boolean

Range

False, True

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
Dim Visible As Boolean
Visible = Display1.MarkerDisplayName
Display1.MarkerDisplayName = True
```

MarkerGroup

Description

Group of the selected marker. This property is used to create marker groups. All markers that are members of a same group can be removed from the display using the **RemoveMarkerGroup** method.

Syntax

control.**MarkerGroup**(*Name* As String) [= *Value* As Integer]

Type

Integer

Range

Minimum: 0 (default value)
Maximum: unlimited

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
Dim ID As Integer
ID = Display1.MarkerGroup("MyMarker")
Display1.MarkerGroup("MyMarker")= 2
```

MarkerLineStyle

Description

Line style used to draw a marker.

Syntax

control.**MarkerLineStyle**(*Name* As String) [= *Value* As hsMarkerLineStyle]

Type

hsMarkerLineStyle

Range

Constant	Value	Description
hsSolid	0	Solid line.
hsDash	16	Dashed line.
hsDot	32	Dotted line.
hsDashDot	48	Dash-dotted line.
hsDashDotDot	64	Dash-dot-dotted line.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

This property applies to the following types of marker: Axes, CircularArc, Line, Rectangle and Target.

Example

```
Dim LineStyle As hsMarkerLineStyle
LineStyle = Display1.MarkerLineStyle("MyMarker")
Display1.MarkerLineStyle("MyMarker") = hsDash
```

MarkerPenWidth

Description

The width of the pen used to draw a marker.

Syntax

control.**MarkerPenWidth**(*Name* As String) [= *Value* As hsPenWidth]

Type

hsPenWidth

Range

Constant	Value	Description
hsPenWidthNone	0	Not drawn.
hsPenWidthThin	1	Drawn using thin lines.
hsPenWidthThick	2	Drawn using thick lines.

Example

```
Dim PenWidth As hsPenWidth
PenWidth = Display1.MarkerPenWidth("MyMarker")
Display1.MarkerPenWidth("MyMarker") = hsPenWidthThick
```

MarkerX

Description

X coordinate of a marker.

Syntax

control.**MarkerX**(Name As String) [= Value As Single]

Type

Single

Range

Boundaries of the display.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The X coordinate of the marker is expressed with respect to a coordinate system, which can be the Image or the World coordinate system depending on the status of the **MarkerCalibrated** property, as shown in the following table. The length units used to express this property is also provided.

MarkerCalibrated property	Reference coordinate system	Length unit used
False	Image	Pixel.
True	World	As defined in UnitsLength property.

Example

```
Dim XPos As Single
XPos = Display1.MarkerX("MyMarker")
Display1.MarkerX("MyMarker") = 120.0
```

MarkerY

Description

Y coordinate of a marker.

Syntax

control.**MarkerY**(Name As String) [= Value As Single]

Type

Single

Range

Boundaries of the display.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The Y coordinate of the marker is expressed with respect to a coordinate system, which can be the Image or the World coordinate system depending on the status of the **MarkerCalibrated** property, as shown in the table below. The length units used to express this property is also provided:

MarkerCalibrated property	Reference coordinate system	Length unit used
False	Image	Pixel.
True	World	As defined in UnitsLength property.

Example

```
Dim YPos As Single
YPos = Display1.MarkerY("MyMarker")
Display1.MarkerY("MyMarker") = 40.0
```

MaximalBoundaryHeight

Description

Maximal height of the display. Read only.

Syntax

control.**MaximalBoundaryHeight**

Type

Single

Range

Minimum: greater than zero.

Maximum: depends on the size in pixels of the image to display or the **UnitsLength** property if the **CalibratedUnitsEnabled** property is set to True.

Example

```
Dim MaxWidth As Single
MaxWidth = Display1.MaximalBoundaryHeight
```

MaximalBoundaryWidth

Description

Maximal width of the display. Read only.

Syntax

control.**MaximalBoundaryWidth**

Type

Single

Range

Minimum: greater than zero.

Maximum: depends on the size in pixels of the image to display or the **UnitsLength** property if the **CalibratedUnitsEnabled** property is set to True.

Example

```
Dim MaxWidth As Single
MaxWidth = Display1.MaximalBoundaryWidth
```

MultiSelectionEnabled

Description

Enables the selection of multiple elements contained in an area.

Syntax

control.**MultiSelectionEnabled** [= *Value* As Integer]

Type

Boolean

Range

Value	Description
True	MultiSelection is enabled.
False	MultiSelection is disabled.

Example

```
Dim Result As Boolean
Result = Display1.MultiSelectionEnabled
Display2.MultiSelectionEnabled = False
```

PointArrayMarkerSize

Description

Size of the Point Array marker.

Syntax

control.**PointArrayMarkerSize**(*Name* As String) [= *Value* As Integer]

Type

Integer

Range

Minimum:0

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
' Translate all points in the Point Array Marker
for i = 0 To Display.PointArrayMarkerSize -1
    Display.PointArrayMarkerX("PointArray",i)= _
        Display.PointArrayMarkerY("PointArray",i) + 10
    Display.PointArrayMarkerY("PointArray",i)= _
        Display.PointArrayMarkerY("PointArray",i) + 20
Next
```

PointArrayMarkerX

Description

X coordinate of the Point Marker in the array.

Syntax

control.**PointArrayMarkerX**(*Name* As String, *Index* As Integer) [= *Value* As Integer]

Type

Integer

Range**Parameter(s)**

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: PointArrayMarkerSize -1

Example

```
' Translate all points in the Point Array Marker
for i = 0 To Display.PointArrayMarkerSize -1
    Display.PointArrayMarkerX("PointArray",i)= _
        Display.PointArrayMarkerX("PointArray",i) + 10
    Display.PointArrayMarkerY("PointArray",i)= _
        Display.PointArrayMarkerY("PointArray",i) + 20
Next
```

PointArrayMarkerY

Description

Y coordinate of the Point Marker in the array.

Syntax

control.**PointArrayMarkerY**(*Name* As String, *Index* As Integer) [= *Value* As Integer]

Type

Integer

Range**Parameter(s)**

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: PointArrayMarkerSize -1

Example

```
' Translate all points in the Point Array Marker
for i = 0 To Display.PointArrayMarkerSize -1
    Display.PointArrayMarkerX("PointArray",i)= _
        Display.PointArrayMarkerX("PointArray",i) + 10
    Display.PointArrayMarkerY("PointArray",i)= _
```

```
Display.PointArrayMarkerY("PointArray",i) + 20
```

Next

PointMarkerConstraints

Description

Constraints that apply to an Point marker.

Syntax

```
control.PointMarkerConstraint(Name As String) [= Value As hsPointMarkerConstraints]
```

Type

hsPointMarkerConstraints

Range

Constant	Value	Description
hsPointNoConstraints	0	No constraint.
hsPointNoEdit	1	Edition disabled.
hsPointNoRefresh	2	Automatic refresh disabled. If the marker is modified interactively using the mouse, the Display control is not refreshed automatically.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The default constraint for a Point marker is hsPointNoEdit.

Example

```
Dim Constraints As hsPointMarkerConstraints
Constraints = Display1.PointMarkerConstraints("MyMarker")
Display1.PointMarkerConstraints("MyMarker") = hsPointNoEdit
```

PriorityMask

Description

PriorityMask allows you to change the drawing priority of the display. The default priority (drawing order) of layers, from bottommost to topmost is: Image, Scene, Entities, Markers. When **hsMaskMarkerGroupPriority** is set to true, markers are drawn by group according to their **MarkerGroup**: Group0 is the first group drawn (bottommost), then Group1, Group2 and so forth.

Syntax

```
control.PriorityMask(Index As hsPriorityMask) [= Value As Boolean]
```

Type

Boolean

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Index</i>	Index of the Priority Mask	Integer	See table below.

Range

Constant	Value	Description
hsMaskEntitiesOnTop	1	When true, the Entities layer is drawn over the marker layer.
hsMaskMarkerGroupPriority	2	When set to true, Markers are drawn by group, with group 0 being the first (bottommost) to be drawn.

Example

Display1.**PriorityMask**(hsMaskEntitiesOnTop) = True

RectangleMarkerConstraints

Description

Constraints that apply to a Rectangle marker.

Syntax

control.**RectangleMarkerConstraints**(*Name* As String) [= *Value* As
hsRectangleMarkerConstraints]

Type

hsRectangleMarkerConstraints

Range

Constant	Value	Description
hsRectangleNoConstraints	0	No constraints.
hsRectangleNoEdit	1	Edition disabled.
hsRectangleNoRefresh	2	Automatic refresh disabled. If the marker is modified interactively using the mouse, the Display control is not refreshed automatically.
hsRectangleNoResizeCorner	4	Corner points resizing disabled.
hsRectangleNoResizeMidpoint	8	Middle point resizing disabled.
hsRectangleNoMove	16	Move disabled.
hsRectangleNoRotation	32	Rotation disabled.
hsRectangleNoSkew	64	Skew disabled.
hsRectangleCornerBasedEdition	256	When dragging a resize handle, the rectangle dimension is not mirrored with respect to its center.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The default constraint for a Rectangle marker is `hsRectangleNoEdit`.

Example

```
Dim Constraints As hsRectangleMarkerConstraints
Constraints = Display1.RectangleMarkerConstraints("MyMarker")
Display1.RectangleMarkerConstraints("MyMarker") _
    = hsRectangleNoEdit + hsRectangleNoMove
```

RectangleMarkerHeight

Description

Height of a Rectangle marker.

Syntax

control.**RectangleMarkerHeight**(Name As String) [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the display.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The property is expressed in pixels if the **MarkerCalibrated** property is set to False, otherwise it is expressed in a unit length defined by the **UnitsLength** property.

Example

```
Dim Height As Single
Height = Display1.RectangleMarkerHeight("MyMarker")
Display1.RectangleMarkerHeight("MyMarker") = 200.0
```

RectangleMarkerRotation

Description

Angle of rotation of a Rectangle marker.

Syntax

control.**RectangleMarkerRotation**(Name As String) [= Value As Single]

Type

Single

Range

This range of this property depends on the **UnitsOrientation** property.

UnitsOrientation property	Range of RectangleMarkerRotation property
hsDegree	Minimum: -180 Maximum: +180

UnitsOrientation property	Range of RectangleMarkerRotation property
hsRadian	Minimum: -Pi Maximum: +Pi

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

This property is defined as the angle between the X-axis of coordinate of a reference coordinate system and one of the horizontal lines of the marker. The reference coordinate system can be the Image or the World coordinate system depending on the status of the **MarkerCalibrated** property.

MarkerCalibrated property	Reference coordinate system
False	Image
True	World

Example

```
Dim Angle As Single
Angle = Display1.RectangleMarkerRotation("MyMarker")
Display1.RectangleMarkerRotation("MyMarker") = 30.0
```

RectangleMarkerSkew

Description

Skew angle of a Rectangle marker.

Syntax

```
control.RectangleMarkerSkew(Name As String) [= Value As Single]
```

Type

Single

Range

This range of this property depends on the **UnitsOrientation** property.

UnitsOrientation property	Range of RectangleMarkerSkew property
hsDegree	Minimum: -90 Maximum: +90
hsRadian	Minimum: -Pi/2 Maximum: +Pi/2

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Example

```
Dim Angle As Single
Angle = Display1.RectangleMarkerSkew("MyMarker")
Display1.RectangleMarkerSkew("MyMarker") = 30.0
```

RectangleMarkerWidth

Description

The width of a Rectangle marker.

Syntax

control.**RectangleMarkerWidth**(*Name* As String) [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the display.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The property is expressed in pixels if the **MarkerCalibrated** property is set to False, otherwise it is expressed in a unit length defined by the **UnitsLength** property.

Example

```
Dim Width As Single
Width = Display1.RectangleMarkerWidth("MyMarker")
Display1.RectangleMarkerWidth("MyMarker") = 200.0
```

RulersVisible

Description

Specifies if the rulers of the display are visible.

Syntax

control.**RulersVisible** [= *Value* As Boolean]

Type

Boolean

Range

False, True

Remark(s)

If the **ScaleToFitEnabled** property is set to True, the status of the **RulersVisible** property is not considered and the rulers are not displayed.

Example

```
Dim Visible As Boolean
Visible = Display1.RulersVisible
Display1.RulersVisible = True
```

ScaleToFitEnabled

Description

Specifies if the "Scale to fit" mode is enabled. When set to True, the rulers, tool bar, status bar, scroll bars and title are removed and the image is scaled in order to fit entirely in the display.

Syntax

control.**ScaleToFitEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = Display1.ScaleToFitEnabled
Display1.ScaleToFitEnabled = False
```

SceneColor

Description

The color used to draw the scene at a given layer in the display.

Syntax

control.**SceneColor**(*Index* As Integer) [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
```

SceneDatabase

Description

The handle to the database containing the scene to display at a given layer in the display.

Syntax

control.**SceneDatabase**(*Index* As Integer) [= *Value* As OLE_HANDLE]

Type

OLE_HANDLE

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
```

SceneDatabasePassword

Description

Reserved for internal use.

SceneName

Description

The name of the scene to display at a given layer in the display.

Syntax

control.**SceneName**(*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
```

ScenePenWidth

Description

The width of the pen used to draw the scene at a given layer in the display.

Syntax

control.**ScenePenWidth**(*Index* As Integer) [= *Value* As hsPenWidth]

Type

hsPenWidth

Range

Constant	Value	Description
hsPenWidthNone	0	Not drawn.
hsPenWidthThin	1	Drawn using thin lines.
hsPenWidthThick	2	Drawn using thick lines.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
```

```

Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene

```

SceneSelection

Description

Specifies the contents of a scene selection to display at a given layer in the display. A **SceneSelection** is a string containing the indexes of the selected items. For example, to select the edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of a scene, the **SceneSelection** string would be: "2:5,7:9,12,15". The type of items selected in the scene with **SceneSelection** is set with **SceneSelectionType**. The scene layer to which **SceneSelection** refers to is set with **SceneSelectionIndex**.

Syntax

```
control.SceneSelection(Index As Integer) [= Value As String]
```

Type

String

Range

A valid selection string which consists of a list of comma separated indexes. Ranges can also be specified in selection strings using ":". Examples of valid selection strings: "0, 1, 2, 5, 6, 7, 8, 9, 11, 23", "0:2, 5:9, 11, 23", "0:10, 12, 15:19", etc.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: SceneSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```

' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
' Highlight edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of
' the Outline contours in red
Display1.SceneSelectionColor(0) = hsRed
Display1.SceneSelectionPenWidth(0) = hsPenWidthThin
Display1.SceneSelectionType(0) = hsTypeEdgePath

```



```
Display1.SceneSelectionIndex(0) = 0
Display1.SceneSelection(0) = "2:5,7:9,12,15"
```

SceneSelectionColor

Description

The color used to draw the scene selection at a given layer in the display.

Syntax

control.**SceneSelectionColor**(*Index* As Integer) [= *Value* As hsColor]

Type

hsColor

Range

The hsColor type is a set of predefined OLE_COLOR colors. The valid range for hsColor is:

Constant	Value	Description
hsBlack	&H000000	Black. RGB(0,0,0).
hsRed	&H0000FF	Red. RGB(255,0,0).
hsGreen	&H00FF00	Green. RGB(0,255,0).
hsYellow	&H00FFFF	Yellow. RGB(0,255,255).
hsBlue	&HFF0000	Blue. RGB(255,0,0).
hsMagenta	&HFF00FF	Magenta. RGB(255,0,255).
hsCyan	&HFFFF00	Cyan. RGB(255,255,0).
hsWhite	&HFFFFFF	White. RGB(255,255,255).

Any OLE_COLOR color in the range RGB(0,0,0) to RGB(255,255,255) can also be used instead of one of the predefined colors.

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: SceneSize - 1

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
' Highlight edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of
' the Outline contours in red
Display1.SceneSelectionColor(0) = hsRed
Display1.SceneSelectionPenWidth(0) = hsPenWidthThin
Display1.SceneSelectionType(0) = hsTypeEdgePath
Display1.SceneSelectionIndex(0) = 0
```

```
Display1.SceneSelection(0) = "2:5,7:9,12,15"
```

SceneSelectionIndex

Description

Specifies to which scene layer the scene selection at a given layer in the display refers to. The contents of the scene selection is set with **SceneSelection**. The type of items selected in the scene with **SceneSelection** is set with **SceneSelectionType**.

Syntax

```
control.SceneSelectionIndex(Index As Integer) [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: 4

Parameter(s)

Parameter(s)	Description	Type	Range
Index	Index of the layer.	Integer	Minimum: 0 Maximum: SceneSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
' Highlight edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of
' the Outline contours in red
Display1.SceneSelectionColor(0) = hsRed
Display1.SceneSelectionPenWidth(0) = hsPenWidthThin
Display1.SceneSelectionType(0) = hsTypeEdgePath
Display1.SceneSelectionIndex(0) = 0
Display1.SceneSelection(0) = "2:5,7:9,12,15"
```

SceneSelectionPenWidth

Description

The width of the pen used to draw the scene selection at a given layer in the display.

Syntax

control.**SceneSelectionPenWidth**(*Index* As Integer) [= *Value* As hsPenWidth]

Type

hsPenWidth

Range

Constant	Value	Description
hsPenWidthNone	0	Not drawn.
hsPenWidthThin	1	Drawn using thin lines.
hsPenWidthThick	2	Drawn using thick lines.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
' Highlight edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of
' the Outline contours in red
Display1.SceneSelectionColor(0) = hsRed
Display1.SceneSelectionPenWidth(0) = hsPenWidthThin
Display1.SceneSelectionType(0) = hsTypeEdgePath
Display1.SceneSelectionIndex(0) = 0
Display1.SceneSelection(0) = "2:5,7:9,12,15"
```

SceneSelectionType

Description

Specifies the type of items selected by the scene selection at a given layer in the display. The contents of the scene selection is set with **SceneSelection**. The scene layer to which **SceneSelection** refers to is set with **SceneSelectionIndex**.

Syntax

control.**SceneSelectionType**(*Index* As Integer) [= *Value* As hsSceneSelectionType]

Type

hsSceneSelectionType

Range

Constant	Value	Description
hsTypeEdgel	0	The indexes in the scene selection string refer to edge elements.
hsTypeEdgePath	1	The indexes in the scene selection string refer to edge paths.
hsTypeGroup	2	The indexes in the scene selection string refer to edge path groups.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: SceneSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
' Highlight edge paths 2, 3, 4, 5, 7, 8, 9, 12 and 15 of
' the Outline contours in red
Display1.SceneSelectionColor(0) = hsRed
Display1.SceneSelectionPenWidth(0) = hsPenWidthThin
Display1.SceneSelectionType(0) = hsTypeEdgePath
Display1.SceneSelectionIndex(0) = 0
Display1.SceneSelection(0) = "2:5,7:9,12,15"
```

SceneSize**Description**

Maximum number of scenes that can be displayed.

Syntax

```
control.SceneSize[= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: 1000

Default: 5

Example

```
[...]
Display.SceneSize = 10
[...]
```

SceneViewName

Description

The name of the database view containing the scene to display at a given layer in the display.

Syntax

control.**SceneViewName**(*Index* As Integer) [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: ImageSize - 1

Remark(s)

The display supports 5 image layers, 5 scene layers, 10 scene selection layers and one marker layer. When the display is refreshed, the image at layer 0 is first drawn, followed by the image at layer 1, and so on. Once the 5 image layers are drawn, the scene at layer 0 is drawn along with its associated scene selection layers, followed by the scene at layer 1 and its associated scene selection layers, and so on. After the 5 scene layers and their associated scene selection layers are drawn, the marker layer is drawn.

Example

```
' Display the detected Outline contours in magenta
Display1.SceneColor(0) = hsMagenta
Display1.ScenePenWidth(0) = hsPenWidthThin
Display1.SceneDatabase(0) = Application1.Database.Handle
Display1.SceneViewName(0) = Locator.OutputOutlineSceneView
Display1.SceneName(0) = Locator.OutputOutlineScene
```

ScrollBarsVisible

Description

Specifies if the scroll bars are visible.

Syntax

control.**ScrollBarsVisible** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = Display1.ScrollBarsVisible
Display1.ScrollBarsVisible = False
```

StatusBarVisible

Description

Specifies if the status bar is visible.

Syntax

control.**StatusBarVisible** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Visible As Boolean
Visible = Display1.StatusBarVisible
Display1.StatusBarVisible = True
```

TargetMarkerConstraints

Description

Constraints that apply to a selected Target marker.

Syntax

control.**TargetMarkerConstraints**(*Name* As String) [= *Value* As hsTargetMarkerConstraints]

Type

hsTargetMarkerConstraints

Range

Constant	Value	Description
hsTargetNoConstraints	0	No constraints.
hsTargetNoEdit	1	Edition disabled.
hsTargetNoRefresh	2	Automatic refresh disabled. If the marker is modified interactively using the mouse, the Display control is not refreshed automatically.
hsTargetFixedSize	128	Fixed size mode enabled. The Target marker has always the same size with respect to the Display.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The default constraint for a Target marker is hsTargetNoEdit + hsTargetFixedSize.

Example

```
Dim Constraints As hsTargetMarkerConstraints
Constraints = Display1.TargetMarkerConstraints("MyMarker")
Display1.TargetMarkerConstraints("MyMarker")= _
    hsTargetNoEdit + hsTargetNoRefresh
```

TargetMarkerRadius

Description

Radius of a Target marker.

Syntax

control.**TargetMarkerRadius**(Name As String) [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the display.

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

The property is expressed in pixels if the **MarkerCalibrated** property is set to False, otherwise it is expressed in a unit length defined by the **UnitsLength** property.

Example

```
Dim Radius As Single
```

```
Radius = Display1.TargetMarkerRadius("MyMarker")
```

```
Display1.TargetMarkerRadius("MyMarker") = 10.0
```

TargetMarkerRotation

Description

Angle of rotation of a Target marker.

Syntax

control.**TargetMarkerRotation**(Name As String) [= Value As Single]

Type

Single

Range

This range of this property depends on the **UnitsOrientation** property.

UnitsOrientation property	Range of RectangleMarkerRotation property
hsDegree	Minimum: -180 Maximum: +180
hsRadian	Minimum: -Pi Maximum: +Pi

Parameter(s)

Parameter	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.

Remark(s)

This property is the angle between the X-axis of a reference coordinate system and the horizontal line of the marker, going from the former to the latter. The coordinate system used as a reference depends on the status of the **MarkerCalibrated** property.

MarkerCalibrated property	Reference coordinate system
False	Image
True	World

Example

```
Dim Angle As Single
Angle = Display1.TargetMarkerRotation("MyMarker")
Display1.TargetMarkerRotation("MyMarker") = 30.0
```

Title

Description

String displayed in title bar when visible.

Syntax

control.**Title** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim DisplayTitle As String
DisplayTitle = Display1.Title
Display1.Title = "Application Display"
```

TitleVisible

Description

Specifies if the title bar of the Display is visible.

Syntax

control.**TitleVisible** [= Value As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Visible As Boolean
Visible = Display1.TitleVisible
Display1.TitleVisible = False
```


ToolBarVisible

Description

Specifies if the tool bar is visible.

Syntax

control.**ToolBarVisible** [= Value As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Visible As Boolean
Visible = Display1.ToolBarVisible
Display1.ToolBarVisible = True
```

UnitsLength

Description

Unit used to express calibrated dimensions. This property is applicable only if the **CalibratedUnitsEnabled** property is set to True.

Syntax

control.**UnitsLength** [= Value As hsUnitsLength]

Type

hsUnitsLength

Range

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = Display1.UnitsLength
Display1.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express angles.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Display1.UnitsOrientation
Display1.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Display1.Version
```

WheelOperationMode

Description

Controls if the display content is zoomed in or out on mouse wheel interaction.

Syntax

control.**WheelOperationMode**

Type

Integer

Range

Value	Description
0	Zoom operation allowed (default)
1	Zoom operation allowed only if mouse pointer actually inside Display area
2	Zoom operation allowed only if mouse pointer actually outside Display area

Value	Description
3	No Zoom operation allowed

Example

```
Dim Mode As Integer
Mode = Display1.WheelOperationMode
```

Zoom

Description

Zoom factor used to display Images or Scenes.

Syntax

```
control.Zoom [= Value As Single]
```

Type

Single

Range

-1: ZoomToPage
 Minimum: 0.10 (10%)
 Maximum: 64.0 (6400%)

Example

```
Dim Zoom As Single
Zoom = Display1.Zoom
Display1.Zoom = 2.0
```

ZoomToPage

Description

ZoomToPage returns True when current zoom mode is Zoom to Page. This property is affected by the **Zoom** property, which always returns the current zoom as a percentage. Read only.

Syntax

```
control.ZoomToPage
```

Type

Boolean

Range

True when the current zoom is Zoom to Page. False otherwise.

Example

```
' Synchronize two displays
If Display1.ZoomToPage = True Then
    Display2.Zoom = -1
Else
    Display2.Zoom = Display1.Zoom
End If
```

Methods

AboutBox

Description

Displays the Display control About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

Display1.**AboutBox**

AddAxesMarker

Description

Adds an Axes marker to the display.

Syntax

control.**AddAxesMarker**(*Name* As String, *X* As Single, *Y* As Single, *Length1* As Single, *Length2* As Single, *Rotation* As Single, *MidAngle* As Single, *Calibrated* As Boolean) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.
<i>X</i>	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Y</i>	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Length1</i>	Length of the axe #1 in pixels or in calibrated units.	Single	Minimum: 0 Maximum: unlimited
<i>Length2</i>	Length of the axe #2 in pixels or in calibrated units.	Single	Minimum: 0 Maximum: unlimited
<i>Rotation</i>	Rotation angle of the marker.	Single	Any Single value.
<i>MidAngle</i>	Angle between the axes of the marker.	Single	If the UnitsOrientation property is set to hsDegree : Minimum: -180 deg, Maximum: 180 deg otherwise : Minimum: -Pi, Maximum: Pi

Name	Description	Type	Range
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
<code>hsSuccess</code>	0	Operation succeeded.
<code>hsFailed</code>	1	Operation failed.
<code>hsChanged</code>	2	The marker was already created and has been modified using the provided parameters.
<code>hsAlreadyExists</code>	3	A marker of another type already uses the specified <i>Name</i> .

Remark(s)

By default this method sets the `hsAxesNoEdit` marker constraint.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddAxesMarker _
    ("Axes Marker", 125, 50, 100, 100, 0, 90, False)
```

AddCircularArcMarker

Description

Adds a Circular Arc marker to the display.

Syntax

control.**AddCircularArcMarker**(*Name* As String, *X* As Single, *Y* As Single, *RadiusMin* As Single, *Thickness* As Single, *Rotation* As Single, *Opening* As Single, *Calibrated* As Boolean) As `hsMarkerStatus`

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.
<i>X</i>	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Y</i>	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Radius</i>	Mean radius in pixels or in calibrated units.	Single	Any Single value.

Name	Description	Type	Range
<i>Thickness</i>	Radial distance between the two bounding annuli of the marker.	Single	Any Single value.
<i>Rotation</i>	Angle between bisector of the marker and the X-Axis of the World or Image coordinate system (depending if the marker is calibrated or not).	Single	Any Single value.
<i>Opening</i>	Angle between the two bounding radii of the marker.	Single	If the UnitsOrientation property is set to hsDegree : Minimum: 0 deg, Maximum: 360 deg otherwise : Minimum: 0, Maximum: 2*Pi
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False.

Remark(s)

By default this method sets the `hsCircularArcNoEdit` and `hsCircularArcNoProjection` marker constraints.

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
<code>hsSuccess</code>	0	Operation succeeded.
<code>hsFailed</code>	1	Operation failed.
<code>hsChanged</code>	2	The marker was already created and has been modified using the provided parameters.
<code>hsAlreadyExists</code>	3	A marker of another type already uses the specified <i>Name</i> .

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddCircularArcMarker _
    ("Arc Marker", 100, 100, 25, 50, 30, 90, False)
```

AddLabelMarker

Description

Adds a Label marker to the Display output.

Syntax

```
control.AddLabelMarker(Name As String, X As Single, Y As Single, Text As String, Calibrated As Boolean As hsMarkerStatus
```

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Text	Label text string	String	Not applicable.
Calibrated	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsChanged	2	The marker was already created and has been modified using the provided parameters.
hsAlreadyExists	3	A marker of another type already uses the specified <i>Name</i> .

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddLabelMarker _
    ("Label Marker", 230, 420, "Label String", False)
```

AddLineMarker

Description

Adds a Line marker to the Display output.

Syntax

```
control.AddLineMarker(Name As String, X As Single, Y As Single, X2 As Single, Y2 As Single,
    Calibrated As Boolean As hsMarkerStatus)
```

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image

Name	Description	Type	Range
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image
X2	"X" end of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image
Y2	"Y" end of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
<code>hsSuccess</code>	0	Operation succeeded.
<code>hsFailed</code>	1	Operation failed.
<code>hsChanged</code>	2	The marker was already created and has been modified using the provided parameters.
<code>hsAlreadyExists</code>	3	A marker of another type already uses the specified <i>Name</i> .

Remark(s)

By default this method sets the `hsLineNoEdit` marker constraint.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddLineMarker _
    ("Line Marker", 73, 120, 143, 160, False)
```

AddPointArrayMarker

Description

Adds a Point Array Marker to the Display output. Using an array of point markers improves performance when a large number of point markers is required.

Syntax

control.**AddPointArrayMarker**(*Name* As String, *X* As Single, *Y* As Single, *Size* As Integer, *Calibrated* As Boolean) As `hsMarkerStatus`

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	Default X origin of the markers in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.

Name	Description	Type	Range
<i>Y</i>	Default Y origin of the markers in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Size</i>	Number of point markers to display.	Integer	Minimum: 0 Maximum: 1000 Default: 50
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a **hsMarkerStatus**. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsChanged	2	The marker was already created and has been modified using the provided parameters.
hsAlreadyExists	3	A marker of another type already uses the specified <i>Name</i> .

Example

```

AddPointArrayMarker("PointArray",0,0,10,false)
Dim Index as Integer
Index = 0
For x = 10 to 30 Step 10
    For y = 10 to 30 Step 10
        SetPointArrayMarkerPosition("PointArray",Index,x,y)
    Next y
Next x

```

AddPointMarker

Description

Adds a Point marker to the Display output.

Syntax

control.**AddPointMarker**(*Name* As String, *X* As Single, *Y* As Single, *Calibrated* As Boolean) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.

Name	Description	Type	Range
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Calibrated	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsChanged	2	The marker was already created and has been modified using the provided parameters.
hsAlreadyExists	3	A marker of another type already uses the specified <i>Name</i> .

Remark(s)

By default this method sets the **hsPointNoEdit** marker constraint (See **PointMarkerConstraints**).

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddPointMarker _
    ("Point Marker", 350, 175, False)
```

AddRectangleMarker

Description

Adds a Rectangle marker to the Display output. The Rectangle is defined with respect to the center of the marker, NOT the bottom left corner.

Syntax

control.**AddRectangleMarker**(*Name* As String, *X* As Single, *Y* As Single, *Width* As Single, *Height* As Single, *Calibrated* As Boolean) As `hsMarkerStatus`

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.

Name	Description	Type	Range
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
<code>hsSuccess</code>	0	Operation succeeded.
<code>hsFailed</code>	1	Operation failed.
<code>hsChanged</code>	2	The marker was already created and has been modified using the provided parameters.
<code>hsAlreadyExists</code>	3	A marker of another type already uses the specified <i>Name</i> .

Remark(s)

By default this method sets the `hsRectangleNoEdit` marker constraint.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddRectangleMarker_
    ("Rectangle Marker", 230, 350, 20, 50, False)
```

AddTargetMarker

Description

Adds a Target marker to the Display output.

Syntax

control.**AddTargetMarker**(*Name* As String, *X* As Single, *Y* As Single, *Radius* As Single, *Calibrated* As Boolean) As `hsMarkerStatus`

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.

Name	Description	Type	Range
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
<code>hsSuccess</code>	0	Operation succeeded.
<code>hsFailed</code>	1	Operation failed.
<code>hsChanged</code>	2	The marker was already created and has been modified using the provided parameters.
<code>hsAlreadyExists</code>	3	A marker of another type already uses the specified <i>Name</i> .

Remark(s)

By default this method sets the `hsTargetNoEdit` marker and the `hsTargetFixedSize` marker constraints.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.AddTargetMarker _
    ("Target Marker", 60, 280, 30, False)
```

ChangeAxesMarker

Description

Modifies a previously created Axes marker.

Syntax

control.**ChangeAxesMarker**(*Name* As String, *X* As Single, *Y* As Single, *Length1* As Single, *Length2* As Single, *Rotation* As Single, *MidAngle* As Single, *Calibrated* As Boolean) As `hsMarkerStatus`

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.
<i>X</i>	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Y</i>	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Length1</i>	Length of the axe #1 in pixels or in calibrated units.	Single	Minimum: 0 Maximum: unlimited

Name	Description	Type	Range
<i>Length2</i>	Length of the axe #2 in pixels or in calibrated units.	Single	Minimum: 0 Maximum: unlimited
<i>Rotation</i>	Rotation angle of the marker	Single	Any Single value.
<i>MidAngle</i>	Angle between the axes of the marker.	Single	If the UnitsOrientation property is set to hsDegree : Minimum: -180 deg, Maximum: 180 deg otherwise : Minimum: -Pi, Maximum: Pi
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.ChangeAxesMarker _
    ("Axes Marker", 125, 50, 100, 100, 0, 90, False)
```

ChangeCircularArcMarker

Description

Modifies a previously created Circular Arc marker.

Syntax

control.**ChangeCircularArcMarker**(*Name* As String, *X* As Single, *Y* As Single, *Radius* As Single, *Thickness* As Single, *Rotation* As Single, *Opening* As Single, *Calibrated* As Boolean) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the marker.	String	Not applicable.
<i>X</i>	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Y</i>	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.

Name	Description	Type	Range
<i>Radius</i>	Mean radius in pixels or in calibrated units.	Single	Any Single value.
<i>Thickness</i>	Radial distance between the two bounding annuli of the marker.	Single	Any Single value.
<i>Rotation</i>	Angle between bisector of the marker and the X-Axis of the World or Image coordinate system (depending if the marker is calibrated or not.	Single	Any Single value.
<i>Opening</i>	Angle between the two bounding radii of the marker.	Single	If the UnitsOrientation property is set to hsDegree : Minimum: 0 deg, Maximum: 360 deg otherwise : Minimum: 0, Maximum: 2*Pi
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As Integer
Status = Display1.ChangeCircularArcMarker _
    ("Arc Marker", 100, 100, 25, 50, 30, 90, False)
```

ChangeLabelMarker

Description

Modifies a previously created Label marker.

Syntax

control.**ChangeLabelMarker**(Name As String, X As Single, Y As Single, Text As String, Calibrated As Boolean As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the marker	String	Not applicable.

Name	Description	Type	Range
Y	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
X	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Text	Label text string.	String	Not applicable.
Calibrated	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.ChangeLabelMarker _
    ("Label Marker", 230, 420, "Label String", False)
```

ChangeLineMarker

Description

Modifies a previously created Line marker.

Syntax

control.**ChangeLineMarker**(Name As String, X As Single, Y As Single, X2 As Single, Y2 As Single, Calibrated As Boolean) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
X2	"X" end of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y2	"Y" end of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.

Name	Description	Type	Range
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a **hsMarkerStatus**. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.ChangeLineMarker _
    ("Line Marker", 73, 120, 143, 160, False)
```

ChangePointArrayMarker

Description

Modifies an array of Point markers.

Syntax

control.**ChangePointArrayMarker**(*Name* As String, *X* As Single, *Y* As Single, *Size* As Integer, *Calibrated* As Boolean) As **hsMarkerStatus**

Parameter(s)

Name	Description	Type	Range
<i>I</i>	Name of the marker.	String	Not applicable.
<i>X</i>	Default X origin of the markers in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Y</i>	Default Y origin of the markers in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Size</i>	New array size.	Integer	Minimum: 0 Maximum: 1000 Default: 50
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a **hsMarkerStatus**. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = _
    Display1.ChangePointArrayMarker("PointArray",0,0,10,false)
```

ChangePointMarker

Description

Modifies a previously created Point marker.

Syntax

```
control.ChangePointMarker(Name As String, X As Single, Y As Single, Calibrated As Boolean)
As hsMarkerStatus
```

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Calibrated</i>	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.ChangePointMarker _
    ("Point Marker", 350, 175, False)
```

ChangeRectangleMarker

Description

Modifies a previously created Rectangle marker.

Syntax

control.**ChangeRectangleMarker**(Name As String, X As Single, Y As Single, Width As Single, Height As Single, Calibrated As Boolean) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Width	Width in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Height	Height in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Calibrated	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.ChangeRectangleMarker _
    ("Rectangle Marker", 230, 350, 20, 50, False)
```

ChangeTargetMarker

Description

Modifies a previously created Target marker.

Syntax

control.**ChangeTargetMarker**(Name As String, X As Single, Y As Single, Radius As Single, Calibrated As Boolean) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
Name	Name of the marker.	String	Not applicable.
X	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Y	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Radius	Radius of the target in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
Calibrated	Specifies if the marker is positioned and sized in calibrated units with respect to the World coordinate system.	Boolean	True, False

Return

This method returns a `hsMarkerStatus`. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.
hsNotFound	4	The specified name is not valid.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.ChangeTargetMarker _
    ("Target Marker", 60, 280, 30, False)
```

ClearSceneSelection

Description

Resets the parameters for a given scene selection layer. This method resets the following properties to their default values: **SceneSelection**, **SceneSelectionColor**, **SceneSelectionIndex**, **SceneSelectionPenWidth** and **SceneSelectionType**.

Syntax

control.**ClearSceneSelection** (*Index* As Integer) As Boolean

Parameter(s)

Parameter(s)	Description	Type	Range
<i>Index</i>	Index of the layer.	Integer	Minimum: 0 Maximum: SceneSize - 1

Return

Boolean. Specifies if an error occurred while clearing the scene selection layer.

Range

Value	Description
True	The scene selection has been successfully cleared.
False	An error occurred while clearing the scene selection.

Example

```
' Deactivate the scene selection layer 5
Display1.ClearSceneSelection(5)
```

ControlPropertyExchange

Description

Reserved for internal use.

DoClick

Description

Reserved for internal use.

Drag

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetDisplayCoordinates

Description

This method takes two parameters (X and Y). When the method is executed, the two entered parameters are set to express the current mouse position within the display zone. The coordinates are expressed in calibrated units, if they are enabled (see **CalibratedUnitsEnabled**), otherwise they are expressed in pixels.

Syntax

control.**GetDisplayCoordinates**(X As Single, Y As Single)

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the mouse pointer.	Single	Any Single value.
Y	Y coordinate of the mouse pointer.	Single	Any Single value.

Return

No return value.

Example

```
Dim X, Y As Single
```

DisplayControl.**GetDisplayCoordinates** X,Y

GetFreeBackgroundSelection

Description

Reserved for internal use.

HideProperties

Description

Hides the Display properties interface.

Syntax

control.**HideProperties**

Parameter(s)

None.

Return

Boolean. Specifies if the interface was displayed when the method was called.

Range

Value	Description
True	The interface is displayed when the method is called.
False	The interface is not displayed when the method is called.

Example

```
Dim WasDisplayed As Boolean
WasDisplayed = Display1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

Move

Description

Reserved for internal use.

Refresh

Description

Redraws the entire HSDisplay object. This method redraws the image layers, the scene layers, the scene selection layers, the marker layers, as well as the user interface objects (tool bar, scroll bar, status bar, etc.). **Refresh** should only be used if the entire HSDisplay object needs to be redrawn.

Syntax

control.**Refresh**

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Example

Display1.Refresh

RefreshChangedMarker

Description

Refreshes only the changed markers. This method should only be called in marker change event handlers. To be used only with Marker **NoRefresh** constraints; for example **hsLineNoRefresh** (**LineMarkerConstraints**).

Syntax

control.RefreshChangedMarker ()

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Remark(s)

Replaces the **RefreshMarker** method.

Example

```
Private Sub HSDisplay1.PointMarkerChange(_
    ByVal Name As String, ByVal As Double, ByVal Y As Double)
    if x<10 then
        HSDisplay1.MarkerPositionX = 10
    End If
    RefreshChangedMarker
End Sub
```

RefreshDisplay

Description

Redraws the image layers, the scene layers, the scene selection layers and the marker layers. It does not redraw the user interface of the HSDisplay object. For optimal performance, **RefreshDisplay** should be used instead of **Refresh** in the main loop of an application when only the images, scenes, scene selections and/or markers have changed.

Syntax

control.RefreshDisplay

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Example

```
Display1.RefreshDisplay
```

RefreshMarker

Description

Note: HexSight supports the **RefreshMarker** property for only compatibility with applications built in previous versions. You should use the **RefreshChangedMarker**.

RefreshMarkers

Description

Redraws the marker layers only. It does not redraw the user interface of the HSDisplay object. For optimal performance, **RefreshMarkers** should be used instead of **RefreshDisplay** in the main loop of an application when only the markers have changed.

Syntax

```
control.RefreshMarkers
```

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Example

```
Display1.RefreshMarkers
```

RemoveAllMarker

Description

Removes all the markers from the display.

Syntax

```
control.RemoveAllMarker() As hsMarkerStatus
```

Parameter(s)

None.

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.

Example

```
Dim MarkerStatus As hsMarkerStatus
MarkerStatus = Display1.RemoveAllMarker
```

RemoveMarker

Description

Removes a marker from the display.

Syntax

control.**RemoveMarker**(*Name* As String) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the marker	String	Not applicable.

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.

Example

```
Dim MarkerStatus As hsMarkerStatus
MarkerStatus = Display1.RemoveMarker("Axes Marker")
```

RemoveMarkerGroup

Description

Removes the markers belonging to a group from the display.

Syntax

control.**RemoveMarkerGroup**(*Group* As Integer) As hsMarkerStatus

Parameter(s)

Name	Description	Type	Range
<i>Group</i>	Group of marker returned by the MarkerGroup property.	Integer	Minimum: 0 Maximum: unlimited

Return

This method returns a hsMarkerStatus. The applicable range of this constant type, for this method, is described in the following table.

Range

Constant	Value	Description
hsSuccess	0	Operation succeeded.
hsFailed	1	Operation failed.

Example

```
Dim Status As hsMarkerStatus
Status = Display1.RemoveMarkerGroup(1)
```


SaveConfiguration

Description

Reserved for internal use.

SaveDisplay

Description

Saves a bitmap of the current display control.

Syntax

control.**SaveDisplay**(*FileName* As String, *Mode* As hsDisplayMode) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of the file to save.	String	Must be a string.
<i>Mode</i>	Type of snapshot.	hsDisplayMode	hsDisplayWindow, hsWholeWindowSnapshot

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

The saved file is a *bmp* file. with a *hsDisplayMode* of *hsDisplayWindow*, only the client rectangle is captured. Toolbar, rulers, scrollbars and status bar are not visible. If *hsWholeWindowSnapshot* is used, the whole control is captured, with the toolbar, rulers, scrollbars and status bar, if visible. This is similar to a snapshot of the control. Markers, entities, images and scenes displayed are always captured, in all modes.

Example

```
Dim Succeeded As Boolean
Succeeded = Locator1.SaveDisplay("..\Display.bmp", hsDisplayWindow)
```

SetFocus

Description

Reserved for internal use.

SetFocusXY

Description

Sets the center of the Display's output to a specific coordinate.

Syntax

control.**SetFocusXY**(*X* As Single, *Y* As Single)

Parameter(s)

Parameter	Description	Type	Range
<i>X</i>	X coordinate.	Single	Boundaries of the display.
<i>Y</i>	Y coordinate.	Single	Boundaries of the display.

Return

No return value.

Range

Not applicable.

Remark(s)

The parameters X and Y are expressed in the calibrated units specified by the **UnitsLength** property with respect to the World coordinate system if the **CalibratedUnitsEnabled** property is set to True. Otherwise, these parameters are expressed in pixels with respect to the Image coordinate system.

Example

Display1.**SetFocus** 320.2,240.0

SetMarkerPosition

Description

Sets the X and Y position of the selected marker.

Syntax

control.**SetMarkerPosition**(Name As String, X As Single, Y As Single)

Parameter(s)

Parameter	Description	Type	Range
Name	Name of the selected marker.	String	Not applicable.
X	X coordinate of the marker.	Single	Boundaries of the display
Y	Y coordinate of the marker.	Single	Boundaries of the display

Return

No return value.

Range

Not applicable.

Remark(s)

The position of the marker is expressed with respect to a coordinate system (either the Image or World coordinate system), depending on the status of the **MarkerCalibrated** property as shown in the table below. The length units used to express this property is also provided.

MarkerCalibrated property	Reference coordinate system	Length unit used
False	Image	Pixel.
True	World	As defined in UnitsLength property.

Example

Display1.**SetMarkerPosition**("MyMarker",100.5, -50.1)

SetPointArrayMarkerPosition

Description

Sets the X and Y position of the point in the Point Array.

Syntax

control.**SetPointArrayMarkerPosition**(*Name* As String, *Index* As Integer, *X* As Single, *Y* As Single)

Return

No return value.

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Name</i>	Name of the point array marker.	String	Not applicable.
<i>Index</i>	Index of the point marker in the array.	Integer	Minimum:0 Maximum: PointArrayMarkerSize - 1
<i>X</i>	"X" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.
<i>Y</i>	"Y" origin of the marker in pixels or in calibrated units.	Single	Any value inside the boundaries of the image.

Example

```
AddPointArrayMarker("PointArray",0,0,10,false)
Dim Index as Integer
Index = 0
For x = 10 to 30 Step 10
    For y = 10 to 30 Step 10
        SetPointArrayMarkerPosition ("Point",Index,x,y)
    Next y
Next x
```

ShowProperties

Description

Displays the properties interface of the Display control.

Syntax

control.**ShowProperties**(*Modal* As Boolean) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Display1.ShowProperties(True)
```

Events

AxesMarkerChange

Description

Occurs when the Axes marker is modified interactively using the mouse pointer.

Syntax

Sub *control*.**AxesMarkerChange**(*Name* As String, *X* As Double, *Y* As Double, *Length1* As Double, *Length2* As Double, *Rotation* As Double, *MidAngle* As Double)

Parameter(s)

Parameter	Description	Type
<i>Name</i>	Name of the marker.	String
<i>X</i>	"X" origin of the marker.	Double
<i>Y</i>	"Y" origin of the marker.	Double
<i>Length1</i>	Length of the first axis of the marker.	Double
<i>Length2</i>	Length of the second axis of the marker.	Double
<i>Rotation</i>	Rotation angle of the marker.	Double
<i>MidAngle</i>	Angle between the first and the second axis of the marker.	Double

Remark(s)

Parameters *X*, *Y*, *Length1* and *Length2* are expressed in calibrated units specified by the **UnitsLength** property if the **MarkerCalibrated** property is set to True. Otherwise, these parameters are expressed in pixels.

The *Rotation* and *MidAngle* parameters are expressed in degrees if the **UnitsOrientation** property is set to **hsDegree**. Otherwise, they are expressed in radians.

The *Rotation* parameter is defined as the angle between the X-axis of a reference coordinate system and the first axis of the marker. If the **MarkerCalibrated** property is set to True, this reference coordinate system is the World coordinate system, otherwise it is the Image coordinate system.

CalibratedUnitsChange

Description

Occurs when the **CalibratedUnits** state is changed by the user interface.

Syntax

Sub *control*.**CalibratedUnitsChange**()

Parameter(s)

None

CircularArcMarkerChange

Description

Occurs when the CircularArc marker is modified interactively using the mouse pointer.

Syntax

Sub *control*.**CircularArcMarkerChange**(*Name* As String, *X* As Double, *Y* As Double, *RadiusMin* As Double, *RadiusMax* As Double, *Rotation* As Double, *MidAngle* As Double)

Parameter(s)

Parameter	Description	Type
<i>Name</i>	Name of the marker.	String
<i>X</i>	"X" origin of the marker.	Double
<i>Y</i>	"Y" origin of the marker.	Double
<i>RadiusMin</i>	Minimum radius.	Double
<i>RadiusMax</i>	Maximum radius.	Double
<i>Rotation</i>	Rotation angle of the marker.	Double
<i>MidAngle</i>	Angle between the first and the second axis of the marker.	Double

Remark(s)

The status of the **MarkerCalibrated** property defines the length units and the coordinate system with respect to which the parameters *X*, *Y*, *RadiusMin* and *RadiusMax* are expressed, as shown in the following table:

MarkerCalibrated property	Units used to express the parameter	Coordinate system (applicable for the parameters <i>X</i> and <i>Y</i>)
True	As specified by the UnitsLength property.	World
False	Pixels.	Image

The *Rotation* and *MidAngle* parameters are expressed in degrees if the **UnitsOrientation** property is set to `hsDegree`. Otherwise, they are expressed in radians.

The *Rotation* parameter is defined as the angle between the X-axis of coordinate of a reference coordinate system and the first axis of the marker. If the **MarkerCalibrated** property is set to `True`, this reference coordinate system is the World coordinate system, otherwise it is the Image coordinate system.

Click

Description

Stock event fired when the control captures the mouse, any **BUTTONUP** (left, middle, or right) message is received, and the button is released over the control. The **DisplayMouseDown** and **DisplayMouseUp** events occur before this event.

Syntax

Sub *control*.**Click**()

DbIClick

Description

Stock event similar to **Click** but fired when a **BUTTONDBLCLK** (left, middle, or right) message is received.

Syntax

Sub *control*.**DbIClick**()

DisplayMouseDown

Description

Stock event fired if any **BUTTONDOWN** (left, middle, or right) message is received. The mouse is captured immediately before this event is fired.

Syntax

Sub *control*.**DisplayMouseDown**(*Button* As Integer, *ShiftState* As Integer, *X* As Double, *Y* As Double)

Parameter(s)

Parameter	Description	Type
<i>Button</i>	The numeric value of the mouse button pressed.	Integer
<i>ShiftState</i>	Combination of key flags	Integer
<i>X</i>	X coordinate of the mouse pointer.	Double
<i>Y</i>	Y coordinate of the mouse pointer.	Double

Remark(s)

The *Button* parameter returns an integer that identifies the button that was pressed to cause the event. The *Button* parameter is a bit field with bits corresponding to the left button (bit 0), right button (bit 1), and middle button (bit 2). These bits correspond to the values 1, 2, and 4, respectively. Only one of the bits is set, indicating the button that caused the event. The *ShiftState* parameter returns an integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the button specified in the *Button* parameter is pressed or released. A bit is set if the key is down. The *ShiftState* parameter is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The *ShiftState* parameter indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed. For example, if both CTRL and ALT were pressed, the value of *ShiftState* would be 6. The *X* and *Y* parameters are expressed in calibrated units, if they are enabled (see **CalibratedUnitsEnabled**), otherwise they are expressed in pixels.

DisplayMouseMove

Description

Stock event fired when a **WM_MOUSEMOVE** message is received.

Syntax

Sub *control*.**DisplayMouseMove**(*Button* As Integer, *ShiftState* As Integer, *X* As Double, *Y* As Double)

Parameter(s)

Parameter	Description	Type
<i>Button</i>	The numeric value of the mouse button pressed.	Integer
<i>ShiftState</i>	Combination of key flags	Integer
<i>X</i>	X coordinate of the mouse pointer.	Double
<i>Y</i>	Y coordinate of the mouse pointer.	Double

Remark(s)

See **DisplayMouseDown** event for a description of each parameter.

DisplayMouseUp

Description

Stock event fired if any **BUTTONUP** (left, middle, or right) message is received. The mouse capture is released before this event is fired.

Syntax

Sub *control*.**DisplayMouseUp**(*Button* As Integer, *ShiftState* As Integer, *X* As Double, *Y* As Double)

Parameter(s)

Parameter	Description	Type
<i>Button</i>	The numeric value of the mouse button pressed.	Integer
<i>ShiftState</i>	Combination of key flags	Integer
<i>X</i>	X coordinate of the mouse pointer.	Double
<i>Y</i>	Y coordinate of the mouse pointer.	Double

Remark(s)

See **DisplayMouseDown** event for a description of each parameter.

EntitySelectionChange

Description

Occurs when an entity is selected. An event is fired for each entity at the selected position or each entity in the selected area. Can be enabled or disabled with

InteractiveSelectionEnabled.

Syntax

Sub *control*.**EntitySelectionChange**()

Parameter(s)

None

Error

Description

Reserved for internal use.

FocusChange

Description

Occurs when the **Focus** is changed by the user interface.

Syntax

Sub *control*.**FocusChange**()

Parameter(s)

None

InteractionModeChange

Description

Occurs when a Scene is selected. An event is fired for each entity at the selected position or each entity in the selected area. Can be enabled or disabled with **InteractiveSelectionEnabled**.

Syntax

Sub *control*.**InteractionModeChange**()

Parameter(s)

None

InteractiveSelectionChange

Description

Occurs when the interactive selection is changed by the user interface.

Syntax

Sub *control*.**InteractiveSelectionChange**()

Parameter(s)

None

KeyDown

Description

Stock event fired when a **WM_SYSKEYDOWN** or **WM_KEYDOWN** message is received.

Syntax

Sub *control*.**KeyDown**(*KeyCode* As Integer, *Shift* As Integer)

Parameter(s)

Parameter	Description	Type
<i>KeyCode</i>	A key code, such as vbKeyF1 or vbKeyHome .	Integer
<i>Shift</i>	Combination of key flags	Integer

Remark(s)

The *Shift* parameter returns an integer that corresponds to the state of the SHIFT, CTRL, and ALT keys when the key specified in the *KeyCode* parameter is pressed or released. A bit is set if the key is down. The *Shift* parameter is a bit field with the least-significant bits corresponding to the SHIFT key (bit 0), the CTRL key (bit 1), and the ALT key (bit 2). These bits correspond to the values 1, 2, and 4, respectively. The *Shift* parameter indicates the state of these keys. Some, all, or none of the bits can be set, indicating that some, all, or none of

the keys are pressed. For example, if both CTRL and ALT were pressed, the value of *Shift* would be 6.

KeyPress

Description

Stock event fired when a **WM_CHAR** message is received.

Syntax

Sub *control*.**KeyPress**(*KeyAscii* As Integer)

Parameter(s)

Parameter	Description	Type
<i>KeyAscii</i>	ASCII key code of pressed key.	Integer

KeyUp

Description

Stock event fired when a **WM_SYSKEYUP** or **WM_KEYUP** message is received.

Syntax

Sub *control*.**KeyUp**(*KeyCode* As Integer, *Shift* As Integer)

Parameter(s)

Parameter	Description	Type
<i>KeyCode</i>	A key code, such as vbKeyF1 or vbKeyHome .	Integer
<i>Shift</i>	Combination of key flags	Integer

Remark(s)

See **KeyDown** event for a description of each parameter.

LineMarkerChange

Description

Occurs when the Line marker is modified interactively using the mouse pointer.

Syntax

Sub *control*.**LineMarkerChange**(*Name* As String, *X* As Double, *Y* As Double, *X2* As Double, *Y2* As Double)

Parameter(s)

Parameter	Description	Type
<i>Name</i>	Name of the marker.	String
<i>X</i>	"X" origin of the marker.	Single
<i>Y</i>	"Y" origin of the marker.	Single
<i>X2</i>	"X" end of the marker.	Single
<i>Y2</i>	"Y" end of the marker.	Single

Remark(s)

The status of the **MarkerCalibrated** property defines the length units and the coordinate system with respect to which the parameters X, Y, X2 and Y2 are expressed, as shown in the following table.

MarkerCalibrated property	Units used to express the parameter	Coordinate system
True	As specified by the UnitsLength property.	World
False	Pixels.	Image

PointMarkerChange

Description

Occurs when the Point marker is modified interactively using the mouse pointer.

Syntax

Sub *control*.**PointMarkerChange**(Name As String, X As Double, Y As Double)

Parameter(s)

Parameter	Description	Type
Name	Name of the marker.	String
X	"X" origin of the marker.	Single
Y	"Y" origin of the marker.	Single

Remark(s)

The status of the **MarkerCalibrated** property defines the length units and the coordinate system with respect to which the parameters X and Y are expressed, as shown in the following table.

MarkerCalibrated property	Units used to express the parameter	Coordinate system
True	As specified by the UnitsLength property.	World
False	Pixels.	Image

PropertiesChange

Description

Reserved for internal use.

RectangleMarkerChange

Description

Occurs when the Rectangle marker is modified interactively using the mouse pointer.

Syntax

Sub *control*.**RectangleMarkerChange**(Name As String, X As Double, Y As Double, Width As Double, Height As Double, Rotation As Double, Skew As Double)

Parameter(s)

Parameter	Description	Type
<i>Name</i>	Name of the marker.	String
<i>X</i>	"X" origin of the marker.	Single
<i>Y</i>	"Y" origin of the marker.	Single
<i>Width</i>	"X" end of the marker.	Single
<i>Height</i>	"Y" end of the marker.	Single
<i>Rotation</i>	Rotation angle of the marker.	Single
<i>Skew</i>	Skew angle of the marker.	Single

Remark(s)

The status of the **MarkerCalibrated** property defines the length units and the coordinate system with respect to which the parameters *X*, *Y*, *Width* and *Height* are expressed, as shown in the following table.

MarkerCalibrated property	Units used to express the parameter	Coordinate system (applicable only to parameters <i>X</i> and <i>Y</i>)
True	As specified by the UnitsLength property.	World
False	Pixels.	Image

The *Rotation* and *Skew* parameters are expressed in degrees if the **UnitsOrientation** property is set to `hsDegree`. Otherwise, they are expressed in radians.

The *Rotation* parameter is defined as the angle between the X-axis of a reference coordinate system and one of the horizontal lines of the marker. If the **MarkerCalibrated** property is set to `True`, this reference coordinate system is the World coordinate system, otherwise it is the Image coordinate system.

TargetMarkerChange

Description

Occurs when the Target marker is modified interactively using the mouse pointer.

Syntax

Sub *control*.**TargetMarkerChange**(*Name* As String, *X* As Double, *Y* As Double)

Parameter(s)

Parameter	Description	Type
<i>Name</i>	Name of the marker.	String
<i>X</i>	"X" origin of the marker.	Single
<i>Y</i>	"Y" origin of the marker.	Single

Remark(s)

The status of the **MarkerCalibrated** property defines the length units and the coordinate system with respect to which the parameters *X*, *Y*, *Width* and *Height* are expressed, as shown in the following table.

MarkerCalibrated property	Units used to express the parameter	Coordinate system
True	As specified by the UnitsLength property.	World
False	Pixels.	Image

ZoomChange

Description

Reserved for internal use.

Edge Locator

Description

The Edge Locator finds and locates edges meeting user-defined criteria such as strength and polarity. The Edge Locator also scores and sorts located edges.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
```

```
Result = EdgeLocator1.BilinearInterpolationEnabled
```

```
EdgeLocator2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is Model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = EdgeLocator1.CalibratedUnitsEnabled
EdgeLocator2.CalibratedUnitsEnabled = False
```

Constraints

Description

Bitwise field used to select the edge detection constraints. Constraints can be set for position and/or magnitude and are used to score edges.

Syntax

control.**Constraints** [= Value As hsConstraints]

Type

hsConstraints

Range

The valid range for this property is as follows:

Constant	Value	Description
hsNone	0	No constraint.
hsPosition	1	Position constraint.
hsMagnitude	2	Magnitude constraint.
hsAllConstraints	3	Magnitude and position constraints.

Example

```
Dim EdgeConstraint As Integer
EdgeConstraint = EdgeLocator1.Constraints
EdgeLocator2.Constraints = hsMagnitude
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.

Constant	Value	Description
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input Image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is Model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to Tool coordinate system.

Example

```
Dim Result As hsCoordinateSystem
Result = EdgeLocator1.CoordinateSystem
EdgeLocator1.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = EdgeLocator1.Description
```

EdgeCount

Description

Number of edges detected by the tool. Read only.

Syntax

control.**EdgeCount**

Type

Integer

Range

Minimum: 0

Maximum: unlimited

Example

Dim Count As Integer

Count = EdgeLocator1.**EdgeCount**

EdgeMagnitude

Description

Magnitude of the selected edge. Read only.

Syntax

control.**EdgeMagnitude** (*Index* As Integer)

Type

Single

Range

Minimum: -255

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount -1

Example

Dim Result As Single

Result = EdgeLocator1.**EdgeMagnitude**(0)

EdgeMagnitudeScore

Description

Magnitude score of the selected edge. Read only.

Syntax

control.**EdgeMagnitudeScore** (*Index* As Integer)

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount-1

Example

```
Dim Score As Single
Score = EdgeLocator1.EdgeMagnitude(0)
```

EdgePositionScore

Description

Position score of the selected edge. Read only.

Syntax

control.**EdgePositionScore** (*Index* As Integer)

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount-1

Example

```
Dim Score As Single
Score = EdgeLocator1.EdgePositionScore(0)
```

EdgePositionX

Description

X coordinate of the center of the selected edge in the currently selected coordinate system.
Read only.

Syntax

control.**EdgePositionX** (*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount-1

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or Model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim PosX As Single
PosX = EdgeLocator1.EdgePositionX(0)
```

EdgePositionY

Description

Y coordinate of the center of the selected edge in the currently selected coordinate system.
Read only.

Syntax

control.**EdgePositionY** (*Index* As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount -1

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or Model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim PosX As Single
PosX = EdgeLocator1.EdgePositionY(0)
```

EdgeRotation

Description

Rotation of the selected edge in the currently selected coordinate system. Read only.

Syntax

control.**EdgeRotation** (*Index* As Integer)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	EdgeRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount -1

Remark(s)

The rotation is defined as the angle between the X-axis of the active coordinate system (specified by the **CoordinateSystem** property) and the selected edge, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation As Single
Rotation = EdgeLocator1.EdgeRotationX(0)
```

EdgeScore

Description

Score of the selected edge. The score is computed according the constraints set by the **Constraints** property. Read only.

Syntax

control.**EdgeScore** (*Index* As Integer)

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the selected edge.	Single	Minimum: 0 Maximum: EdgeCount -1

Example

```
Dim Score As Single
Score = EdgeLocator1.EdgeScore ( 0 )
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent PatternLocator1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

FilterHalfWidth

Description

Half-width of the convolution filter used to compute the edge magnitude curve from which actual edges are detected. The filter approximates the first derivative of the projection curve. The half width of the filter should be set in order to match the width of the edge in the projection curve (the extent of the grey-scale transition, expressed in number of pixels).

Syntax

control.**FilterHalfWidth** [= Value As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 25

Example

```
Dim HalfWidth As Integer
HalfWidth = EdgeLocator1.FilterHalfWidth
EdgeLocator2.FilterHalfWidth = 2
```

InputImage

InputGreyScaleImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

control.InputImage [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = EdgeLocator1.InputImage
EdgeLocator2.InputImage = "image1"
```

InputView

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **ModelTransformEnabled** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = EdgeLocator1.InputView
EdgeLocator2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the Model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.InputInstanceScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = EdgeLocator1.InputInstanceScene
EdgeLocator2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View within which is retrieved the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = EdgeLocator1.InputInstanceView
EdgeLocator2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = EdgeLocator1.InputModelScene
EdgeLocator2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details.

Syntax

control.**InputModelView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = EdgeLocator1.InputModelView
EdgeLocator2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

Specifies if the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property.

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user controlled.

Remark(s)

This property is only significant if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Enabled As Boolean
Enabled = EdgeLocator1.InstanceSelectionEnabled
EdgeLocator2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **InstanceSelectionEnabled** property is set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0
Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = EdgeLocator1.InstanceSelectionIndex
EdgeLocator2.InstanceSelectionIndex = 0
```

MagnitudeConstraint

Description

Indexed property used to set the magnitude constraint function. Two points are used: Base and Top.

Syntax

control.**MagnitudeConstraint** (*Index* As hsMagnitudeConstraintIndex) [= *Value* As Single]

Type

Integer

Range

Minimum: 0

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	One of the two points of the magnitude constraint function.	hsMagnitudeConstraintIndex	hsBase: 1 hsTop: 2

Example

```
Dim MagCt As Integer
MagCt = EdgeLocator1.MagnitudeConstraint(hsTop)
EdgeLocator2.MagnitudeConstraint(hsBase) = 75
```

MagnitudeThreshold

Description

Magnitude threshold is used to find edges on the magnitude curve. A subpixel peak detection algorithm is applied on the region of every minimum or maximum of the curve that exceeds this threshold in order to locate edges.

Syntax

control.**MagnitudeThreshold** [= *Value* As Single]

Type

Single

Range

Single: 0

Maximum: 255

Example

```
Dim MagTh As Single
MagTh = EdgeLocator1.MagnitudeThreshold
EdgeLocator2.MagnitudeThreshold = 50.0
```

ModelTransformEnabled

Description

Specifies if the tool is Model-based. This means that the tool is defined on a Model Image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.ModelTransformEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The control is Model-based.
False	The control is not Model-based.

Example

```
Dim Enabled As Boolean
Enabled = EdgeLocator1.ModelTransformEnabled
EdgeLocator2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.Name [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = EdgeLocator1.Name
EdgeLocator2.Name = "EdgeLocA"
```

Polarity

Description

Edge selection criterion. The grey-scale transition of the edge must respect the polarity set by this property.

Syntax

control.Polarity [= Value As hsPolarity]

Type

hsPolarity

Range

The valid range for this property is as follows:

Name	Value	Description
hsDarkToLight	0	The greylevel value must go from dark to light when crossing an edge.
hsLightToDark	1	The greylevel value must go from light to dark when crossing an edge.

Name	Value	Description
hsEitherPolarity	2	The change in greylevel value change is not a criterion for locating an edge.

Example

```
Dim Pol As hsPolarity
Pol = EdgeLocator1.Polarity
EdgeLocator2.Polarity = hsLightToDark
```

PositionConstraint

Description

Indexed property used to set the position constraint function. Four points are used: Base Left, Top Left, Top Right and Base Right.

Syntax

control.**PositionConstraint** (*Index* As hsPositionConstraintIndex) [= *NewValue* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	One of the four points of the position constraint function.	hsPositionConstraintIndex	hsBaseLeft: 1 hsTopLeft: 2 hsTopRight: 3 hsBaseRight: 4

Remark(s)

A position constraint value of 0.0 represents the left of the tool while 1.0 represents the right of the tool.

Example

```
Dim PosCt As Single
PosCt = EdgeLocator1.PositionConstraint (hsTopRight)
EdgeLocator2.PositionConstraint(hsTopLeft) = 0.3
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be performed using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SamplingStep

Description

Step used in the tool's last execution to sample the rectangle from the input image. When the tool is Model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

```
control.SamplingStep
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = EdgeLocator1.SamplingStep
```

SamplingStepCustom

Description

Sampling-step factor defined by the user to override the default sampling step that can be computed by the tool. This value is used only if the **SamplingStepCustomEnabled** property is set to True.

Syntax

```
control.SamplingStepCustom [= Value As Single]
```

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

Dim Step As Single

Step = EdgeLocator1.**SamplingStepCustom**EdgeLocator2.**SamplingStepCustom** = 2.1

SamplingStepCustomEnabled

DescriptionValue that specifies if the default sampling step is overridden by the **SamplingStepCustom** property.**Syntax***control*.**SamplingStepCustomEnabled** [= Value As Boolean]**Type**

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Remark(s)

A custom sampling step is usually not recommended.

Example

Dim Enabled As Boolean

Enabled = EdgeLocator1.**SamplingStepCustomEnabled**EdgeLocator2.**SamplingStepCustomEnabled** = True

SamplingStepDefault

Description

Default sampling step for the current input Image. Read only.

Syntax*control*.**SamplingStepDefault****Type**

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Example

Dim Sampling As Single

Sampling = EdgeLocator.**SamplingStepDefault**

ScoreThreshold

Description

Minimum score to accept an edge. The score of an edge is returned by the **EdgeScore** property.

Syntax

control.**ScoreThreshold** [= *Value* As Single]

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Example

```
Dim ScoreTh As Single
ScoreTh = EdgeLocator1.ScoreThreshold
EdgeLocator2.ScoreThreshold = 0.65
```

SortResultsEnabled

Description

Property that specifies if edges are sorted in descending order of score values.

Syntax

control.**SortResultsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The edges are not sorted.
True	The edges are sorted in descending order of score values.

Example

```
Dim SortEnabled As Boolean
SortEnabled = EdgeLocator1.SortResultsEnabled
EdgeLocator2.SortResultsEnabled = False
```

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.**ToolHeight** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = EdgeLocator1.ToolHeight
EdgeLocator2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionX** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim X_Pos As Single
X_Pos = EdgeLocator1.ToolPositionX
EdgeLocator2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = EdgeLocator1.ToolPositionY
EdgeLocator2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = EdgeLocator1.ToolRotation
EdgeLocator2.ToolRotation = -10.0
```

ToolSkew

Description

Skew angle of the tool rectangle.

Syntax

control.**ToolSkew** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolSkew property range
hsDegree	Minimum: -90 deg Maximum: +90 deg
hsRadian	Minimum: -Pi/2 rad Maximum: +Pi/2 rad

Example

```
Dim Skew As Single
Skew = EdgeLocator1.ToolSkew
EdgeLocator1.ToolSkew = 33.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = EdgeLocator1.ToolWidth
EdgeLocator2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale,

Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = EdgeLocator1.TransformRotation
EdgeLocator2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of a transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero
Maximum: unlimited

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = EdgeLocator1.TransformScale
EdgeLocator2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in the length units specified by the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = EdgeLocator1.TransformTranslationX
EdgeLocator2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the control if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in the length units specified by the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = EdgeLocator1.TransformTranslationY
EdgeLocator2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSEdgeLocator".

Example

```
Dim ControlType As String
ControlType = EdgeLocator1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= Value As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Remark(s)

This property is only applicable if **ModelTransformEnabled** or **CalibratedUnitsEnabled** is set to True.

Example

```
Dim Unit As hsUnitsLength
Unit = EdgeLocator1.UnitsLength
EdgeLocator2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit

Syntax

control.**UnitsOrientation** [= Value As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = EdgeLocator1.UnitsOrientation
EdgeLocator2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)"

Example

```
Dim Version As String
Version = EdgeLocator1.Version
```

Methods

AboutBox

Description

Displays the Edge Locator tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

ExampleEdgeLocator1.**AboutBox**

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Edge Locator process.

Syntax*control*.**Execute**()**Parameter(s)**

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

ExampleEdgeLocator1.**Execute**

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax*control*.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat**Parameter(s)**

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax*control*.**HideProperties**()**Parameter(s)**

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = EdgeLocator1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax*control*.**ShowProperties**(*Modal* As Boolean, [*Variant Option*]) As Boolean

Parameters

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the table below.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked.
hsExecuteRemaining	64	The tool and all subsequent tools in the Process Manager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Dim Success As Boolean
Success = EdgeLocator1.ShowProperties _
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

EvaluatorTool

Description

The Evaluator tool converts a specific result from any other tool as a Pass or Fail value, in order to trigger an event with an upcoming OutputTool, depending on the decision result. Comparisons and mathematical operations are available. Multiple Evaluator tools can be used to obtain a final decision.

Properties

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = OutputTool1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = OutputTool1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & OutputTool1.ErrorObject.Description
```

LeftInputConstant

Description

Constant to use as left operand. Ignored if LeftInputType is set to hsInputTypeTool.

Syntax

control.**LeftInputConstant** [= *Value* As Single]

Type

Single

Range

Not applicable.

Example

```
Dim Constant As Single
Constant = Evaluator1.LeftInputConstant
```

LeftInputToolFunction

Description

Name of the function used to extract a value from another tool.

Syntax

control.**LeftInputToolFunction** [= *Value* As String]

Type

String

Range

Any available property from the tool referenced by the LeftInputToolName property.

Example

```
Evaluator2.LeftInputToolName = "Evaluator1"
Evaluator2.LeftInputToolFunction = "Result"
```

LeftInputToolFunctionIndex

Description

Index used as first parameter for the property referenced by LeftInputToolFunction. This parameter is optional, only used if the LeftInputToolFunction is indexed.

Syntax

```
control.LeftInputToolFunctionIndex [= Index As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Maximum number of index for LeftInputToolFunction

Example

```
Evaluator1.LeftInputToolName = "HSLocator"
Evaluator1.LeftInputToolFunction = "InstanceMatchQuality"
Evaluator1.LeftInputToolFunctionIndex = 3
```

LeftInputToolName

Description

Name of the tool used to extract a value from a specific function.

Syntax

```
control.LeftInputToolName [= Value As String]
```

Type

String

Range

Any available tool from the Process Manager. The tool should provide at least one property visible to the EvaluatorTool.

Example

```
Evaluator2.LeftInputToolName = "ImageHistogram"
Evaluator2.LeftInputToolFunction = "Mean"
```

LeftInputType

Description

The type of input used for the left operand.

Syntax

```
control.LeftInputType
```

Type

String

Range

The valid range for this property is as follows:

hsInputTypeConstant	0	Operand defined as a constant
hsInputTypeTool	1	Operand referencing a tool result

Example

```
Dim InputType As hsInputType
InputType = Evaluator1.LeftInputType
Evaluator1.LeftInputType = hsInputTypeConstant
```

LeftResult

Description

The evaluated result for the left operand.

Syntax

control.**LeftResult**

Type

Single

Range

Not applicable.

Example

```
Dim Result As Single
Result = Evaluator1.LeftResult
```

Mode

Description

Determines the current mode of the Evaluator tool.

Syntax

control.**Mode**(= *Value* As hsEvaluatorToolMode)

Type

hsEvaluatorToolMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsModeGreaterThan	0x1001	Result is True if the left operand is greater than the right operand.
hsModeGreaterThanOrEqualTo	0x1002	Result is True if the left operand is greater than or equal to the right operand.
hsModeLessThan	0x1003	Result is True if the left operand is less than the right operand.
hsModeLessThanOrEqualTo	0x1004	Result is True if the left operand is less than or equal to the right operand.
hsModeEqualTo	0x1005	Result is True if the left operand is equal to the right operand.
hsModeNOT	0x1006	Result is the negative of the left operand. Right operand is ignored.
hsModeAND	0x1007	Result of the And operation of the left and right operands.
hsModeNAND	0x1008	Result is the negative of the And operation of the left and right operands.

Constant	Value	Description
hsModeOR	0x1009	Result of the Or operation of the left and right operands.
hsModeXOR	0x100A	Result of the exclusive Or operation of the left and right operands.
hsModeAddition	0x100B	Result of the addition of the left and the right operands.
hsModeSubtraction	0x100C	Result of the subtraction of the left operand by the right operand.
hsModeMultiplication	0x100D	Result of the multiplication of the left and the right operands.
hsModeDivision	0x100E	Result of the division of the left operand by the right operand.
hsModeIndex	0x100F	Result is the right operand with the index being the evaluated result of the left operand.

Example

```
Evaluator.Mode = hsModeEqualTo
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

```
control.Name [= Value As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ProcessName As String
ProcessName = Evaluator.Name
Evaluator.Name = "EqualityTest"
```

ProcessManager

Description

Reserved for internal use.

Result

Description

The evaluated result for the operation defined by Mode.

Syntax

```
control.Result
```

Type

Single

Range

Not applicable.

Example

```
Dim Result As Single
Result = Evaluator1.Result
```

RightInputConstant

Description

Constant to use as right operand. Ignored if RightInputType is set to hsInputTypeTool.

Syntax

control.**RightInputConstant** [= *Value* As Single]

Type

Single

Range

Not applicable.

Example

```
Dim Constant As Single
Constant = Evaluator1.RightInputConstant
```

RightInputToolFunction

Description

Name of the function used to extract a value from another tool.

Syntax

control.**RightInputToolFunction** [= *Value* As String]

Type

String

Range

Any available property from the tool referenced by the RightInputToolName property.

Example

```
Evaluator2.RightInputToolName = "Evaluator1"
Evaluator2.RightInputToolFunction = "Result"
```

RightInputToolFunctionIndex

Description

Index used as first parameter for the property referenced by RightInputToolFunction. This parameter is optional, only used if the RightInputToolFunction is indexed.

Syntax

control.**RightInputToolFunctionIndex** [= *Index* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Maximum number of index for RightInputToolFunction

Example

```
Evaluator1.RightInputToolName = "HSLocator"
Evaluator1.RightInputToolFunction = "InstanceMatchQuality"
Evaluator1.RightInputToolFunctionIndex = 3
```

RightInputToolName

Description

Name of the tool used to extract a value from a specific function.

Syntax

control.**RightInputToolName** [= Value As String]

Type

String

Range

Any available tool from the Process Manager. The tool should provide at least one property visible to the EvaluatorTool.

Example

```
Evaluator2.RightInputToolName = "ImageHistogram"
Evaluator2.RightInputToolFunction = "Mean"
```

RightInputType

Description

The type of input used for the Right operand.

Syntax

control.**RightInputType**

Type

String

Range

The valid range for this property is as follows:

hsInputTypeConstant	0	Operand defined as a constant
hsInputTypeTool	1	Operand referencing a tool result

Example

```
Dim InputType As hsInputType
InputType = Evaluator1.RightInputType
Evaluator1.RightInputType = hsInputTypeConstant
```

RightResult

Description

The evaluated result for the right operand.

Syntax

control.**RightResult**

Type

Single

Range

Not applicable.

Example

```
Dim Result As Single
Result = Evaluator1.RightResult
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSEvaluatorTool".

Example

```
Dim ControlType As String
ControlType = Evaluator1.TypeName
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Evaluator1.Version
```

Methods

AboutBox

Description

Displays the tool **About** box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Return

No return value.

Example

Evaluator1.AboutBox

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool in the current Mode.

Syntax

control.Execute()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

Evaluator1.Execute

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the tool properties interface.

Syntax

control.HideProperties()

Parameter(s)

None

Return

Boolean

Range

Value	Description
True	The interface was displayed when the method was called.
False	The interface was not displayed.

Example

Dim InterfaceWasDisplayed As Boolean

InterfaceWasDisplayed = Evaluator1.**HideProperties**

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the table below.

<i>Option</i> parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool process is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool process and the subsequent tools in the ProcessManager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = Evaluator1.ShowProperties _
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

OCRFixedFont

Description

The OCR Fixed Font tool recognizes and reads fixed-font character strings.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* as Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result as Boolean
Result = OCRFixedFont1.BilinearInterpolationEnabled
OCRFixedFont2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* as Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result as Boolean
Result = OCRFixedFont1.CalibratedUnitsEnabled
OCRFixedFont2.CalibratedUnitsEnabled = False
```

CharacterConfidence

Description

Confidence value for each read character. This value reflects the match percentage between the actual character and its ideal shape. Read only.

Syntax

control.**CharacterConfidence**(RowIndex As Integer, CharIndex As Integer)

Type

Integer

Range

Minimum: 0

Maximum: 100

Parameter(s)

Name	Description	Type	Range
<i>RowIndex</i>	Index of the row.	Integer	0,1,2,3
<i>CharIndex</i>	Index of the character in the row.	Integer	Minimum: 0 Maximum: 29

Example

```
Dim Result As String
Dim Confidence() As Integer
Dim i As Integer
' Get the string and the confidence of each character
OCRFixedFont1.Execute
Result = OCRFixedFont1.String(0)
ReDim Confidence(Len(Result))
For i = 0 To Len(Result) - 1
    Confidence(i) = OCRFixedFont1.CharacterConfidence(0,i)
Next i
```

CharacterCornerX

Description

X coordinate of one of the 4 indexed corners of a given character instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the length units specified by the **UnitsLength** property when the World, the Object or the Tool coordinate system is selected. Expressed in pixels when the Image coordinate system is selected. In the exceptional case where the tool is not configured to work with calibrated units

(**CalibratedUnitsEnabled** and **ModelTransformEnabled** properties are both set to False), values returned in the Tool coordinate system are expressed in pixels, and values cannot be returned in the World or Object coordinate systems. Read only.

Syntax

control.**CharacterCornerX**(*RowIndex* As Integer, *CharIndex* As Integer, *CornerIndex* As Integer)

Type

Single

Range

Any value inside the boundaries of the selected coordinate system.

Parameter(s)

Name	Description	Type	Range
<i>RowIndex</i>	Index of the row.	Integer	0,1,2,3
<i>CharIndex</i>	Index of the character in the row.	Integer	Minimum: 0 Maximum: 29
<i>CornerIndex</i>	Index of the corner of the character.	Integer	0,1,2,3

Example

```
Dim Result As String
Dim Confidence() As Integer
Dim i As Integer
' Get the string and draw a green box around each character
OCRFixedFont1.CoordinateSystem = hsCoordinateSystemImage
OCRFixedFont1.Execute
Result = OCRFixedFont1.String(0)
ReDim Confidence(Len(Result))
For i = 0 To Len(Result) - 1
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,0), _
        OCRFixedFont1.CharacterCornerY(0,i,0), _
        OCRFixedFont1.CharacterCornerX(0,i,1), _
        OCRFixedFont1.CharacterCornerY(0,i,1), _
        Green
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,1), _
        OCRFixedFont1.CharacterCornerY(0,i,1), _
        OCRFixedFont1.CharacterCornerX(0,i,2), _
        OCRFixedFont1.CharacterCornerY(0,i,2), _
        Green
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,2), _
        OCRFixedFont1.CharacterCornerY(0,i,2), _
        OCRFixedFont1.CharacterCornerX(0,i,3), _
        OCRFixedFont1.CharacterCornerY(0,i,3), _
        Green
```

```

        DrawLine OCRFixedFont1.CharacterCornerX(0,i,3), _
                OCRFixedFont1.CharacterCornerY(0,i,3), _
                OCRFixedFont1.CharacterCornerX(0,i,0), _
                OCRFixedFont1.CharacterCornerY(0,i,0), _
                Green
    Next i

```

CharacterCornerY

Description

Y coordinate of one of the 4 indexed corners of a given character instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the length units specified by the **UnitsLength** property when the World, the Object or the Tool coordinate system is selected. Expressed in pixels when the Image coordinate system is selected. In the exceptional case where the tool is not configured to work with calibrated units (**CalibratedUnitsEnabled** and **ModelTransformEnabled** properties are both set to False), values returned in the Tool coordinate system are expressed in pixels, and values cannot be returned in the World or Object coordinate systems. Read only.

Syntax

control.**CharacterCornerY**(RowIndex As Integer, CharIndex As Integer, CornerIndex As Integer)

Type

Single

Range

Any value inside the boundaries of the selected coordinate system.

Parameter(s)

Name	Description	Type	Range
<i>RowIndex</i>	Index of the row.	Integer	0,1,2,3
<i>CharIndex</i>	Index of the character in the row.	Integer	Minimum: 0 Maximum: 29
<i>CornerIndex</i>	Index of the corner of the character.	Integer	0,1,2,3

Example

```

Dim Result As String
Dim Confidence() As Integer
Dim i As Integer
' Get the string and draw a green box around each character
OCRFixedFont1.CoordinateSystem = hsCoordinateSystemImage
OCRFixedFont1.Execute
Result = OCRFixedFont1.String(0)
ReDim Confidence(Len(Result))
For i = 0 To Len(Result) - 1
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,0), _
            OCRFixedFont1.CharacterCornerY(0,i,0), _

```

```

        OCRFixedFont1.CharacterCornerX(0,i,1), _
        OCRFixedFont1.CharacterCornerY(0,i,1), _
        Green
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,1), _
        OCRFixedFont1.CharacterCornerY(0,i,1), _
        OCRFixedFont1.CharacterCornerX(0,i,2), _
        OCRFixedFont1.CharacterCornerY(0,i,2), _
        Green
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,2), _
        OCRFixedFont1.CharacterCornerY(0,i,2), _
        OCRFixedFont1.CharacterCornerX(0,i,3), _
        OCRFixedFont1.CharacterCornerY(0,i,3), _
        Green
    DrawLine OCRFixedFont1.CharacterCornerX(0,i,3), _
        OCRFixedFont1.CharacterCornerY(0,i,3), _
        OCRFixedFont1.CharacterCornerX(0,i,0), _
        OCRFixedFont1.CharacterCornerY(0,i,0), _
        Green
Next i

```

CharacterHeight

Description

Height of the characters, in pixels, in the string to read. Calculated from the characters in the training string, after a successful **Learn**. This property should never be modified manually.

Syntax

control.**CharacterHeight** [= *Value* As Integer]

Type

Integer

Range

Minimum: Height of the primary character box minus a few pixels

Maximum: Height of the primary character box plus a few pixels

Example

```

Dim CharacterHeight As Integer
Dim CharacterWidth As Integer
Dim CharacterSpacing As Integer
Dim StrokeWidth As Integer

' Get the learned character dimensions

CharacterHeight = OCRFixedFont1.CharacterHeight
CharacterWidth = OCRFixedFont1.CharacterWidth
CharacterSpacing = OCRFixedFont1.CharacterSpacing

```



```
StrokeWidth = OCRFixedFont1.StrokeWidth
```

CharactersOnRow

Description

Number of characters to read in a given row. This property is auto-configured after a successful **Learn**.

Syntax

```
control.CharactersOnRow(RowIndex As Integer) [= Value As Integer]
```

Type

Integer

Range

Minimum: 2

Maximum: 30

Parameter(s)

Name	Description	Type	Range
<i>RowIndex</i>	Index of the character row.	Integer	0,1,2,3

Example

```
' Read a 10 character string
OCRFixedFont1.RowsToRead = 1
OCRFixedFont1.CharactersOnRow(0) = 10
OCRFixedFont1.Execute
```

CharacterSpacing

Description

Spacing, in pixels, between characters in the string to read. Calculated from the characters in the training string, after a successful **Learn**. Character spacing is the clear vertical zone between two characters. This property should never be modified manually.

Syntax

```
control.CharacterSpacing [= Value As Single]
```

Type

Single

Range

Minimum: 1

Maximum: Width of the training box

Example

```
Dim CharacterHeight As Integer
Dim CharacterWidth As Integer
Dim CharacterSpacing As Integer
Dim StrokeWidth As Integer

' Get the learned character dimensions

CharacterHeight = OCRFixedFont1.CharacterHeight
```

```

CharacterWidth = OCRFixedFont1.CharacterWidth
CharacterSpacing = OCRFixedFont1.CharacterSpacing
StrokeWidth = OCRFixedFont1.StrokeWidth

```

CharacterStrokeWidth

Description

Stroke width, in pixels, of the characters in the string to read. Calculated from the characters in the training string, after a successful **Learn**. The stroke width is the width of the line that forms a character. This property should never be modified manually.

Syntax

```
control.CharacterStrokeWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: 1

Maximum: Minimum of either character size or width

Example

```

Dim CharacterHeight As Integer
Dim CharacterWidth As Integer
Dim CharacterSpacing As Integer
Dim StrokeWidth As Integer

' Get the learned character dimensions

CharacterHeight = OCRFixedFont1.CharacterHeight
CharacterWidth = OCRFixedFont1.CharacterWidth
CharacterSpacing = OCRFixedFont1.CharacterSpacing
StrokeWidth = OCRFixedFont1.CharacterStrokeWidth

```

CharacterWidth

Description

Width of the characters, in pixels, in the string to read. Calculated from the characters in the training string after a successful **Learn**. This property should never be modified manually.

Syntax

```
control.CharacterWidth [= Value As Integer]
```

Type

Integer

Range

Minimum: Width of the primary character box minus a few pixels

Maximum: Width of the primary character box plus a few pixels

Example

```

Dim CharacterHeight As Integer
Dim CharacterWidth As Integer
Dim CharacterSpacing As Integer
Dim StrokeWidth As Integer

```

```
' Get the learned character dimensions
```

```
CharacterHeight = OCRFixedFont1.CharacterHeight
CharacterWidth = OCRFixedFont1.CharacterWidth
CharacterSpacing = OCRFixedFont1.CharacterSpacing
StrokeWidth = OCRFixedFont1.StrokeWidth
```

ConfidenceThreshold

Description

Minimum confidence value required by the tool to accept a character in the string. The string will be rejected if the confidence value of all the characters in the string fall below the required value. The confidence value reflects the match percentage between a decoded character and its ideal shape. A **ConfidenceThreshold** of 100 requires a perfect match of all the characters in the string. The default **ConfidenceThreshold** value is 40, which will tolerate poorly formed or damaged characters in the string.

Syntax

```
control.ConfidenceThreshold [= Value As Integer]
```

Type

Integer

Range

Minimum: 0
Maximum:100

Remark(s)

The string may be successfully read even if the confidence value of one or more characters (but not all) is less than the threshold value. In such cases the tool reports a 6001 status code, indicating an OCR confidence checking failure. See **StatusCode** for additional information on status codes.

Example

```
' Accept only a nearly perfect string
OCRFixedFont1.ContrastThreshold = 95
OCRFixedFont1.Execute
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

```
control.CoordinateSystem [= Value As hsCoordinateSystem]
```

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

hsCoordinateSystem	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```

Dim CurrentSystem As hsCoordinateSystem
' Get the current coordinate system
CurrentSystem = OcrFixedFont1.CoordinateSystem
' Express the results in the Object coordinate system
OcrFixedFont1.CoordinateSystem = hsCoordinateSystemObject

```

CustomFielding

Description

Assigns a fielding option to a given character position in a string. For each row, each character position in the string corresponds to one individual field. Each field can be set as either alphanumeric (AN), alpha characters only (AL), numeric only (NU), or any subset of alphanumeric characters. **CustomFielding** is enabled only after a successful **Learn**.

Syntax

control.**CustomFielding**(RowIndex As Integer, CharIndex As Integer) [= Value As String]

Type

String

Range

The valid range for this property is as follows:

Value	Description
"AN"	The field contains alphanumeric characters.
"AL"	The field contains alpha characters only.
"NU"	The field contains numeric characters only.
Single character string: Examples: "A" or "3"	The field contains a fixed character. The read result for this field always corresponds to the character specified and its confidence value is always 100. If the space character is specified, the Detect Space fine tune method should be disabled.

Value	Description
String containing multiple comma separated characters: Examples: "A,B,C" or "1,2,3"	The field contains a known set of characters. The read result for this field always corresponds to one of the characters in the comma separated list.

Parameter(s)

Name	Description	Type	Range
<i>RowIndex</i>	Index of the row.	Integer	0,1,2,3
<i>CharIndex</i>	Index of the character in the row.	Integer	Minimum: 0 Maximum: 29

Example

```

OCRFixedFont1.CharactersOnRow(0) = 10
' Character 0 is alpha
OcrFixedFont1.CustomFielding(0,0) = "AL"
' Character 1 is alpha
OcrFixedFont1.CustomFielding(0,1) = "AL"
' Character 2 is numeric
OcrFixedFont1.CustomFielding(0,2) = "NU"
' Character 3 is numeric
OcrFixedFont1.CustomFielding(0,3) = "NU"
' Character 4 is one of X, Y, or Z
OcrFixedFont1.CustomFielding(0,4) = "X,Y,Z"
' Character 5 is always -
OcrFixedFont1.CustomFielding(0,5) = "-"
' Character 6 to 9 can be anything
OcrFixedFont1.CustomFielding(0,6) = "AN"
OcrFixedFont1.CustomFielding(0,7) = "AN"
OcrFixedFont1.CustomFielding(0,8) = "AN"
OcrFixedFont1.CustomFielding(0,9) = "AN"
' Read the string
OcrFixedFont1.Execute

```

Database**Description**

Reserved for internal use.

DatabasePassword**Description**

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Info As String
Info = OCRFixedFont1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = OCRFixedFont1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & _
```

OCRFixedFont1.**ErrorObject**.Description

FineTuneMethodEnabled

Description

Property used to configure methods to fine-tune the performance of the OCR algorithm. This property is auto-configured after a successful **Learn**.

Syntax

```
control.FineTuneMethodEnabled(Method As hsOcrFineTuneMethod) [= Value as Boolean]
```

Type

Boolean

Range

Value	Description
True	The fine-tuning method is enabled
False	The fine-tuning method is disabled

Parameter(s)

Name	Description	Type	Range
<i>Method</i>	Fine-tuning method to enable/disable	hsOcrFineTuneMethod	Enum, See table below

hsOcrFineTuneMethod	Value	Description
hsOcrTuneDetectSpace	0	Enables reading of empty character spaces in the character string. Automatically enabled by a Learn when the training string contains at least one empty space.
hsOcrTuneNumberOfCharactersVary	1	Allows the number of characters in the string to be read to vary from the number of characters in the training string. This option is supported by Standard and Dot Matrix methods.
hsOcrTuneDisablePreFilter	2	Disables the tool's internal prefilter process to prevent removal of too much image detail.
hsOcrTuneCharacterSizeSpaceVary	3	Allows character size and spacing between characters to vary up to 10% from the training string values. Enables handling of nonuniform character size and spacing. This option is ignored when Degraded method is enabled.
hsOcrTuneSpeedEnhance	4	Applies an OCR algorithm that is faster but less robust.
hsOcrTuneRemoveNoise	5	Enables reading of characters in noisy images. This option is automatically enabled when the Degraded method is enabled.
hsOcrTuneReserved1	6	Reserved for future use.
hsOcrTuneReserved2	7	Reserved for future use.
hsOcrTuneSemiMChecksum	8	When enabled, the tool verifies that the read text string passes the SEMI Checksum test,
hsOcrTuneReserved3	9	Reserved for future use.

Example

```
' Optimize the speed for reading a string in a clean image
```

```

OCRFixedFont1.OcrMethod = hsOcrMethodStandard
OCRFixedFont1.FineTuneMethodEnabled _
    (hsOcrTuneSpeedEnhance) = True
OCRFixedFont1.FineTuneMethodEnabled _
    (hsOcrTuneRemoveNoise) = False
OCRFixedFont1.Execute

```

FontType

Description

Specifies the type of font used by the tool to decode the characters in the string to read. This property is auto-configured after a successful **Learn**, it is set to the value specified by the **LearnFontType** property.

Syntax

control.**FontType** [= *Value* As hsOcrFontType]

Type

hsOcrFontType

Range

The valid range for this property is as follows:

hsOcrFontType	Value	Description
hsOcrFontTypeSemi	0	Reads SEMI fonts.
hsOcrFontTypeOcrA	1	Reads OCR-A fonts.
hsOcrFontTypeOcrB	2	Reads OCR-B fonts.

Example

```

' Read a SEMI font string
OcrFixedFont1.FontType = hsOcrFontTypeSemi
OCRFixedFont1.Execute

```

InputImage

Description

Name of the image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```

Dim ImageName As String
ImageName = OcrFixedFont1.InputImage
OcrFixedFont2.InputImage = "image1"

```


InputGreyScaleImage

Description

Name of the image on which the tool is applied.

Syntax

control.InputImage [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = OcrFixedFont1.InputImage
OcrFixedFont2.InputImage = "image1"
```

InputView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = OcrFixedFont1.InputView
OcrFixedFont2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = OcrFixedFont1.InputView
OcrFixedFont2.InputView = "view0"
```

InputInstanceScene

Description

Name of the scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.InputInstanceScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = OcrFixedFont1.InputInstanceScene
OcrFixedFont2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

control.InputInstanceView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = OcrFixedFont1.InputInstanceView
OcrFixedFont2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.InputModelScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = OcrFixedFont1.InputModelScene
OcrFixedFont2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details

Syntax

```
control.InputModelView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = OcrFixedFont1.InputModelView
OcrFixedFont2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = OcrFixedFont1.InstanceSelectionEnabled
OcrFixedFont2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

`control.InstanceSelectionIndex [= Value As Integer]`

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = OcrFixedFont1.InstanceSelectionIndex
OcrFixedFont2.InstanceSelectionIndex = 0
```

LearnElapsedTime

Description

The execution time (in milliseconds) of the Learn process. Read only.

Syntax

`control.LearnElapsedTime`

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = OcrFixedFont1.LearnElapsedTime
```

LearnFontType

Description

Specifies the type of font used by the tool to decode the characters in the training string. After a successful **Learn**, this property is copied to the **FontType** property.

Syntax

`control.LearnFontType [= Value As hsOcrFontType]`

Type

hsOcrFontType

Range

The valid range for this property is as follows:

hsOcrFontType	Value	Description
hsOcrFontTypeSemi	0	Reads SEMI fonts.
hsOcrFontTypeOcrA	1	Reads OCR-A fonts.
hsOcrFontTypeOcrB	2	Reads OCR-B fonts.

Example

```
' Train a SEMI font string
OcrFixedFont1.LearnFontType = hsOcrFontTypeSemi
OcrFixedFont1.Learn
```

LearnOcrMethod

Description

Specifies the OCR method used by the tool to decode the characters in the training string. After a successful **Learn**, this property is copied to the **OcrMethod** property.

Syntax

control.LearnOcrMethod [= *Value* as hsOcrMethod]

Type

hsOcrMethod

Range

The valid range for this property is as follows:

hsOcrMethod	Value	Description
hsOcrFontTypeStandard	0	Method works on clean characters with no significant background noise. Allows the text string to rotate up to 15 degrees from the horizontal or vertical orientation. The actual number of characters allowed to vary from the number specified by CharactersOnRow. Speed of this method is relatively fast.
hsOcrFontTypeDotMatrix	1	Method optimized for Dot Matrix characters. Similar to performance of the Standard method except that it is more tolerant of background noise.
hsOcrFontTypeDegraded	2	Default value. Most robust of all three methods Capable of reading characters with damage and noise. Allows minor rotation from the horizontal or vertical orientation. Reading is slower than the above two methods.

Example

```
' Train the string in a degraded image
OcrFixedFont1.LearnOcrMethod = hsOcrFontTypeDegraded
OcrFixedFont1.Learn
```

LearnPrimaryCharacter

Description

A single character from the training string selected as the primary character for the Learn process. The primary character should be chosen such that the size and the stroke width of the character reflects the average size of the characters in the string.

Syntax

control.**LearnPrimaryCharacter** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
' Use the character 5 as the primary character for the
' Learn process
OcrFixedFont1.LearnPrimaryCharacter = "5"
OcrFixedFont1.Learn
```

LearnPrimaryCharacterBoxBottom

Description

The bottommost coordinate of the bounding box that encloses the primary character of the training string, expressed in pixels with respect to the Image coordinate system.

Syntax

control.**LearnPrimaryCharacterBoxBottom** [= *Value* As Single]

Type

Single

Range

Any Y coordinate inside the training box.

Example

```
' Set the bounding box around the primary character
OcrFixedFont1.LearnPrimaryCharacterBoxBottom = 150
OcrFixedFont1.LearnPrimaryCharacterBoxTop = 190
OcrFixedFont1.LearnPrimaryCharacterBoxLeft = 210
OcrFixedFont1.LearnPrimaryCharacterBoxRight = 235
OcrFixedFont1.Learn
```

LearnPrimaryCharacterBoxLeft

Description

The leftmost coordinate of the bounding box that encloses the primary character of the training string, expressed in pixels with respect to the Image coordinate system.

Syntax

control.**LearnPrimaryCharacterBoxLeft** [= *Value* As Single]

Type

Single

Range

Any X coordinate inside the training box.

Example

```
' Set the bounding box around the primary character
OcrFixedFont1.LearnPrimaryCharacterBoxBottom = 150
OcrFixedFont1.LearnPrimaryCharacterBoxTop = 190
```

```
OcrFixedFont1.LearnPrimaryCharacterBoxLeft = 210
OcrFixedFont1.LearnPrimaryCharacterBoxRight = 235
OcrFixedFont1.Learn
```

LearnPrimaryCharacterBoxRight

Description

The rightmost coordinate of the bounding box that encloses the primary character of the training string, expressed in pixels with respect to the Image coordinate system.

Syntax

```
control.LearnPrimaryCharacterBoxRight [= Value As Single]
```

Type

Single

Range

Any X coordinate inside the training box.

Example

```
' Set the bounding box around the primary character
OcrFixedFont1.LearnPrimaryCharacterBoxBottom = 150
OcrFixedFont1.LearnPrimaryCharacterBoxTop = 190
OcrFixedFont1.LearnPrimaryCharacterBoxLeft = 210
OcrFixedFont1.LearnPrimaryCharacterBoxRight = 235
OcrFixedFont1.Learn
```

LearnPrimaryCharacterBoxTop

Description

The topmost coordinate of the bounding box that encloses the primary character of the training string, expressed in pixels with respect to the Image coordinate system.

Syntax

```
control.LearnPrimaryCharacterBoxTop [= Value As Single]
```

Type

Single

Range

Any Y coordinate inside the training box.

Example

```
' Set the bounding box around the primary character
OcrFixedFont1.LearnPrimaryCharacterBoxBottom = 150
OcrFixedFont1.LearnPrimaryCharacterBoxTop = 190
OcrFixedFont1.LearnPrimaryCharacterBoxLeft = 210
OcrFixedFont1.LearnPrimaryCharacterBoxRight = 235
OcrFixedFont1.Learn
```

LearnStatusCode

Description

Status code which reports on the success or failure of the Learn process. Read only.

Syntax`control.LearnStatusCode`**Type**

Long

Range

The valid range for this property is as follows:

Value	Description
0000	Success
4601	No Edge Candidate Found
4604	First Edge Not Found or Too Small
4605	Second Edge Not Found
4611	Third Edge Not Found
4612	Fourth Edge Not Found
4620	Symbology Corner Out of Image
4621	Size Test Failed
4622	Unable to Find Row/Column
4633	Border Match Failed
3048	Symbology Decode Failed
4630	Inspection Timed Out
4301	Trigger Overrun
4303	String Mismatch
4304	Angle Match Failed
4305	Excessive Damage
4310	Process Overrun
6001	OCR Conformance Failure
6002	OCR General Error
9999	Dongle Missing

Example

```

Dim ErrorCode as Long
Dim ErrorString as String
' Get the status code and string if the learn fails
If Not OcrFixedFont1.Learn Then
    ErrorCode = OcrFixedFont1.LearnStatusCode
    ErrorString = OcrFixedFont1.LearnStatusCodeString
EndIf

```

LearnStatusCodeString

Description

Status code message associated to the status code returned by the **LearnStatusCode** property. Read only.

Syntax*control*.**LearnStatusCodeString****Type**

String

Range

The valid range for this property is as follows:

Value	Description
0000	Success
1000	No Execute Performed
4601	No Edge Candidate Found
4604	First Edge Not Found or Too Small
4605	Second Edge Not Found
4611	Third Edge Not Found
4612	Fourth Edge Not Found
4620	Symbology Corner Out of Image
4621	Size Test Failed
4622	Unable to Find Row/Column
4633	Border Match Failed
3048	Symbology Decode Failed
4630	Inspection Timed Out
4301	Trigger Overrun
4303	String Mismatch
4304	Angle Match Failed
4305	Excessive Damage
4310	Process Overrun
6001	OCR Conformance Failure
6002	OCR General Error
9999	Dongle Missing

Example

```

Dim ErrorCode as Long
Dim ErrorString as String
' Get the status code and string if the learn fails
If Not OcrFixedFont1.Learn Then
    ErrorCode = OcrFixedFont1.LearnStatusCode
    ErrorString = OcrFixedFont1.LearnStatusCodeString
EndIf

```

LearnTextOrientation

Description

Specifies the orientation of the training string. After a successful **Learn**, this property is copied to the **TextOrientation** property.

Syntax

control.**LearnTextOrientation** [= *Value* As hsOcrTextOrientation]

Type

hsOcrTextOrientation

Range

The valid range for this property is as follows:

hsOcrTextOrientation	Value	Description
hsOcrTextOrientationNormal	0	Normal text orientation: horizontal, read left to right
hsOcrTextOrientationTopToBottom	1	String rotated 90 degrees clockwise with respect to normal position, read top to bottom.
hsOcrTextOrientationUpsideDown	2	String flipped vertically with respect to normal position.
hsOcrTextOrientationBottomToTop	3	String rotated 90 degrees counterclockwise with respect to normal position, read bottom to top.
hsOcrTextOrientationMirrorNormal	10	String flipped horizontally, with respect to normal position.
hsOcrTextOrientationMirrorTopToBottom	11	String flipped horizontally and vertically, with respect to normal position.
hsOcrTextOrientationMirrorUpsideDown	12	String rotated 90 degrees clockwise with respect to normal position, read top to bottom.
hsOcrTextOrientationMirrorBottomToTop	13	String rotated flipped horizontally and 90 degrees counterclockwise with respect to normal position.

Example

```
' Train on an upside down string
OcrFixedFont1.LearnTextOrientation = _
    hsOcrTextOrientationUpsideDown
OcrFixedFont1.Learn
```

LearnTextPolarity

Description

Specifies the polarity of the characters in the training string with respect to their background. After a successful **Learn**, this property is copied to the **TextPolarity** property.

Syntax

control.**LearnTextPolarity** [= *Value* As hsOcrTextPolarity]

Type

hsOcrTextPolarity

Range

The valid range for this property is as follows:

hsOcrTextPolarity	Value	Description
hsOcrTextPolarityLightOnDark	0	Reads light text on dark background.
hsOcrTextPolarityDarkOnLight	1	Reads dark text on light background.
hsOcrTextPolarityUnknown	2	Reads either dark text on light background or light text on dark background

Example

```
' Train on a string of light characters on a dark
' background
OCRFixedFont1.LearnTextPolarity = hsTextPolarityLightOnDark
OCRFixedFont1.Learn
```

LearnTrainingBoxBottom

Description

The bottommost coordinate of the bounding box that encloses the training string, expressed in pixels.

Syntax

control.**LearnTrainingBoxBottom** [= *Value* As Single]

Type

Single

Range

Any Y coordinate inside the image.

Example

```
' Set the bounding box around the training string
OcrFixedFont1.LearnTrainingBoxBottom = 145
OcrFixedFont1.LearnTrainingBoxTop = 195
OcrFixedFont1.LearnTrainingBoxLeft = 180
OcrFixedFont1.LearnTrainingBoxRight = 365
OcrFixedFont1.Learn
```

LearnTrainingBoxLeft

Description

The leftmost coordinate of the bounding box that encloses the training string, expressed in pixels.

Syntax

control.**LearnTrainingBoxLeft** [= *Value* As Single]

Type

Single

Range

Any X coordinate inside the image.

Example

```
' Set the bounding box around the training string
OcrFixedFont1.LearnTrainingBoxBottom = 145
```

```
OcrFixedFont1.LearnTrainingBoxTop = 195
OcrFixedFont1.LearnTrainingBoxLeft = 180
OcrFixedFont1.LearnTrainingBoxRight = 365
OcrFixedFont1.Learn
```

LearnTrainingBoxRight

Description

The rightmost coordinate of the bounding box that encloses the training string, expressed in pixels.

Syntax

control.**LearnTrainingBoxRight** [= *Value* As Single]

Type

Single

Range

Any X coordinate inside the image.

Example

```
' Set the bounding box around the training string
OcrFixedFont1.LearnTrainingBoxBottom = 145
OcrFixedFont1.LearnTrainingBoxTop = 195
OcrFixedFont1.LearnTrainingBoxLeft = 180
OcrFixedFont1.LearnTrainingBoxRight = 365
OcrFixedFont1.Learn
```

LearnTrainingBoxTop

Description

The topmost coordinate of the bounding box that encloses the training string, expressed in pixels.

Syntax

control.**LearnTrainingBoxTop** [= *Value* As Single]

Type

Single

Range

Any Y coordinate inside the image.

Example

```
' Set the bounding box around the training string
OcrFixedFont1.LearnTrainingBoxBottom = 145
OcrFixedFont1.LearnTrainingBoxTop = 195
OcrFixedFont1.LearnTrainingBoxLeft = 180
OcrFixedFont1.LearnTrainingBoxRight = 365
OcrFixedFont1.Learn
```

LearnTrainingString

Description

Specifies the sequence of characters contained in the training string.

Syntax

control.LearnTrainingString [= *Value* As String]

Type

String

Range

Not applicable

Example

```
' The training string contains the following sequence of
' characters: T12-ZQ45987
OcrFixedFont1.LearnTrainingString = "T12-ZQ45987"
OcrFixedFont1.Learn
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.ModelTransformEnabled [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled as Boolean
Enabled = OCRFixedFont1.ModelTransformEnabled
OCRFixedFont2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.Name [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
```

```
ProcName = OCRFixedFont1.Name
OCRFixedFont2.Name = "EdgeLocA"
```

OcrMethod

Description

Specifies the OCR method used by the tool to decode the characters in the string to read. This property is auto-configured after a successful **Learn**, it is set to the value specified by the **LearnOcrMethod** property.

Syntax

```
control.OcrMethod [= Value as hsOcrMethod]
```

Type

hsOcrMethod

Range

The valid range for this property is as follows:

hsOcrMethod	Value	Description
hsOcrFontTypeStandard	0	Method works on clean characters with no significant background noise. Allows the text string to rotate up to 15 degrees from the horizontal or vertical orientation. The actual number of characters allowed to vary from the number specified by CharactersOnRowi. Speed of this method is relatively fast.
hsOcrFontTypeDotMatrix	1	Method optimized for Dot Matrix characters. Similar to performance of the Standard method except that it is more tolerant of background noise.
hsOcrFontTypeDegraded	2	Default value. Most robust of all three methods Capable of reading characters with damage and noise. Allows minor rotation from the horizontal or vertical orientation. Reading is slower than the above two methods.

Example

```
' Read a string in a degraded image
OcrFixedFont1.OcrMethod = hsOcrFontTypeDegraded
OcrFixedFont1.Execute
```

ProcessingAlgorithm

Description

Activates one of the optional enhancement preprocessing algorithms that can be applied to the input image prior to the decoding process. The selected preprocessing algorithm is also applied prior to a **Learn**.

Syntax

```
control.ProcessingAlgorithm [= Value As hsOcrProcessing]
```

Type

hsOcrProcessing

Range

The valid range for this property is as follows:

hsOcrProcessing	Value	Description
hsOcrProcessingNone	0	No pre-processing is applied
hsOcrProcessingInvert	1	Reverses the polarity of the image
hsOcrProcessingErode	2	Erodes light objects
hsOcrProcessingDilate	3	Dilates light objects
hsOcrProcessingOpen	4	Removes small light defects
hsOcrProcessingClose	5	Removes small dark defects
hsOcrProcessingGaussian	6	Smooths the edges in the image
hsOcrProcessingSharpen	7	Enhances the edges in the image
hsOcrProcessingSharpenLow	8	Enhances the edges in the image after a low-pass filter is applied
hsOcrProcessingMedian	9	Removes noise with a median filter
hsOcrProcessingHistogramStretching	10	Stretches the histogram to increase contrast

Example

```

' Filter out high frequency noise in the image prior to
' reading the code
OcrFixedFont1.ProcessingAlgorithm = _
    hsOcrProcessingMedian
OcrFixedFont1.ProcessingSize = 5
OcrFixedFont1.Execute

```

ProcessingSize

Description

Sets the **Size** parameter that is used by the preprocessing algorithm selected by the **ProcessingAlgorithm** property.

Syntax

control.**ProcessingSize** [= *Value* As Integer]

Type

Integer

Range

The range depends on the value of **ProcessingSize**:

hsBarcodeProcessing	Range	Description
hsBarcodeProcessingNone	Not applicable.	Not applicable.
hsBarcodeProcessingInvert	Not applicable.	Not applicable.
hsBarcodeProcessingErode	Greater than 0	Size of neighborhood
hsBarcodeProcessingDilate	Greater than 0	Size of neighborhood
hsBarcodeProcessingOpen	Greater than 0	Size of neighborhood
hsBarcodeProcessingClose	Greater than 0	Size of neighborhood
hsBarcodeProcessingGaussian	3,5,7	Kernel size

hsBarcodeProcessing	Range	Description
hsBarcodeProcessingSharpen	3,5,7	Kernel size
hsBarcodeProcessingSharpenLow	3,5,7	Kernel size
hsBarcodeProcessingMedian	3,5,7	Kernel size
hsBarcodeProcessingHistogramStretching	Not applicable.	Not applicable.

Example

```
' Filter out high frequency noise in the image prior to
' reading the code
OcrFixedFont1.ProcessingAlgorithm = _
    hsOcrProcessingMedian
OcrFixedFont1.ProcessingSize = 5
OcrFixedFont1.Execute
```

RowsToRead

Description

Number of rows of character to read.

Syntax

control.**RowsToRead** [= *Value* As Integer]

Type

Not applicable.

Range

Minimum: 1

Maximum: 4

Example

```
' Read a single row of characters
OCRFixedFont1.RowsToRead = 1
OcrFixedFont1.Execute
```

SamplingStep

Description

Step used in the tool's last execution to sample the region of interest from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = OCRFixedFont1.SamplingStep
```

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = OCRFixedFont1.SamplingStepCustom
OCRFixedFont2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Example

```
Dim Enabled as Boolean
Enabled = OCRFixedFont1.SamplingStepCustomEnabled
OCRFixedFont2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

control.SamplingStepDefault

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Example

Dim Sampling As Single

Sampling = OCRFixedFont1.**SamplingStepDefault**

StatusCode

Description

Status code which reports on the success or failure of the reading process. Read only.

Syntax

control.StatusCode

Type

Long

Range

The valid range for this property is as follows:

Value	Description
0000	Success
1000	No Execute Performed
4601	No Edge Candidate Found
4604	First Edge Not Found or Too Small
4605	Second Edge Not Found
4611	Third Edge Not Found
4612	Fourth Edge Not Found
4620	Symbology Corner Out of Image
4621	Size Test Failed
4622	Unable to Find Row/Column
4633	Border Match Failed
3048	Symbology Decode Failed
4630	Inspection Timed Out
4301	Trigger Overrun
4303	String Mismatch

Value	Description
4304	Angle Match Failed
4305	Excessive Damage
4310	Process Overrun
6001	OCR Conformance Failure
6002	OCR General Error
9999	Dongle Missing

Example

```

Dim Result as String
Dim Confidence as Integer
Dim ErrorCode as Long
Dim ErrorString as String
' Read the string
OcrFixedFont1.Execute
Result = OcrFixedFont1.String(0)
' Get the confidence value, status code and status string
Confidence = OcrFixedFont1.StringConfidence(0)
ErrorCode = OcrFixedFont1.StatusCode
ErrorString = OcrFixedFont1.StatusCodeString

```

StatusCodeString

Description

Status code message associated to the status code returned by the **StatusCode** property.
Read only.

Syntax

control.**StatusCodeString**

Type

String

Range

Not applicable.

Example

```

Dim Result as String
Dim Confidence as Integer
Dim ErrorCode as Long
Dim ErrorString as String
' Read the string
OcrFixedFont1.Execute
Result = OcrFixedFont1.String(0)
' Get the confidence value, status code and status string
Confidence = OcrFixedFont1.StringConfidence(0)
ErrorCode = OcrFixedFont1.StatusCode

```

ErrorString = OcrFixedFont1.**StatusCodeString**

String

Description

Character string read in the specified row. Read only.

Syntax

control.**String**(RowIndex As Integer)

Type

String

Range

Not applicable.

Parameter(s)

<i>Index Parameter</i>	Description	Type	Range
<i>RowIndex</i>	Index of the row.	Integer	0,1,2,3

Example

```
Dim Result as String
Dim Confidence as Integer
Dim ErrorCode as Long
Dim ErrorString as String
' Read the string
OcrFixedFont1.Execute
Result = OcrFixedFont1.String(0)
' Get the confidence value, status code and status string
Confidence = OcrFixedFont1.StringConfidence(0)
ErrorCode = OcrFixedFont1.StatusCode
ErrorString = OcrFixedFont1.StatusCodeString
```

StringConfidence

Description

The confidence value of the character string read in the specified row. The confidence value of a string is defined as the minimum of the individual confidence values of all the characters read in the string. See the **CharacterConfidence** property for more details on character confidence values. A string may be successfully read even if its confidence value is less than the threshold value set by the **ConfidenceThreshold** property because the confidence value of all the characters in the string must be lower than the threshold for the string to be rejected.

Syntax

control.**StringConfidence**(RowIndex As Integer) [= Value As Integer]

Type

Integer

Range

Minimum: 0
Maximum:100

Parameter(s)

<i>Index Parameter</i>	Description	Type	Range
<i>RowIndex</i>	Index of the row.	Integer	0,1,2,3

Example

```

Dim Result as String
Dim Confidence as Integer
Dim ErrorCode as Long
Dim ErrorString as String
' Read the string
OcrFixedFont1.Execute
Result = OcrFixedFont1.String(0)
' Get the confidence value, status code and status string
Confidence = OcrFixedFont1.StringConfidence(0)
ErrorCode = OcrFixedFont1.StatusCode
ErrorString = OcrFixedFont1.StatusCodeString

```

StringLengthVariation

Description

The percentage of allowable variation (+ or -) in the length of an actual string from that of the training string. **StringLengthVariation** is always set back to 0 by the **Learn** process. This parameter is only applicable to the Degraded **OcrMethod**, it is ignored when the Standard or the Dot Matrix methods are used.

Syntax

control.**StringLengthVariation** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 10

Example

```

' Read a string in a degraded image
OcrFixedFont1.OcrMethod = hsOcrFontTypeDegraded
OcrFixedFont1.StringLengthVariation = 5
OcrFixedFont1.Execute

```

TextOrientation

Description

Specifies the orientation of the string to read. This property is auto-configured after a successful **Learn**, it is set to the value specified by the **LearnTextOrientation** property.

Syntax

control.**TextOrientation** [= *Value* As hsOcrTextOrientation]

Type

hsOcrTextOrientation

Range

The valid range for this property is as follows:

hsOcrTextOrientation	Value	Description
hsOcrTextOrientationNormal	0	Normal text orientation: horizontal, read left to right
hsOcrTextOrientationTopToBottom	1	String rotated 90 degrees clockwise with respect to normal position, read top to bottom.
hsOcrTextOrientationUpsideDown	2	String flipped vertically with respect to normal position.
hsOcrTextOrientationBottomToTop	3	String rotated 90 degrees counterclockwise with respect to normal position, read bottom to top.
hsOcrTextOrientationMirrorNormal	10	String flipped horizontally, with respect to normal position.
hsOcrTextOrientationMirrorTopToBottom	11	String flipped horizontally and vertically, with respect to normal position.
hsOcrTextOrientationMirrorUpsideDown	12	String rotated 90 degrees clockwise with respect to normal position, read top to bottom.
hsOcrTextOrientationMirrorBottomToTop	13	String rotated flipped horizontally and 90 degrees counterclockwise with respect to normal position.

Example

```
' Read an upside down string
OcrFixedFont1.TextOrientation = _
    hsOcrTextOrientationUpsideDown
OcrFixedFont1.Execute
```

TextPolarity

Description

Specifies the polarity of the characters in the string to read with respect to their background. This property is auto-configured after a successful **Learn**, it is set to the value specified by the **LearnTextPolarity** property.

Syntax

control.**TextPolarity** [= Value As hsTextPolarity]

Type

hsTextPolarity

Range

The valid range for this property is as follows:

hsBarcodePolarity	Value	Description
hsTextPolarityLightOnDark	0	Light characters on a dark background
hsTextPolarityDarkOnLight	1	Dark characters on a light background
hsTextPolarityUnknown	2	Unknown polarity. Supported only by the Degraded OcrMethod .

Example

```
' Read a string of light characters on a dark background
OCRFixedFont1.TextPolarity = hsTextPolarityLightOnDark
```

OCRFixedFont1.Execute

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.**ToolHeight** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = OCRFixedFont1.ToolHeight
OCRFixedFont2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionX** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = OCRFixedFont1.ToolPositionX
OCRFixedFont2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = OCRFixedFont1.ToolPositionY
OCRFixedFont2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = OCRFixedFont1.ToolRotation
OCRFixedFont2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

```
control.ToolWidth [= Value As Single]
```

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = OCRFixedFont1.ToolWidth
OCRFixedFont2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

```
control.TransformRotation [= Value As Single]
```

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = OCRFixedFont1.TransformRotation
OCRFixedFont2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = OCRFixedFont1.TransformScale
OCRFixedFont2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Example

```
Dim X_Trans As Single
X_Trans = OCRFixedFont1.TransformTranslationX
OCRFixedFont2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = OCRFixedFont1.TransformTranslationY
OCRFixedFont2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSOCRFixedFontTool".

Example

```
Dim ControlType As String
ControlType = OCRFixedFont1.TypeName
```

UnitsLength

Description

Unit used to express lengths and coordinates.

Syntax

control.UnitsLength [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

hsUnitsLength	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = OCRFixedFont1.UnitsLength
OCRFixedFont2.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.UnitsOrientation [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

hsUnitsOrientation	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = OCRFixedFont1.UnitsOrientation
OCRFixedFont2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.2.000 R 3217 (release)".

Example

```
Dim Version As String
Version = OCRFixedFont1.Version
```

Methods

AboutBox

Description

Displays the tool About box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Range

No return value.

Example

```
OCRFixedFont1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

```
OCRFixedFont1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideLearnInterface

Description

Hides and terminates the Learn dialog.

Syntax

```
control.HideLearnInterface()
```

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
OcrFixeFont1.HideLearnInterface
```

HideProperties

Description

Hides the property pages of the tool if they are displayed.

Syntax

```
control.HideProperties()
```

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = OCRFixedFont1.HideProperties
```

Learn

Description

Uses a training string to optimize the following character dimension parameters:

CharacterHeight, **CharacterWidth**, **CharacterSpacing** and **CharacterStrokeWidth**. The following training properties must be set before calling **Learn**: **LearnPrimaryCharacter**, **LearnFontType**, **LearnOcrMethod**, **LearnTextPolarity** and **LearnTextOrientation**. Before calling **Learn**, bounding boxes must also be defined for the training string and for the primary character using the following properties: **LearnTrainingBoxBottom**, **LearnTrainingBoxLeft**, **LearnTrainingBoxRight**, **LearnTrainingBoxTop**, **LearnPrimaryCharacterBoxBottom**, **LearnPrimaryCharacterBoxLeft**, **LearnPrimaryCharacterBoxRight**, and **LearnPrimaryCharacterBoxTop**. After a successful call to **Learn**, the following parameters are automatically configured: **FineTuneMethodEnabled**, **FontType**, **OcrMethod**, **TextOrientation**, **TextPolarity**, **CharactersOnRow**, and **StringLengthVariation**.

Syntax

control.**Learn**

Parameter(s)

None

Return

Boolean

Example

```
Dim ErrorCode as Long
Dim ErrorString as String
' Get the status code and string if the learn fails
If Not OcrFixedFont1.Learn Then
    ErrorCode = OcrFixedFont1.LearnStatusCode
    ErrorString = OcrFixedFont1.LearnStatusCodeString
EndIf
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowLearnInterface

Description

Shows the Learn dialog. After completion of the learn process, the status can be retrieved with the **LearnStatusCode** property.

Syntax

control.**ShowLearnInterface**(*Modal* As Boolean)

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Dim ErrorCode As Long
Dim ErrorString as String
OCRFixedFont1.ShowLearnInterface(True)
ErrorCode = OcrFixedFont1.LearnStatusCode
ErrorString = OcrFixedFont1.LearnStatusCodeString
```

ShowProperties

Description

Shows the property pages of the tool.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button.

Option parameter: bitfield elements		
Element	Value	Description
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success as Boolean
Success = OCRFixedFont1.ShowProperties(True, _
    hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Image Histogram

Description

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

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
```

```
Result = Histogram1.BilinearInterpolationEnabled
```

```
Histogram2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = Histogram1.CalibratedUnitsEnabled
Histogram2.CalibratedUnitsEnabled = False
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

*control.***Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Histogram1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

*control.***ElapsedTime**

Type

Single

Range

Greater than 0.0

Remark(s)

This property gives the total elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Histogram1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Histogram1.ErrorObject.Description
```

GreylevelRange

Description

Range of greylevel values of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. The range is equal to MaximumGreylevelValue - MinimumGreylevelValue + 1. Read only.

Syntax

control.**GreylevelRange**

Type

Integer

Range

Minimum: 0

Maximum: 256

Example

```
Dim GreylevelRange As Integer
```

```
GreylevelRange = Histogram1.GreylevelRange
```

Histogram

Description

Histogram of greylevel values of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. The histogram comprises 256 bins. To each of the 256 possible greylevel values is associated one histogram bin. It contains the number of pixels with the corresponding greylevel value in the region of interest. Read only.

Syntax

control.**Histogram**(*Index As Integer*)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Name	Description	Type	Range
Index	Index of the histogram bin.	Integer	Minimum: 0 Maximum: 255

Example

```
Dim i As Integer
Dim Total As Long
' Compute the total number of pixels in the histogram
' (equivalent to the HistogramPixelCount property)
Total = 0
For i = 0 To 255
    Total = Total + Histogram1.Histogram(i)
Next i
```

HistogramPixelCount

Description

Total number of pixels in the histogram. Equal to the sum of all of the histogram's bins. Read only.

Syntax

control.HistogramPixelCount

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim Count As Long
Count = Histogram1.HistogramPixelCount
```

ImageHeight

Description

Height of the tool's region of interest expressed in pixels. Read only.

Syntax

control.ImageHeight

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim Height As Long
Dim Width As Long
Dim Count As Long
```

```

Height = Histogram1.ImageHeight
Width = Histogram1.ImageWidth
Count = Height * Width

```

ImagePixelCount

Description

Number of pixels in the tool's region of interest. Read only.

Syntax

```
control.ImagePixelCount
```

Type

Long

Range

Greater than or equal to 0.

Example

```

Dim Count As Long
Count = Histogram1.ImagePixelCount

```

ImageSubsampling

Description

Factor used to subsample the image in the tool's region of interest. With a subsampling factor of 1, the image is not subsampled. With a subsampling factor of 2, the image is subsampled in tiles of 2x2 pixels. With a subsampling factor of 3 the image is subsampled in tiles of 3x3 pixels and so forth.

Syntax

```
control.ImageSubsampling [= Value As Integer]
```

Type

Integer

Range

[1,2,3,4,5,6,7,8]

Remark(s)

Using a higher subsampling factor speeds up the generation of the histogram but slightly reduces the accuracy of the statistics computed from the histogram. The pixel properties computed by the Image Histogram tool are normalized with respect to the subsampling factor (**HistogramPixelCount**, **ImageHeight**, **ImagePixelCount** and **ImageWidth**). So for instance, the total number of pixels in the histogram should remain the same at any subsampling factor. Note that there might be slight differences in the values of these properties when either of the width or the height of the region of interest is not a multiple of the subsampling factor used.

Example

```

Dim Subsampling As Integer
Subsampling = Histogram1.ImageSubsampling
Histogram2.ImageSubsampling = 8

```

ImageWidth

Description

Width of the tool's region of interest expressed in pixels. Read only.

Syntax

control.**ImageWidth**

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim Height As Long
Dim Width As Long
Dim Count As Long
Height = Histogram1.ImageHeight
Width = Histogram1.ImageWidth
Count = Height * Width
```

InputImage

Description

Name of the image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Histogram1.InputImage
Histogram2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Histogram1.InputImage
Histogram2.InputImage = "image1"
```


InputView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Histogram1.InputView
Histogram2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Histogram1.InputView
Histogram2.InputView = "view0"
```

InputInstanceScene

Description

Name of the scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.InputInstanceScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = Histogram1.InputInstanceScene
```

```
Histogram2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

```
control.InputInstanceView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Histogram1.InputInstanceView
Histogram2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

```
control.InputModelScene [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = Histogram1.InputModelScene
Histogram2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details.

Syntax

```
control.InputModelView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
```

```
ModelViewName = Histogram1.InputModelView
Histogram2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

*control.***InstanceSelectionEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = Histogram1.InstanceSelectionEnabled
Histogram2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

*control.***InstanceSelectionIndex** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = Histogram1.InstanceSelectionIndex
Histogram2.InstanceSelectionIndex = 0
```

MaximumGreylevelValue

Description

Highest greylevel value of all pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Read only.

Syntax

control.**MaximumGreylevelValue**

Type

Integer

Range

Minimum: 0

Maximum: 255

Example

Dim MaxGreylevel As Integer

MaxGreylevel = Histogram1.**MaximumGreylevelValue**

Mean

Description

Mean of the greylevel distribution of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Read only.

Syntax

control.**Mean**

Type

Single

Range

Minimum: 0

Maximum: 255

Example

Dim Mean As Single

Mean = Histogram1.**Mean**

Median

Description

Median of the greylevel distribution of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Read only.

Syntax

control.**Median**

Type

Single

Range

Minimum: 0

Maximum: 255

Example

```
Dim Median As Single
Median = Histogram1.Median
```

MinimumGreylevelValue

Description

Lowest greylevel value of all pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Read only.

Syntax

control.**MinimumGreylevelValue**

Type

Integer

Range

Minimum: 0

Maximum: 255

Example

```
Dim MinGreylevelValue As Integer
MinGreylevelValue = Histogram1.MinimumGreylevelValue
```

Mode

Description

Mode of the greylevel distribution of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. The mode is the greylevel value which corresponds to the histogram bin with the highest number of pixels. Read only.

Syntax

control.**Mode**

Type

Integer

Range

Minimum: 0

Maximum: 255

Example

```
Dim HistogramMode As Integer
HistogramMode = Histogram1.Mode
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim ModelBased As Boolean
ModelBased = Histogram1.ModelTransformEnabled
Histogram2.ModelTransformEnabled = True
```

ModePixelCount

Description

Number of pixels in the histogram bin which corresponds to the mode of the greylevel distribution of all pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. The mode is the greylevel value which corresponds to the histogram bin with the highest number of pixels. Read only.

Syntax

control.**ModePixelCount**

Type

Long

Range

Greater than or equal to 0.

Example

```
Dim PixelCount As Long
PixelCount = Histogram1.ModePixelCount
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = Histogram1.Name
Histogram2.Name = "HistoA"
```

SamplingStep

Description

Step used in the tool's last execution to sample the region of interest from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

Dim Step As Single

Step = Histogram1.**SamplingStep**

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax

control.**SamplingStepCustom** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

Dim Step As Single

Step = Histogram1.**SamplingStepCustom**

Histogram2.**SamplingStepCustom** = 2.1

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax

control.**SamplingStepCustomEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default optimal sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Example

```
Dim Enabled As Boolean
Enabled = Histogram1.SamplingStepCustomEnabled
Histogram2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

control.**SamplingStepDefault**

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = Histogram1.SamplingStepDefault
```

StandardDeviation

Description

Standard deviation of the greylevel distribution of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Read only.

Syntax

control.**StandardDeviation**

Type

Single

Range

Minimum: 0
Maximum: 255

Example

```
Dim StdDev As Single
```


StdDev = Histogram1.**StandardDeviation**

TailBlack

Description

Amount of pixels to ignore at the dark end of the greylevel distribution in the tool's region of interest. **TailBlack** is expressed as a percentage of the total number of pixels in the histogram before tails are removed. After its creation, the histogram is scanned, starting from bin 0. The bins at the dark end of the histogram are then cleared until the amount of pixels defined by **TailBlack** is reached.

Syntax

control.**TailBlack** [= Value As Single]

Type

Single

Range

Minimum: 0

Maximum: 100

Example

```
Dim Minimum As Integer
Dim Maximum As Integer
' Ignore the first 5% of darkest and brightest pixels in
' the distribution
Histogram1.TailBlack = 5
Histogram1.TailWhite = 5
Histogram1.Execute
' Get the darkest and brightest greylevels of the
' histogram
Minimum = Histogram1.TailBlackGreylevelValue
Maximum = Histogram1.TailWhiteGreylevelValue
```

TailBlackGreylevelValue

Description

When is **TailBlack** is used to ignore pixels at the dark end of the greylevel distribution, **TailBlackGreylevelValue** represents the darkest greylevel value that remains in the histogram after the tail is removed. Read only.

Syntax

control.**TailBlackGreylevelValue**

Type

Integer

Range

Minimum: 0

Maximum: 255

Example

```
Dim Minimum As Integer
Dim Maximum As Integer
```

```

' Ignore the first 5% of darkest and brightest pixels in
' the distribution
Histogram1.TailBlack = 5
Histogram1.TailWhite = 5
Histogram1.Execute
' Get the darkest and brightest greylevels of the
' histogram
Minimum = Histogram1.TailBlackGreylevelValue
Maximum = Histogram1.TailWhiteGreylevelValue

```

TailWhite

Description

Amount of pixels to ignore at the bright end of the greylevel distribution in the tool's region of interest. **TailWhite** is expressed as a percentage of the total number of pixels in the histogram before tails are removed. After its creation, the histogram is scanned, starting from bin 255. The bins at the bright end of the histogram are then cleared until the amount of pixels defined by **TailWhite** is reached.

Syntax

```
control.TailWhite [= Value As Single]
```

Type

Single

Range

Minimum: 0

Maximum: 100

Example

```

Dim Minimum As Integer
Dim Maximum As Integer
' Ignore the first 5% of darkest and brightest pixels in
' the distribution
Histogram1.TailBlack = 5
Histogram1.TailWhite = 5
Histogram1.Execute
' Get the darkest and brightest greylevels of the
' histogram
Minimum = Histogram1.TailBlackGreylevelValue
Maximum = Histogram1.TailWhiteGreylevelValue

```

TailWhiteGreylevelValue

Description

When is **TailWhite** is used to ignore pixels at the bright end of the greylevel distribution, **TailWhiteGreylevelValue** represents the brightest greylevel value that remains in the histogram after the tail is removed. Read only.

Syntax

```
control.TailWhiteGreylevelValue
```

Type

Integer

Range

Minimum: 0

Maximum: 255

Example

```

Dim Minimum As Integer
Dim Maximum As Integer
' Ignore the first 5% of darkest and brightest pixels in
' the distribution
Histogram1.TailBlack = 5
Histogram1.TailWhite = 5
Histogram1.Execute
' Get the darkest and brightest greylevels of the
' histogram
Minimum = Histogram1.TailBlackGreylevelValue
Maximum = Histogram1.TailWhiteGreylevelValue

```

ThresholdBlack

Description

Darkest greylevel value to consider when building the histogram. Greylevel values below **ThresholdBlack** are ignored during the histogram creation process. When a threshold is used and the tool is also configured to remove a percentage of pixels at the dark tail of the histogram (see the **TailBlack** property), the tail removal process begins to scan the histogram at the bin corresponding to **ThresholdBlack**, instead of starting at bin 0.

Syntax*control*.**ThresholdBlack****Type**

Integer

Range

Minimum: 0

Maximum: 255

Example

```

' Ignore greylevel values below 20 and above 235
Histogram1.ThresholdBlack = 20
Histogram1.ThresholdWhite = 235
Histogram1.Execute

```

ThresholdWhite

Description

Brightest greylevel value to consider when building the histogram. Greylevel values above **ThresholdWhite** are ignored during the histogram creation process. When a threshold is used and the tool is also configured to remove a percentage of pixels at the bright tail of the histogram (see the **TailWhite** property), the tail removal process begins to scan the histogram at the bin corresponding to **ThresholdWhite**, instead of starting at bin 255.

Syntax*control*.**ThresholdWhite****Type**

Integer

Range

Minimum: 0

Maximum: 255

Example

```
' Ignore greylevel values below 20 and above 235
Histogram1.ThresholdBlack = 20
Histogram1.ThresholdWhite = 235
Histogram1.Execute
```

ToolHeight

Description

Height of the tool rectangle.

Syntax*control*.**ToolHeight** [= *Value* As Single]**Type**

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = Histogram1.ToolHeight
Histogram2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax*control*.**ToolPositionX** [= *Value* As Single]**Type**

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = Histogram1.ToolPositionX
Histogram2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = Histogram1.ToolPositionY
Histogram2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = Histogram1.ToolRotation
Histogram2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = Histogram1.ToolWidth
Histogram2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = Histogram1.TransformRotation
Histogram2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = Histogram1.TransformScale
```

Histogram2.TransformScale = 1.10

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = Histogram1.TransformTranslationX
Histogram2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the

InstanceSelectionEnabled property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = Histogram1.TransformTranslationY
```


Histogram2.TransformTranslationY = 12.2

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSImageHistogram".

Example

```
Dim ControlType As String
ControlType = Histogram1.TypeName
```

UnitsLength

Description

Unit used to express lengths and coordinates.

Syntax

control.UnitsLength [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Remark(s)

This property is only applicable if **ModelTransformEnabled** or **CalibratedUnitsEnabled** is set to True.

Example

```
Dim Unit As hsUnitsLength
Unit = Histogram1.UnitsLength
```

Histogram2.**UnitsLength** = hsInch

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Histogram1.UnitsOrientation
Histogram2.UnitsOrientation = hsDegree
```

Variance

Description

Variance of the greylevel distribution of the pixels in the tool's region of interest that are included in the final histogram. Pixels removed from the histogram by tails or thresholds are not included in this calculation. Read only.

Syntax

control.**Variance**

Type

Single

Range

Minimum: 0

Maximum: 65535

Example

```
Dim StdDev As Single
StdDev = Histogram1.Variance
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Histogram1.Version
```

Methods

AboutBox

Description

Displays the Image Histogram tool About box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Return

No return value.

Example

```
Histogram1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

Example

```
Histogram1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the tool's property pages if they are displayed.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Histogram1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Shows the tool's property pages.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
Option	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was Successful.	True, False

Example

```
Dim Success As Boolean
Success = Histogram1.ShowProperties(True, _
    hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Image Processing Tool

Description

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

Properties

ArithmeticClippingMode

Description

Clipping mode applied by an arithmetic operation.

Syntax

control.ArithmeticClippingMode [= Value As hsImageProcessingClippingMode]

Type

hsImageProcessingClippingMode

Range

The valid range for this property is as follows:

Name	Value	Description
hsClippingNormal	0	Normal clipping method is used.
hsClippingAbsolute	1	Absolute clipping method is used.

Parameter(s)

Not applicable.

Example

```
Dim Clipping As hsImageProcessingClippingMode
Clipping = ImageProcessing1.ArithmeticClippingMode
ImageProcessing2.ArithmeticClippingMode = hsClippingNormal
```

ArithmeticConstant

Description

Constant applied by an arithmetic operation when no valid operand image is specified.

Syntax

control.ArithmeticConstant [= Value As Long]

Type

Long

Range

Unlimited

Parameter(s)

Not applicable.

Example

```
Dim ConstantValue As Long
ConstantValue = ImageProcessing1.ArithmeticConstant
ImageProcessing2.ArithmeticConstant = 128
```

ArithmeticScale

Description

Scaling factor applied by an arithmetic operation.

Syntax

control.**ArithmeticScale** [= Value As Single]

Type

Single

Range

Unlimited

Parameter(s)

Not applicable.

Example

```
Dim ScaleFactor As Single
ScaleFactor = ImageProcessing1.ArithmeticScale
ImageProcessing2.ArithmeticScale = 2.5
```

AssignmentConstant

Description

Constant applied by an assignment operation.

Syntax

control.**AssignmentConstant** [= Value As Long]

Type

Long

Range

Unlimited

Parameter(s)

Not applicable.

Example

```
Dim ConstantValue As Long
ConstantValue = ImageProcessing1.AssignmentConstant
ImageProcessing2.AssignmentConstant = 128
```

AssignmentHeight

Description

Constant value that defines the height of the output image.

Syntax

control.**AssignmentHeight** [= Value As Integer]

Type

Integer

Range

Unlimited but values from 1 to 2048 are preferable.

Parameter(s)

Not applicable.

Example

```
Dim DefaultHeight As Integer
DefaultHeight = ImageProcessing1.AssignmentHeight
ImageProcessing2.AssignmentHeight = 480
```

AssignmentWidth

Description

Constant value that defines the width of the output image.

Syntax

control.**AssignmentWidth** [= *Value* As Integer]

Type

Integer

Range

Unlimited but values from 1 to 2048 are preferable.

Parameter(s)

Not applicable.

Remark(s)

None

Example

```
Dim DefaultWidth As Integer
DefaultWidth = ImageProcessing1.AssignmentWidth
ImageProcessing2.AssignmentWidth = 640
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = ImageProcessing1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = ImageProcessing1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

FilteringClippingMode

Description

Clipping mode applied by a filtering operation.

Syntax

control.**FilteringClippingMode** [= *Value* As hsImageProcessingClippingMode]

Type

hsImageProcessingClippingMode

Range

The valid range for this property is as follows:

Name	Value	Description
hsClippingNormal	0	Normal clipping method is used.
hsClippingAbsolute	1	Absolute clipping method is used.

Parameter(s)

Not applicable.

Example

```
Dim Clipping As hsImageProcessingClippingMode
Clipping = ImageProcessing1.FilteringClippingMode
ImageProcessing2.FilteringClippingMode = _
    hsClippingAbsolute
```

FilteringCustomKernelAnchorX

Description

The horizontal position of the kernel anchor for a custom filtering operation.

Syntax

control.**FilteringCustomKernelAnchorX** [= *Value* As Integer]

Type

Integer

Range

Valid anchor horizontal position ranges from 1 to 9.

Parameter(s)

Not applicable.

Example

```
Dim AnchorX As Integer
AnchorX = ImageProcessing1.FilteringCustomKernelAnchorX
ImageProcessing2.FilteringCustomKernelAnchorX = 3
```

FilteringCustomKernelAnchorY

Description

The vertical position of the kernel anchor for a custom filtering operation.

Syntax

control.**FilteringCustomKernelAnchorY** [= *Value* As Integer]

Type

Integer

Range

Valid anchor vertical position ranges from 1 to 9.

Parameter(s)

Not applicable.

Example

```
Dim AnchorY As Integer
AnchorY = ImageProcessing1.FilteringCustomKernelAnchorY
```

ImageProcessing2.**FilteringCustomKernelAnchorY** = 3

FilteringCustomKernelHeight

Description

Kernel height applied by a custom filtering operation.

Syntax

control.**FilteringCustomKernelHeight** [= *Value* As Integer]

Type

Integer

Range

Valid height ranges from 1 to 9.

Parameter(s)

Not applicable.

Example

```
Dim KernelHeight As Integer
KernelHeight = ImageProcessing1. _
    FilteringCustomKernelHeight
ImageProcessing2.FilteringCustomKernelHeight = 9
```

FilteringCustomKernelValue

Description

Kernel values applied by a custom filtering operation.

Syntax

control.**FilteringCustomKernelValue**(column, line) [= *Value* As Integer]

Type

Integer

Range

Unlimited

Parameter(s)

Name	Description	Type	Range
Column	Column of the custom filtering kernel	Integer	From 1 to 9
Line	Line of the custom filtering kernel	Integer	From 1 to 9

Example

```
Dim KernelValue As Integer
KernelValue = ImageProcessing1. _
    FilteringCustomKernelValue(1,1)
ImageProcessing2.FilteringCustomKernelValue(2,3) = 25
```

FilteringCustomKernelWidth

Description

Kernel width applied by a custom filtering operation.

Syntax

control.**FilteringCustomKernelWidth** [= *Value* As Integer]

Type

Integer

Range

Valid width ranges from 1 to 9.

Parameter(s)

Not applicable.

Example

```
Dim KernelWidth As Integer
KernelWidth = ImageProcessing1. _
    FilteringCustomKernelWidth
ImageProcessing2.FilteringCustomKernelWidth = 2
```

FilteringKernelSize

Description

Kernel size applied by a fixed (predefined) filtering operation.

Syntax

control.**FilteringKernelSize** [= *Value* As Integer]

Type

Integer

Range

Valid sizes are: 3, 5 and 7.

Parameter(s)

Not applicable.

Example

```
Dim Kernel As Integer
Kernel = ImageProcessing1.FilteringKernelSize
ImageProcessing2.FilteringKernelSize = 5
```

FilteringScale

Description

Scaling factor applied by a filtering operation.

Syntax

control.**FilteringScale** [= *Value* As Single]

Type

Single

Range

Unlimited

Parameter(s)

Not applicable.

Example

```
Dim ScaleFactor As Single
ScaleFactor = ImageProcessing1.FilteringScale
ImageProcessing2.FilteringScale = 0.5
```

HistogramThreshold

Description

Threshold value applied by a histogram thresholding operation.

Syntax

control.HistogramThreshold [= *Value* As Long]

Type

Long

Range

Unlimited

Parameter(s)

Not applicable.

Example

```
Dim Threshold As Long
Threshold = ImageProcessing1.HistogramThreshold
ImageProcessing2.HistogramThreshold = 225
```

InputImage

Description

Name of the input Image.

Syntax

control.InputImage [= *Name* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ImageName As String
ImageName = ImageProcessing1.InputImage
ImageProcessing2.InputImage = "input_image1"
```

InputView

Description

Name of the View containing the input Image.

Syntax

control.InputView [= *Name* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ViewName As String
```

```
ViewName = ImageProcessing1.InputView
ImageProcessing2.InputView = "view0"
```

LogicalConstant

Description

Constant applied by a logical operation when no valid operand image is specified.

Syntax

```
control.LogicalConstant [= Value As Long]
```

Type

Long

Range

Unlimited

Parameter(s)

Not applicable.

Example

```
Dim ConstantValue As Long
ConstantValue = ImageProcessing1.LogicalConstant
ImageProcessing2.LogicalConstant = 128
```

MorphologicalNeighborhoodSize

Description

Neighborhood applied by a morphological operation.

Syntax

```
control.MorphologicalNeighborhoodSize [= Value As Integer]
```

Type

Integer

Range

Actually fixed to 3

Parameter(s)

Not applicable.

Example

```
Dim Neighborhood As Integer
Neighborhood = ImageProcessing1. _
    MorphologicalNeighborhoodSize
ImageProcessing2.MorphologicalNeighborhoodSize = 3
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

```
control.Name [= Name As String]
```

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ProcessName As String
ProcessName = ImageProcessing1.Name
ImageProcessing2.Name = "FirstProcessing"
```

OperandImage

Description

Name of the operand Image.

Syntax

control.**OperandImage** [= *Name* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ImageName As String
ImageName = ImageProcessing1.OperandImage
ImageProcessing2.OperandImage = "image1"
```

OperandView

Description

Name of the View containing the input operand Image.

Syntax

control.**OperandView** [= *Name* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ViewName As String
ViewName = ImageProcessing1.OperandView
ImageProcessing2.OperandView = "Operand View"
```

Operation

Description

Selected processing operation.

Syntax

control.**Operation** [= *Value* As hslImageProcessingOperation]

Type

hslImageProcessingOperation

Range

The valid range for this property is as follows:

Name	Value	Description
hsArithmeticAddition	0	Addition
hsArithmeticSubtraction	1	Subtraction
hsArithmeticMultiplication	2	Multiplication
hsArithmeticDivision	3	Division
hsArithmeticLightest	4	Lightest
hsArithmeticDarkest	5	Darkest
hsAssignmentInitialization	6	Initialization
hsAssignmentCopy	7	Copy
hsAssignmentInversion	8	Inversion
hsLogicalAnd	9	AND
hsLogicalNAnd	10	NAND
hsLogicalOr	11	OR
hsLogicalXOr	12	XOR
hsLogicalNOr	13	NOR
hsFilteringCustom	14	Custom filtering
hsFilteringAverage	15	Average filtering
hsFilteringLaplacian	16	Laplacian filtering
hsFilteringHorizontalSobel	17	Horizontal Sobel filtering
hsFilteringVerticalSobel	18	Vertical Sobel filtering
hsFilteringSharpen	19	Sharpen filtering
hsFilteringSharpenLow	20	SharpenLow filtering
hsFilteringHorizontalPrewitt	21	Horizontal Prewitt filtering
hsFilteringVerticalPrewitt	22	Vertical Prewitt filtering
hsFilteringGaussian	23	Gaussian filtering
hsFilteringHighPass	24	High Pass filtering
hsFilteringMedian	25	Median filtering
hsMorphologicalDilate	26	Dilate
hsMorphologicalErode	27	Erode
hsMorphologicalClose	28	Close
hsMorphologicalOpen	29	Open
hsHistogramEqualization	30	Histogram equalization
hsHistogramStretching	31	Histogram stretching
hsHistogramLightThreshold	32	Light threshold

Name	Value	Description
hsHistogramDarkThreshold	33	Dark threshold
hsTransformFFT	34	Fast Fourier Transform
hsTransformDCT	35	Discrete Cosine Transform

Parameter(s)

Not applicable.

Example

```
Dim Operation As hsImageProcessingOperation
Operation = ImageProcessing1.Operation
ImageProcessing2.Operation = hsMorphologicalErode
```

OutputImage

Description

Name of the Output Image.

Syntax

control.**OutputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ImageName As String
ImageName = ImageProcessing1.OutputImage
ImageProcessing2.OutputImage = "Processed Image"
```

OutputView

Description

Name of the View containing the Output Image.

Syntax

control.**OutputView** [= *Name* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim ViewName As String
ViewName = ImageProcessing1.OutputView
ImageProcessing2.OutputView = "View"
```

OverrideType

Description

Output image type when **OverrideTypeEnabled** property is set to True.

Syntax

control.OverrideType [= *Value* As hsImageProcessingType]

Type

hsImageProcessingType

Range

The valid range for this property is as follows:

Name	Value	Description
hsType8Bits	1	Unsigned 8-bit image.
hsType16Bits	10	Signed 16-bit image.
hsType32Bits	7	Signed 32-bit image.

Parameter(s)

Not applicable.

Example

```
Dim OverrideType As hsImageProcessingType
OverrideType = ImageProcessing1.OverrideType
ImageProcessing2.OverrideType = hsType32Bits
```

OverrideTypeEnabled

Description

Enables or disable the **OverrideType** property.

Syntax

control.OverrideTypeEnabled [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Enabled. The OverrideType represents the wanted output image type.
False	Disabled. The output image type is automatically selected to the same type as the output image if it exists otherwise, it is created with the same type as the input image.

Parameter(s)

Not applicable.

Example

```
Dim OverrideTypeEnabled As Boolean
OverrideTypeEnabled = ImageProcessing1.OverrideTypeEnabled
ImageProcessing2.OverrideTypeEnabled = False
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be perform using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = ImageProcessing1.ProcessingFormat
ImageProcessing2.ProcessingFormat = hsProcessingFormatGreyScale
```

ResultsDisplayMode

Description

Mode of output for the display in the Results tab.

Syntax

control.**ResultsDisplayMode** [= *Value* As hsImageProcessingDisplayMode]

Type

hsImageProcessingDisplayMode

Range

The valid range for this property is as follows:

Name	Value	Description
hsInputImage	0	Input Image will be displayed.
hsOperandImage	1	Operand Image will be displayed.
hsOutputImage	2	Output Image from last execution will be displayed.

Parameter(s)

Not applicable.

Example

```
Dim DisplayMode As hsImageProcessingDisplayMode
DisplayMode = ImageProcessing1.ResultsDisplayMode
```

ImageProcessing2.**ResultsDisplayMode** = hsInputImage

TransformFlags

Description

Sets the flag used by a transform operation, either FFT or DCT.

Syntax

control.**TransformFlags** [= Value As hsImageProcessingTransformFlags]

Type

hsImageProcessingTransformFlags

Range

The valid range for this property is as follows:

Name	Value	Description
hsTransformFlag2DLinear	0	Transform results output in linear 2D format
hsTransformFlag2DLogarithmic	1	Transform results in logarithmic scale
hsTransformFlag1DLinear	2	Transform results in 1D linear format
hsTransformFlagHistogram	3	Transform results as histogram

Parameter(s)

Not applicable.

Example

```
Dim TransformFlag As hsImageProcessingTransformFlags
TransformFlag = ImageProcessing1.TransformFlags
ImageProcessing2.TransformFlags = _
    hsTransformFlag2DLogarithmic
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Remark(s)

This property is always "HSImageProcessing".

Example

```
Dim TypeName As String
TypeName = ImageProcessing1.TypeName
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = ImageProcessing1.Version
```

Methods

AboutBox

Description

Displays the tool About box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Range

No return value.

Example

```
ImageProcessing1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

```
ImageProcessing1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the property pages of the tool if they are displayed.

Syntax

```
control.HideProperties( )
```

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = ImageProcessing1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Shows the property pages of the tool.

Syntax

```
control.ShowProperties(Modal As Boolean, [Variant Option])
```

Parameter(s)

Name	Description	Type	Range
Modal	A window is modal when it must be closed before the application can continue.	Boolean	False, True
Option	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements

Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
ImageProcessing1.ShowProperties_  
(True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Image Sharpness

Description

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

Properties

AutomaticCandidateCountEnabled

Description

When **AutomaticCandidateCountEnabled** is True the number of candidate measurement points is automatically determined according to the dimension of the tool's region of interest. When **AutomaticCandidateCountEnabled** is False, the number of candidate measurement points is set manually through the **CandidatePointsCount** property.

Syntax

control.**AutomaticCandidateCountEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The number of candidate measurement points is set automatically.
False	The number of candidate measurement points is set manually.

Example

```
' Set the number of candidate measurement points manually
Sharpness1.AutomaticCandidateCountEnabled = False
Sharpness2.CandidatePointsCount = 100
```

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
Result = Sharpness1.BilinearInterpolationEnabled
Sharpness2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = Sharpness1.CalibratedUnitsEnabled
Sharpness2.CalibratedUnitsEnabled = False
```

CandidatePointsCount

Description

Sets the number of candidate locations where the tool tries to evaluate the sharpness. When the tool is executed, it scans the region of interest and identifies a number of candidate locations (equal to **CandidatePointsCount**) where the local standard deviation is the highest. The local sharpness is then evaluated at each of the candidate location that has a local standard deviation above **StandardDeviationThreshold**. The number of locations where the sharpness is actually measured is return by the **MeasurementPointsCount** property. When the **AutomaticCandidateCountEnabled** property is True, the number of candidate measurement points is determined automatically according to the size of the region of interest and **CandidatePointsCount** is read only.

Syntax

control.**CandidatePointsCount** [= *Value* As Long]

Type

Long

Range

Greater than 0

Example

```
' Set the number of candidate measurement points manually
Sharpness1.AutomaticCandidateCountEnabled = False
Sharpness2.CandidatePointsCount = 100
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Sharpness1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Greater than 0.0

Remark(s)

This property gives the total elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Sharpness1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Sharpness1.ErrorObject.Description
```

InputImage

Description

Name of the image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Sharpness1.InputImage
Sharpness2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the image on which the tool is applied.

Syntax

control.**InputImage** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Sharpness1.InputImage
Sharpness2.InputImage = "image1"
```

InputView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Sharpness1.InputView
Sharpness2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.InputView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Sharpness1.InputView
Sharpness2.InputView = "view0"
```

InputInstanceScene

Description

Name of the scene — created by the Locator tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool in order to inspect a selected object instance.

Syntax

control.InputInstanceScene [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = Sharpness1.InputInstanceScene
```

```
Sharpness2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View from which the Instance Scene is retrieved. See **InputInstanceScene** for more details.

Syntax

```
control.InputInstanceView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Sharpness1.InputInstanceView
Sharpness2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

```
control.InputModelScene [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = Sharpness1.InputModelScene
Sharpness2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View from which the Model Scene is retrieved. See **InputModelScene** for more details.

Syntax

```
control.InputModelView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
```

```
ModelViewName = Sharpness1.InputModelView
Sharpness2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = Sharpness1.InstanceSelectionEnabled
Sharpness2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = Sharpness1.InstanceSelectionIndex
Sharpness2.InstanceSelectionIndex = 0
```


KernelSize

Description

Sets the size of the operator that is used to compute the local standard deviation in the candidate points identification process. Also sets the size of the operator used to evaluate the sharpness at the candidate locations kept as measurement points. A **KernelSize** value of n configures the tool to use an operator with a size of $n \times n$ pixels

Syntax

control.**KernelSize** [= *Value* As Integer]

Type

Integer

Range

Minimum: 2

Maximum: 16

Example

```
' Identify the candidate measurement points and compute
' the sharpness using a 8x8 pixels operator
Sharpness1.KernelSize = 8
Sharpness1.Execute
```

MeasurementPointsCount

Description

The number of points where the local sharpness is evaluated; this number is always equal or smaller than **CandidatePointsCount**. Can also be zero. When the tool is executed, it scans the region of interest and identifies a number of candidate locations (equal to **CandidatePointsCount**) where the local standard deviation is the highest. The local sharpness is then evaluated at each of the candidate location that has a local standard deviation above **StandardDeviationThreshold**. The number of locations where the sharpness is actually measured is return by the **MeasurementPointsCount** property. Read only.

Syntax

control.**MeasurementPointsCount**

Type

Long

Range

Minimum: 0

Maximum: CandidatePointsCount

Example

```
Dim MeasurePoints As Long
' Set the number of candidate measurement points manually
Sharpness1AutomaticCandidateCountEnabled = False
Sharpness1.CandidatePointsCount = 100
Sharpness1.Execute
' Out of the 100 candidate locations, get the number of
' locations where the sharpness was actually measured
MeasurePoints = Sharpness1.MeasurementPointsCount
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a model image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.ModelTransformEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = Sharpness1.ModelTransformEnabled
Sharpness2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.Name [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = Sharpness1.Name
Sharpness2.Name = "SharpA"
```

SamplingStep

Description

Step used in the tool's last execution to sample the region of interest from the input image. When the tool is model-based this step is expressed in calibrated units, otherwise this step is expressed either in pixels or in calibrated units as defined by the **CalibratedUnitsEnabled** and the **ModelTransformEnabled** properties. All pixels in the sampled rectangle are square and of the same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax*control*.**SamplingStep****Type**

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

Dim Step As Single

Step = Sharpness1.**SamplingStep**

SamplingStepCustom

Description

When **SamplingStepCustomEnabled** is True, defines the sampling step used to sample the region of interest from the input image. When **SamplingStepCustomEnabled** is False, **SamplingStepDefault** is used instead as the sampling step.

Syntax*control*.**SamplingStepCustom** [= *Value* As Single]**Type**

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

Dim Step As Single

Step = Sharpness1.**SamplingStepCustom**Sharpness2.**SamplingStepCustom** = 2.1

SamplingStepCustomEnabled

Description

When enabled, the tool uses the user-defined sampling step (**SamplingStepCustom**) instead of the default optimal sampling step to sample the region of interest from the input image.

Syntax*control*.**SamplingStepCustomEnabled** [= *Value* As Boolean]**Type**

Boolean

Range

Value	Description
False	The tool uses the default optimal sampling step.

Value	Description
True	The default sampling step is overridden by SamplingStepCustom .

Example

```
Dim Enabled As Boolean
Enabled = Sharpness1.SamplingStepCustomEnabled
Sharpness2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default optimal sampling step used by the tool to sample the region of interest from the input image. This sampling step is used by the tool if **SamplingStepCustomEnabled** is False. Read only.

Syntax

control.**SamplingStepDefault**

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = Sharpness1.SamplingStepDefault
```

Sharpness

Description

Average sharpness computed in the region of interest. When the tool is executed, it scans the region of interest and identifies a number of candidate locations (equal to **CandidatePointsCount**) where the local standard deviation is the highest. The local sharpness is then evaluated at each of the candidate location that has a local standard deviation above **StandardDeviationThreshold**. The tool then computes the average of all the local sharpness values computed at every measurement point and returns it through the **Sharpness** property. Read only.

Syntax

control.**Sharpness**

Type

Single

Range

Minimum: 0 (Maximum blurriness)
Maximum:1000 (Maximum sharpness)

Example

```
Dim Sharpness As Single
Sharpness1.Execute
Sharpness = Sharpness1.Sharpness
```

SharpnessHistory

Description

SharpnessHistory is an indexed property that stores a history of the last **Sharpness** values computed by the tool. The number of sharpness values stored in the history is equal to **SharpnessHistorySize**. **Sharpness(0)** is the latest sharpness value and **Sharpness(SharpnessHistorySize - 1)** is the oldest. Read-only.

Syntax

control.**SharpnessHistory**(*Index* As Long)

Type

Single

Range

Minimum: 0 (Maximum blurriness)

Maximum:1000 (Maximum sharpness)

Parameter(s)

Name	Description	Type	Range
Index	Index of the sharpness value.	Long	Minimum: 0 Maximum: SharpnessHistorySize -1

Example

```
Dim i As Integer
Dim Sharpness As Single
' Compute the average of the last 10 sharpness values
' returned by the tool
For i = 0 to 9
    Sharpness = Sharpness1.SharpnessHistory(i)
Next i
Sharpness = Sharpness / 10.0
```

SharpnessHistoryScale

Description

The scale of the Sharpness History viewer and Sharpness gauge in the Results tab of the properties page.

Syntax

control.**SharpnessHistoryScale** [= *Value* As hsSharpnessHistoryScale]

Type

hsSharpnessHistoryScale

Range

The valid range for this property is as follows:

hsSharpnessHistoryScale	Value	Description
hsScaleAutomatic	-1	The scale of the gauge is adjusted automatically.
hsScale0100c	0	The gauge shows sharpness values in the range 0 to 100. Values above 100 are clipped.

hsSharpnessHistoryScale	Value	Description
hsScale0250	1	The gauge shows sharpness values in the range 0 to 250. Values above 250 are clipped.
hsScale0500	2	The gauge shows sharpness values in the range 0 to 500. Values above 500 are clipped.
hsScale1000	3	The gauge shows sharpness values in the range 0 to 1000. Values above 1000 are clipped.

Example

```
' Set the sharpness gauge in its automatic scale mode
Sharpness1.SharpnessHistoryScale = hsScaleAutomatic
```

SharpnessHistorySize

Description

The size of the array used to store the history of previous sharpness values are stored. Read only.

Syntax

```
control.SharpnessHistorySize
```

Type

Long

Range

Always 256

Example

```
Dim Size As Long
Size = Sharpness1.SharpnessHistorySize
```

SharpnessPeak

Description

Maximum **Sharpness** value computed by the tool since the last call to the **ResetHistory** method. Read only

Syntax

```
control.SharpnessPeak
```

Type

Single

Range

Minimum: 0 (Maximum blurriness)
Maximum:1000 (Maximum sharpness)

Example

```
Dim Peak As Single
Peak = Sharpness1.SharpnessPeak
```

StandardDeviationThreshold

Description

Threshold used to validate candidate measurement points before computing their local sharpness. When the tool is executed, it scans the region of interest and identifies a number of

candidate locations (equal to **CandidatePointsCount**) where the local standard deviation is the highest. The local sharpness is then evaluated at each of the candidate location that has a local standard deviation above **StandardDeviationThreshold**.

Syntax

control.**StandardDeviationThreshold** [= *Value* As Single]

Type

Single

Range

Minimum: 0

Maximum: 255

Example

```
' Ignore candidate measurement points with a local
' standard deviation lower than 50
Sharpness1.SharpnessHistoryScale = hsScaleAutomatic
Sharpness1.StandardDeviationThreshold = 50
Sharpness1.Execute
```

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.**ToolHeight** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = Sharpness1.ToolHeight
Sharpness2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionX** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = Sharpness1.ToolPositionX
Sharpness2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = Sharpness1.ToolPositionY
Sharpness2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = Sharpness1.ToolRotation
Sharpness2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = Sharpness1.ToolWidth
Sharpness2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = Sharpness1.TransformRotation
Sharpness2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = Sharpness1.TransformScale
```

Sharpness2.**TransformScale** = 1.10

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation.

This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformTranslationX** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = Sharpness1.TransformTranslationX
Sharpness2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input image. This transformation is used to fit the tool to an object in the input image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the

InstanceSelectionEnabled property is set to True.

Syntax

control.**TransformTranslationY** [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = Sharpness1.TransformTranslationY
```

Sharpness2.TransformTranslationY = 12.2

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSImageSharpness".

Example

```
Dim ControlType As String
ControlType = Sharpness1.TypeName
```

UnitsLength

Description

Unit used to express lengths and coordinates.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Remark(s)

This property is only applicable if **ModelTransformEnabled** or **CalibratedUnitsEnabled** is set to True.

Example

```
Dim Unit As hsUnitsLength
Unit = Sharpness1.UnitsLength
```

Sharpness2.**UnitsLength** = hsInch

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = Sharpness1.UnitsOrientation
Sharpness2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Sharpness1.Version
```

Methods

AboutBox

Description

Displays the Image Sharpness tool About box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Return

No return value.

Example

Sharpness1.**AboutBox**

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

Sharpness1.**Execute**

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the tool's property pages if they are displayed.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Sharpness1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

ResetHistory

Description

Resets the **SharpnessPeak** and **SharpnessHistory** properties.

Syntax

```
control.ResetHistory( )
```

Parameter(s)

None

Return

No return value.

Example

```
Sharpness1.ResetHistory
```

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Shows the tool's property pages.

Syntax

```
control.ShowProperties(Modal As Boolean, [Variant Option])
```

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = Sharpness1.ShowProperties(True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Line Finder

Description

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

Properties

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is Model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

Dim Result As Boolean

Result = LineFinder1.**CalibratedUnitsEnabled**

LineFinder2.**CalibratedUnitsEnabled** = False

ConformityTolerance

Description

Maximum local deviation between the expected edges of a line and the edges actually detected in the input image. It corresponds to the maximum distance in calibrated units by which a matched edge can deviate from either side of its expected position in the line. Expressed in the calibrated units specified by the **UnitsLength** property. This property can only be set when **UseDefaultConformityTolerance** is set to false. Read only otherwise.

Syntax

control.**ConformityTolerance**

Type

Single

Range

Minimum: **MinimumConformityTolerance**
 Maximum: **MaximumConformityTolerance**

Example

```
Dim ConformityTolerance As Single
ConformityTolerance = LineFinder1.ConformityTolerance
LineFinder2.ConformityTolerance = 2.0
```

ContrastThreshold

Description

Defines the minimum contrast needed for an edge to be detected in the input image and used for arc computation. This threshold is expressed in terms of a step in greylevel values. The property is read only except when **ContrastThresholdMode** is set to **hsContrastThresholdFixedValue**.

Syntax

control.**ContrastThreshold** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1
 Maximum: 255

Remark(s)

By default, the tool selects a **ContrastThresholdMode** based on image content to provide flexibility to variations in image lighting conditions and contrast. Adaptive threshold modes are generally recommended. A fixed value Contrast Threshold should only be used when adaptive values do not provide satisfactory results.

Example

```
Dim CurrentThreshold As Integer
CurrentThreshold = LineFinder1.ContrastThreshold
LineFinder2.ContrastThresholdMode = _
    hsContrastThresholdModeFixedValue
LineFinder2.ContrastThreshold = 10
```

ContrastThresholdMode

Description

Selects the method used to compute the threshold used for detecting edges in the input image.

Syntax

control.**ContrastThresholdMode** [= *Value* As hsContrastThresholdMode]

Type

hsContrastThresholdMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContrastThresholdAdaptiveLowSensitivity	0	Uses a low sensitivity adaptive threshold for detecting edges. Adaptive Low Sensitivity reduces the amount of noisy edges but may also cause significant edges to be undetected.
hsContrastThresholdAdaptiveNormalSensitivity	1	Uses a normal sensitivity adaptive threshold for detecting edges.
hsContrastThresholdAdaptiveHighSensitivity	2	Uses a high sensitivity adaptive threshold for detecting edges. Adaptive High Sensitivity can help detect weak-contrast edges but also increases the amount of noisy edges.
hsContrastThresholdFixedValue	3	Uses a fixed value threshold for detecting edges.

Remark(s)

By default, the tool selects a **ContrastThresholdMode** based on image content to provide flexibility to variations in image lighting conditions and contrast. Adaptive threshold modes are generally recommended. A fixed-value contrast threshold should only be used when adaptive values do not provide satisfactory results.

Example

```
Dim CurrentMode As hsContrastThresholdMode
CurrentMode = LineFinder2.ContrastThresholdMode
LineFinder2.ContrastThresholdMode = hsContrastThresholdModeFixedValue
LineFinder2.ContrastThresholdMode = 10
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.

Constant	Value	Description
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input Image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```

Dim CurrentSystem As hsCoordinateSystem
' Get the current coordinate system
CurrentSystem = LineFinder1.CoordinateSystem
' Express the results in the Object coordinate system
LineFinder1.CoordinateSystem = hsCoordinateSystemObject

```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

DefaultConformityTolerance

Description

Default value for **ConformityTolerance** computed by the tool according to the calibration, the edge detection parameters, and the search parameters. See also **ConformityTolerance** and **UseDefaultConformityTolerance**. Read only.

Syntax

control.**DefaultConformityTolerance**

Type

Single

Range

Greater than zero.

Remark(s)

This default value is used for **ConformityTolerance** when **UseDefaultConformityTolerance** is set to True.

Example

```
Dim Default As Single
Default = LineFinder1.DefaultConformityTolerance
```

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = LineFinder1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: Unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = LineFinder1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & LineFinder.ErrorObject.Description
```

InputImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

```
control.InputImage [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = LineFinder1.InputImage
LineFinder2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

```
control.InputImage [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = LineFinder1.InputImage
LineFinder2.InputImage = "image1"
```

InputView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

```
control.InputView [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = LineFinder1.InputView
LineFinder2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = LineFinder1.InputView
LineFinder2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the **Locator** tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = LineFinder1.InputInstanceScene
LineFinder2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = LineFinder1.InputInstanceView
LineFinder2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = LineFinder1.InputModelScene
LineFinder2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View that provides the Model Scene. See **InputModelScene** for more details.

Syntax

control.**InputModelView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = LineFinder1.InputModelView
LineFinder2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

```
control.InstanceSelectionEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = LineFinder1.InstanceSelectionEnabled
LineFinder2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = LineFinder1.InstanceSelectionIndex
LineFinder2.InstanceSelectionIndex = 0
```

LineAngle

Description

Angle of the found line, expressed in the currently selected coordinate system. A line is defined as the line passing through the point coordinates **LineVectorPointPositionX** and **LineVectorPointPositionY** at the angle given by **LineAngle**. Read only.

Syntax

```
control.LineAngle
```

Type

Single

Range

-180 to 180 degrees

Example

```
Dim LineAngle as Single
```

`LineAngle = LineFinder1.LineAngle`

LineAverageContrast

Description

Average contrast between light and dark pixels on either side of the found line, expressed in greylevel values. Read only.

Syntax

`control.LineAverageContrast`

Type

Integer

Range

Minimum: greater than 0.

Maximum: 255

Example

```
Dim AverageContrast As Integer
LineFinder1.Execute
AverageContrast = LineFinder1.LineAverageContrast
```

LineEndPointPositionX

Description

X coordinate of the line end point, expressed in the currently selected coordinate system. A found line is bounded at either end by a start point and an end point. The line's start point, defined by **LineStartPointPositionX** and **LineStartPointPositionY**, is the point of intersection having the lowest Y Tool coordinate value. Conversely, the line's end point, defined by **LineEndPointPositionX** and **LineEndPointPositionY**, is the intersection point between line and Search Area that has the highest Y Tool coordinate value. Read only.

Syntax

`control.LineEndPointPositionX`

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim LineEndX as Single
LineEndX = LineFinder1.LineEndPointPositionX
```

LineEndPointPositionY

Description

Y coordinate of the line end point, expressed in the currently selected coordinate system. A found line is bounded at either end by a start point and an end point. The line's start point,

defined by **LineStartPointPositionX** and **LineStartPointPositionY**, is the point of intersection having the lowest Y Tool coordinate value. Conversely, the line's end point, defined by **LineEndPointPositionX** and **LineEndPointPositionY**, is the intersection point between line and Search Area that has the highest Y Tool coordinate value. Read only.

Syntax

control.**LineEndPointPositionY**

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim LineEndY as Single
LineEndY = LineFinder1.LineEndPointPositionY
```

LineFitQuality

Description

Normalized average error between the calculated line and the actual edges matched to the found line. Fit quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to conformity tolerance. Read only.

Syntax

control.**LineFitQuality**

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Example

```
Dim FitQuality As Single
FitQuality = LineFinder1.LineFitQuality
```

LineFound

Description

When True, indicates that a line was found. Read only.

Syntax

control.**LineFound**

Type

Boolean

Range

Not applicable.

Example

```
Dim LineFound as Boolean
LineFound = LineFinder1.LineFound
```

LineMatchQuality

Description

Percentage of edges actually matched to the found line. Match Quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that edges were matched for every point along the found line. Similarly, a value of 0.2 means edges were matched to 20% of the points along the found line. Read only.

Syntax

control.**LineMatchQuality**

Type

Single

Range

Minimum: greater than zero.

Maximum: 1.0

Example

```
Dim MatchQuality As Single
MatchQuality = LineFinder1.LineMatchQuality
```

LineStartPointPositionX

Description

X coordinate of the line start point, expressed in the currently selected coordinate system. A found line is bounded at either end by a start point and an end point. The line's start point, defined by **LineStartPointPositionX** and **LineStartPointPositionY**, is the point of intersection having the lowest Y Tool coordinate value. Conversely, the line's end point, defined by **LineEndPointPositionX** and **LineEndPointPositionY**, is the intersection point between line and Search Area that has the highest Y Tool coordinate value. Read only

Syntax

control.**LineStartPointPositionX**

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim LineStartX as Single
LineStartX = LineFinder1.LineStartPointPositionX
```

LineStartPointPositionY

Description

Y coordinate of the line start point, expressed in the currently selected coordinate system. A found line is bounded at either end by a start point and an end point. The line's start point, defined by **LineStartPointPositionX** and **LineStartPointPositionY**, is the point of intersection having the lowest Y Tool coordinate value. Conversely, the line's end point, defined by **LineEndPointPositionX** and **LineEndPointPositionY**, is the intersection point between line and Search Area that has the highest Y Tool coordinate value. Read only.

Syntax

control.**LineStartPointPositionY**

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim LineStartY as Single
LineStartY = LineFinder1.LineStartPointPositionY
```

LineVectorPointPositionX

Description

X coordinate of the point that defines the found line, expressed in the currently selected coordinate system. A line is defined as the line passing through the point coordinates **LineVectorPointPositionX** and **LineVectorPointPositionY** at the angle given by **LineAngle**. Read only.

Syntax

control.**LineVectorPointPositionX**

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim LinePointX as Single
LinePointX = LineFinder1.LineVectorPointPositionX
```

LineVectorPointPositionY

Description

Coordinate of the point that defines the found line, expressed in the currently selected coordinate system. A line is defined as the line passing through the point coordinates

LineVectorPointPositionX and **LineVectorPointPositionY** at the angle given by **LineAngle**. Read only.

Syntax

control.**LineVectorPointPositionY**

Type

Single

Range

Unbounded

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim LinePointY as Single
LinePointY = LineFinder1.LineVectorPointPositionY
```

MaximumAngleDeviation

Description

Maximum deviation relates to the deviation in angle between hypothesis and the line found by the tool. By default the Line Finder accepts a 20 degree deviation. However, the tool uses the defined MaximumAngleDeviation value to test the hypothesis and refine the pose of the found line. Expressed in the orientation units specified by the **UnitsOrientation** property.

Syntax

control.**MaximumAngleDeviation** [= Value As Single]

Type

Single

Range

Minimum: 0 degrees

Maximum: 20 degrees

Example

```
Dim Value as Single
Value = LineFinder1.MaximumAngleDeviation
LineFinder2.MaximumAngleDeviation = 10
```

MaximumConformityTolerance

Description

Maximum value allowed for the **ConformityTolerance** property. Read only.

Syntax

control.**MaximumConformityTolerance**

Type

Single

RangeGreater than **MinimumConformityTolerance**.**Remark(s)**

This property is computed by the Line Finder by analyzing the calibration, edge detection parameters, and the search parameters. See also **ConformityTolerance** and **DefaultConformityTolerance**.

Example

```
Dim MaxConformityTolerance As Single
MaxConformityTolerance = LineFinder1.MaximumConformityTolerance
```

MessageCount

Description

The number of messages issued during the last execution. The identification number and description of each message is retrieved using respectively the **MessageNumber** and **MessageDescription** properties. Read only.

Syntax

```
control.MessageCount
```

Type

Integer

Range

Minimum: 0

Maximum: Unlimited

Example

```
Dim m As Integer
For m = 0 To LineFinder1.MessageCount - 1
    Description.Text = LineFinder1.MessageDescription(m)
    Number.Text = Str(LineFinder1.MessageNumber(m))
Next m
```

MessageDescription

Description

The description of a message issued during the last execution. The identification number of the message corresponding to this description is retrieved using the **MessageNumber** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

```
control.MessageDescription(Index As Integer)
```

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```

Dim m As Integer
For m = 0 To LineFinder1.MessageCount - 1
    Description.Text = LineFinder1.MessageDescription(m)
    Number.Text = Str(LineFinder1.MessageNumber(m))
Next m

```

MessageNumber

Description

The identification number of a message issued during the last execution. The description of the message corresponding to this identification number is retrieved using the **MessageDescription** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageNumber**(*Index* As Integer)

Type

Integer

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```

Dim m As Integer
For m = 0 To LineFinder1.MessageCount - 1
    Description.Text = LineFinder1.MessageDescription(m)
    Number.Text = Str(LineFinder1.MessageNumber(m))
Next m

```

MinimumConformityTolerance

Description

Minimum value allowed for the **ConformityTolerance** property. Read only.

Syntax

control.**MinimumConformityTolerance**

Type

Single

Range

Minimum: greater than zero.

Maximum: lower than **MaximumConformityTolerance**.

Remark(s)

This property is computed by the Locator by analyzing the calibration, the edge detection parameters, and the search parameters. See also **ConformityTolerance** and **DefaultConformityTolerance**.

Example

```
Dim MinConformityTolerance As Single
```

```
MinConformityTolerance = LineFinder1.MinimumConformityTolerance
```

MinimumLinePercentage

Description

Minimum percentage of line contours that need to be matched for a line hypothesis to be considered as valid.

Syntax

```
control.MinimumLinePercentage [= Value As Single]
```

Type

Single

Range

Minimum: greater than zero.

Maximum: 100.0

Example

```
Dim MinModelPercent As Single
```

```
MinModelPercent = LineFinder1.MinimumModelPercentage
```

```
LineFinder2.MinimumModelPercentage = 50.0
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a Model Image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

```
control.ModelTransformEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = LineFinder1.ModelTransformEnabled
LineFinder2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = LineFinder1.Name
LineFinder2.Name = "LineFindA"
```

OutputEntity

Description

Name of the entity created by the tool. This entity is optionally output according to the value of the **OutputEntityEnabled** property.

Syntax

control.**OutputEntity** [= *Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim Line As String
Line = LineFinder1.OutputEntity
LineFinder1.OutputEntity = "Line0"
```

OutputEntityEnabled

Description

Specifies if a found entity will be output to the runtime database.

Syntax

control.**OutputEntityEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	The entity will not be output.
True	The entity will be output.

Example

```
Dim Enable As Boolean
Enable = LineFinder1.OutputEntityEnabled
LineFinder1.OutputEntityEnabled = False
```

OutputEntityView

Description

Name of the View to which the Output Entity is output. See **OutputEntity** for more details.

Syntax

```
control.OutputEntityView [= Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = LineFinder1.OutputEntityView
LineFinder2.OutputEntityView = "LineView0"
```

PolarityMode

Description

Selects the type of polarity accepted for finding a line. Polarity identifies the change in greylevel values along the tool's X axis, in the positive direction.

Syntax

```
control.PolarityMode [= Value As hsPolarityMode]
```

Type

hsPolarityMode

Range

The valid range for this property is as follows:

hsPolarityMode	Value	Description
<i>hsDarkToLight</i>	0	The Line Finder searches only for line instances occurring at a dark to light transition in greylevel values.
<i>hsLightToDark</i>	1	The Line Finder searches only for line instances occurring at a light to dark transition in greylevel values.
<i>hsEither</i>	2	The Line Finder searches only for line instances occurring either at a light to dark or dark to light transition in greylevel values.

hsPolarityMode	Value	Description
<i>hsDontCare</i>	3	The Line Finder searches only for line instances occurring at any transition in greylevel values including reversals in contrast along the line, for example on an unevenly colored background.

Remark(s)

Enabling **hsEither** can significantly increase the tool's execution time.

Example

```
' Search for a line at a dark to light transition
LineFinder1.PolarityMode = hsLightToDark
LineFinder1.Execute
```

PositioningLevel

Description

Configurable effort level of the entity positioning process. A value of 0 will provide coarser positioning and lower execution time. Conversely, a value of 10 will provide high accuracy positioning of line entities.

Syntax

control.**PositioningLevel** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0
Maximum: 10

Example

```
Dim PositioningLevel As Integer
PositioningLevel = LineFinder1.PositioningLevel
LineFinder2.PositioningLevel = 5
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be performed using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SearchMode

Description

Specifies the method used by the tool to select a hypothesis.

Syntax

control.**SearchMode** [= *Value* as hsSearchMode]

Type

hsSearchMode

Range

The valid range for this property is as follows:

hsSearchMode	Value	Description
hsBestLine	0	Selects the best line according to hypothesis strength.
hsLineClosestToGuideline	1	Selects the line hypothesis closest to the Guideline.
hsLineWithMaximumNegativeXOffset	2	Selects the line hypothesis closest to the Search Area bound that is at maximum negative X offset.
hsLineWithMaximumPositiveXOffset	3	Selects the line hypothesis closest to the Search Area bound that is at maximum positive X offset.

Example

```
' Search for Line closest to guideline
Dim Mode As hsSearchMode
LineFinder2.SearchMode = hsLineClosestToGuideline
Mode = LineFinder2.SearchMode
```

ShowEdges

Description

If true, edges in the Search Area are displayed in the Results display. Candidate edges for hypothesis generation are displayed in a different color (green) than other edges (blue).

Syntax

control.ShowEdges [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = LineFinder1.ShowEdges
LineFinder1.ShowEdges = False
```

ShowEntityFound

Description

When **ShowEntityFound** is set to true, the entity found by the tool is displayed in the Results display.

Syntax

control.ShowEntityFound [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = LineFinder1.ShowEntityFound
LineFinder1.ShowEntityFound = False
```

ShowEntityFoundAs

Description

Mode used to represent the entity in the Results display.

Syntax

control.ShowEntityFoundAs [= *Value* As hsEntityOutputMode]

Type

Integer

Range

The valid range for this property is as follows:

hsEntityOutputMode	Value	Description
hsFittedEntity	0	The line is represented by the entire line, fitted to the actual edges detected in the input image.

hsEntityOutputMode	Value	Description
hsMatchedEntity	1	The line is represented by the sections of the line that have been matched to edges actually detected in the input image.

Example

```
Dim ShowAs as hsEntityOutputMode
Linefinder.ShowEntityFoundAs = hsFittedEntity
ShowAs = LineFinder2.ShowEntityFoundAs
```

ShowImage

Description

When **ShowImage** is set to True, the input image is shown in the Results display.

Syntax

control.**ShowImage** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
LineFinder1.ShowImage = True
```

ShowGreyScaleImage

Description

When **ShowImage** is set to True, the input image is shown in the Results display.

Syntax

control.**ShowImage** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
LineFinder1.ShowImage = True
```

ShowSearchArea

Description

Specifies if the Search Area is shown in the Results Viewer.

Syntax

control.**ShowSearchArea** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = LineFinder1.ShowSearchArea
LineFinder1.ShowSearchArea = False
```

SubsamplingLevel

Description

Subsampling level used to detect edges that are used by the tool to generate hypotheses. High values provide a coarser search and lower execution time than lower values.

Syntax

control.**SubsamplingLevel** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1
Maximum: 8

Example

```
Dim SearchLevel As Single
SearchLevel = LineFinder1.SubsamplingLevel
LineFinder2.SubsamplingLevel = 4
```

ToolGuidelineOffset

Description

The X-axis offset of the Guideline marker from the center of the tool Search Area.

Syntax

control.**ToolGuidelineOffset** [= *Value* As Single]

Type

Single

Range

Maximum: $+0.5 \times \text{ToolWidth}$
Minimum: $-0.5 \times \text{ToolWidth}$

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
LineFinder1.ToolWidth = 25
LineFinder1.ToolGuidelineOffset = -5
LineFinder1.Execute
```

ToolHeight

Description

Height of the tool Search Area, defined along the tool Y-axis.

Syntax

control.**ToolHeight** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = LineFinder1.ToolHeight
LineFinder2.ToolHeight = 15.2
```

ToolPositionX

Description

X coordinate of the position of the tool Search Area

Syntax*control*.**ToolPositionX** [= Value As Single]**Type**

Single

Range

Unbounded

Remark(s)

The property is expressed in length units, and with respect to a coordinate system, as specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim X_Pos As Single
X_Pos = LineFinder1.ToolPositionX
LineFinder2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the position of the tool Search Area.

Syntax*control*.**ToolPositionY** [= Value As Single]**Type**

Single

Range

Unbounded

Remark(s)

The property is expressed in length units, and with respect to a coordinate system, as specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = LineFinder1.ToolPositionY
LineFinder2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool Search Area.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system, as specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = LineFinder1.ToolRotation
LineFinder2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool Search Area, defined along the X-axis.

Syntax

control.**ToolWidth** [= *Value* As Single]

Type

Single

Range

Greater than 0.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property, if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width as Single
Width = LineFinder1.ToolWidth
LineFinder2.ToolWidth = 15.2
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = LineFinder1.TransformRotation
LineFinder2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: Greater than zero.
Maximum: Unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = LineFinder1.TransformScale
LineFinder2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = LineFinder1.TransformTranslationX
```

LineFinder2.TransformTranslationX = 22.0

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = LineFinder1.TransformTranslationY
LineFinder2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSLineFinder".

Example

```
Dim ControlType As String
ControlType = LineFinder1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.UnitsLength [= Value As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = LineFinder1.UnitsLength
LineFinder2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

```
control.UnitsOrientation [= Value As hsUnitsOrientation]
```

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = LineFinder1.UnitsOrientation
LineFinder2.UnitsOrientation = hsDegree
```

UseDefaultConformityTolerance

Description

Specifies whether the default conformity tolerance returned by the **DefaultConformityTolerance** property is used instead of the user-defined conformity

tolerance set by the **ConformityTolerance** property. When set to True, the default conformity tolerance is used.

Syntax

control.UseDefaultConformityTolerance [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim UseDefaultCT As Boolean
UseDefaultCT = LineFinder1.UseDefaultConformityTolerance
LineFinder2.UseDefaultConformityTolerance = False
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.Version

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = LineFinder1.Version
```

Methods

AboutBox

Description

Displays the Line Finder tool **About** box.

Syntax

control.AboutBox

Parameter(s)

None

Return

No return value.

Example

```
LineFinder1.AboutBox
```


ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Line Finder process.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

LineFinder1.**Execute**

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

control.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

control.**HideProperties**()

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = LineFinder1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and what is the behavior of the control when the OK button is clicked. When this parameter is not provided all the elements of the bitfield are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

Dim Success As Boolean

Success = LineFinder1.**ShowProperties**(True, hsShowOK + hsShowExecute)

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Locator

The Locator finds and locates objects based on the geometry of their contours. Scale factor, orientation, and position are provided for each located instance. Models are built using the integrated Model Editor.

Properties

AdvancedParametersVisible

Description

Specifies if the Advanced Parameters for the Locator are visible. When **AdvancedParametersVisible** is set to True, the Locator is in *Advanced* setup mode and all Locator parameters are visible in the user interface. When **AdvancedParametersVisible** is set to False, the Locator is in *Basic* setup mode, and the Advanced Parameters are hidden from view.

Syntax

control.**AdvancedParametersVisible** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Advanced Parameters are displayed. All Locator parameters are available in the user interface.
False	Advanced Parameters are not displayed in the user interface and can only be modified programmatically.

Remark(s)

Disabling **AdvancedParametersVisible** affects the user interface only. All Locator properties and methods are available programmatically even if they are hidden from the user interface. The following are hidden from the interface in Basic setup mode: All model-based related parameters, the Location, Transform, Contour tabs, Model Optimization and Model Disambiguation parameters, Results Viewer options. The only available Search parameters and constraints are: Scale, Rotation, Number of Instances to Find and Minimum Model Percentage.

Example

Locator.**AdvancedParametersVisible** = True

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = Locator1.CalibratedUnitsEnabled
Locator2.CalibratedUnitsEnabled = False
```

ConformityTolerance

Description

Maximum local deviation between the expected model contours of an instance and the contours actually detected in the input image. It corresponds to the maximum distance in calibrated units by which a matched contour can deviate from either side of its expected position in the model. Expressed in the calibrated units specified by the **UnitsLength** property. This property can only be set when **UseDefaultConformityTolerance** is set to false. Read only otherwise.

Syntax

control.**ConformityTolerance** [= Value As Single]

Type

Single

Range

Minimum: **MinimumConformityTolerance**

Maximum: **MaskImageFileName**

Remark(s)

This property can be set to any positive value if **ConformityToleranceRangeEnabled** is set to False.

Example

```
Dim ConformityTolerance As Single
ConformityTolerance = Locator1.ConformityTolerance
Locator2.ConformityTolerance = 2.0
```

ConformityToleranceRangeEnabled

Description

When **ConformityToleranceRangeEnabled** is set to True, the allowable range of values for **ConformityTolerance** is set by the read only **MinimumConformityTolerance** and **MaskImageFileName** properties. When set to False, **ConformityTolerance** can be set to any positive value.

Syntax

control.**ConformityToleranceRangeEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	ConformityToleranceRange is enabled.
False	ConformityToleranceRange is disabled.

Remark(s)

Disabling the conformity tolerance range can be useful for finding deformable objects that requires a high conformity tolerance value for a better match.

Example

```
Dim Result As Boolean
Result = Locator1.ConformityToleranceRangeEnabled
Locator2.ConformityToleranceRangeEnabled = False
```

ContrastPolarity

Description

Selects the type of polarity accepted for object recognition. Contrast polarity identifies the direction of change in greylevel values between an object and its surrounding area. Polarity is always defined with respect to the initial polarity of the Model (in the Model image).

Syntax

control.**ContrastPolarity** [= *Value* As hsContrastPolarity]

Type

hsContrastPolarity

Range

The valid range for this property is as follows:

hsContrastPolarity	Value	Description
hsContrastPolarityNormal	0	The Locator accepts only instances having the same polarity as that of the model and does not recognize local changes in polarity.
hsContrastPolarityReverse	1	The Locator accepts only instances having the inverse polarity as that of the model and does not recognize local changes in polarity.
hsContrastPolarityNormalAndReverse	2	The Locator accepts only instances having a polarity that is either the same or the inverse of the model's polarity but does not recognize local changes in polarity.
hsContrastPolarityDontCare	3	Accepts any polarity for the object, <i>INCLUDING</i> local changes in polarity.

Remark(s)

HexSight supports the **RespectContrastPolarity** property for compatibility with applications built in previous versions. If **RespectContrastPolarity** is True, the polarity is set to **hsContrastPolarityNormal**, if False, polarity is set to **hsContrastPolarityDontCare**.

Example

```
Dim CurrentMode As hsContrastPolarityMode
CurrentMode = Locator1.ContrastPolarityMode
Locator2.ParametersBasedOn = hsContourParametersCustom
Locator2.ContrastPolarityMode = hsContrastPolarityModeFixedValue
Locator2.ContrastPolarity = 10
```

ContrastThreshold

Description

Defines the minimum contrast needed for a contour to be detected in the input image. This threshold is expressed in terms of step in greylevel values. This property can only be set when **ContrastThresholdMode** is set to hsContrastThresholdFixedValue. Read only otherwise.

Syntax

control.**ContrastThreshold** [= Value As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 255

Example

```
Dim CurrentThreshold As Integer
CurrentThreshold = Locator1.ContrastThreshold
Locator2.ContrastThresholdMode = _
    hsContrastThresholdModeFixedValue
Locator2.ContrastThreshold = 10
```

ContrastThresholdMode

Description

Selects the method used to compute the threshold used for detecting contours in the input image.

Syntax

control.**ContrastThresholdMode** [= Value As hsContrastThresholdMode]

Type

hsContrastThresholdMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContrastThresholdAdaptiveLowSensitivity	0	Uses a low sensitivity adaptive threshold for detecting contours.

Constant	Value	Description
hsContrastThresholdAdaptiveNormalSensitivity	1	Uses a normal sensitivity adaptive threshold for detecting contours.
hsContrastThresholdAdaptiveHighSensitivity	2	Uses a high sensitivity adaptive threshold for detecting contours.
hsContrastThresholdFixedValue	3	Uses a fixed value threshold for detecting contours.
hsContrastThresholdModelsMode	4	Uses the most sensitive mode, selected from Models specified by the ParametersBasedOn selection.
hsContrastThresholdModelsValue	5	Uses the most sensitive value, selected from Models specified by the ParametersBasedOn selection.
hsContrastThresholdModelValuesAndMode	6	Uses either the most sensitive mode or the most sensitive value, selected from Models specified by the ParametersBasedOn selection.

Example

```
Dim CurrentMode As hsContrastThresholdMode
CurrentMode = Locator1.ContrastThresholdMode
Locator2.ContrastThresholdMode = _
    hsContrastThresholdAdaptiveNormalSensitivity
```

CoordinateSystem

Description

Coordinate system used to express the results.

Syntax

control.**CoordinateSystem** [= *Value* As hsLocatorCoordinateSystem]

Type

hsLocatorCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsLocatorImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsLocatorWorld	1	Results are expressed in calibrated units with respect to the World coordinate system.
hsLocatorTool	3	Results are expressed in pixels with respect to the Tool coordinate system. This coordinate system is positioned at the center of the Search Area of the Locator.

Example

```
Dim CoordSyst As hsLocatorCoordinateSystem
CoordSyst = Locator1.CoordinateSystem
```

Locator1.**CoordinateSystem** = hsLocatorTool

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

DefaultConformityTolerance

Description

Default value for **ConformityTolerance** computed by the Locator by analyzing the calibration, the contour detection parameters, and the search parameters. Read only.

Syntax

control.**DefaultConformityTolerance**

Type

Single

Range

Greater than zero.

Remark(s)

This default value is used for **ConformityTolerance** when **UseDefaultConformityTolerance** is set to True.

Example

```
Dim Default As Single
Default = Locator1.DefaultConformityTolerance
```

Description

Description

String that describes the standard usage of the Locator. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Locator1.Description
```

DetailLevel

Description

The coarseness of the contours at the Detail level. This property can only be set when **ParametersBasedOn** is set to **hsParametersCustom**. Read only otherwise.

Syntax

control.**DetailLevel** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1

Maximum: 16

Remark(s)

For most applications, the **ParametersBasedOn** property should be set to **hsParametersAllModels**. Custom contour detection should only be used when the default values do not work correctly.

Example

```
Locator2.DetailLevel = 1
```

```
Locator2.Execute
```

DynamicUpdate

Description

When **DynamicUpdate** is enabled, the instances are updated dynamically in the Results Viewer as soon as they are found during the search process.

Syntax

control.**DynamicUpdate** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Displaying of object instances is updated dynamically in the results Viewer.
False	Object instance are displayed in Results View after completion of Search process.

Example

```
Locator1.ShowResultsViewer
```

```
Locator1.DynamicUpdate = True
```

```
Locator1.Execute
```

ElapsedTime

Description

Total time elapsed (in milliseconds) during the last execution of the Locator tool. This time includes the time for the learn process, the time for the search process and the overhead required to create and output the results structures. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the total elapsed time, not the CPU time used.

Example

```
Dim TotalTime As Single
Dim LearnTime As Single
Dim SearchTime As Single
Locator1.Execute
TotalTime = Locator1.ElapsedTime
LearnTime = Locator1.LearnTime
SearchTime = Locator1.SearchTime
```

ErrorObject

Description

Returns a reference to the internal **hsErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

hsErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

FirstModelName

Description

Name of the first model in the models database. Returns an empty string when the database is empty. Accessing this property resets the internal model name iterator to the first model in the database. This internal iterator can be used to browse the model name with the following properties: **FirstModelName**, **LastModelName**, **NextModelName** and **PreviousModelName**. Read only.

Syntax

control.**FirstModelName**

Type

String

Range

Not applicable.

Example

```
Dim ModelName As String
```

ModelName = Locator1.**FirstModelName**

InitialDirectory

Description

Name of the file directory used for any file selection operations. If **InitialDirectory** property value is not set, initial directory for file selection operations will be based on default operating system functionality.

Syntax

control.**InitialDirectory**

Type

String

Range

Not applicable.

Example

```
Dim Directory As String
Locator1.InitialDirectory = "c:/application/data"
```

InputImage

InputGreyScaleImage

Description

Name of the image used as the input to the search process.

Syntax

control.**InputImage** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim InputImage As String
InputImage = Locator1.InputImage
Locator2.InputImage = "Image"
```

InputView

InputGreyScaleImageView

Description

Name of the View containing the input image set by the **InputImage** property.

Syntax

control.**InputView** [= *Value* As String]

Type

String

Range

Not applicable.

Remark(s)

See also **InputImage**.

Example

```
Dim InputImageView As String
InputImageView = Locator1.InputView
Locator2.InputView = "Acquisition View"
```

InputInstanceScene

Description

Name of the Scene — created by another instance of the **Locator** tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = Locator2.InputInstanceScene
Locator3.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Locator1.InputInstanceView
Locator2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = Locator1.InputModelScene
Locator2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View that provides the Model Scene. See **InputModelScene** for more details.

Syntax

control.**InputModelView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = Locator1.InputModelView
Locator2.InputModelView = "view0"
```

InstanceClearQuality

Description

Measure of the unencumbered area surrounding the specified object instance. Clear quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that the instance is completely free of obstacles. Clear Quality is only calculated if **MinimumClearPercentage** is enabled, otherwise the InstanceClearQuality is always returns a value of 1.0. Read only.

Syntax

control.**InstanceClearQuality**(*InstanceIndex* As Integer)

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim ClearQuality As Single
```

```
ClearQuality = Locator1.InstanceClearQuality(1)
```

InstanceCount

Description

Number of object instances found by the Locator tool. Read only.

Syntax

```
control.InstanceCount
```

Type

Integer

Range

Greater than or equal to zero.

Example

```
Dim InstanceCount As Integer
InstanceCount = Locator1.InstanceCount
```

InstanceFitQuality

Description

Normalized average error between the matched model contours of the selected object instance and the actual contours detected in the input image. Fit quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that the average error is 0. Conversely, a value of 0 means that the average matched error is equal to conformity tolerance. Read only.

Syntax

```
control.InstanceFitQuality(InstanceIndex As Integer)
```

Type

Single

Range

Minimum: 0.0

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim InstanceFitQuality As Single
InstanceFitQuality = Locator1.InstanceFitQuality(1)
```

InstanceMatchQuality

Description

Amount of matched model contours for the selected object instance. Match quality ranges from 0 to 1, with 1 being the best quality. A value of 1 means that 100% of the model contours were successfully matched to the actual contours detected in the input image. Read only.

Syntax

```
control.InstanceMatchQuality(InstanceIndex As Integer)
```


Type

Single

Range

Minimum: greater than zero.

Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim InstanceMatchQuality As Single
```

```
InstanceMatchQuality = Locator1.InstanceMatchQuality(1)
```

InstanceModel

Description

Index of the model associated to the selected object instance. Read only.

Syntax

```
control.InstanceModel(InstanceIndex As Integer)
```

Type

Integer

Range

Minimum: 0

Maximum: Number of Models - 1

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim InstanceModel As Integer
```

```
InstanceModel = Locator1.InstanceModel(1)
```

InstanceModelName

Description

Name of the model associated to the selected object instance. Read only.

Syntax

```
control.InstanceModelName(InstanceIndex As Integer)
```

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim ModelName As String
ModelName = Locator1.InstanceModelName(1)
```

InstanceOrdering

Description

Order in which the instances are processed and output.

Syntax

control.**InstanceOrdering** [= *Value* As hsInstanceOrdering]

Type

hsInstanceOrdering

Range

The valid range for this property is as follows:

Constant	Value	Description
<i>hsEvidence</i>	0	Instances are processed and output according to their hypothesis strength, beginning with the strongest hypothesis.
<i>hsLeftToRight</i>	1	Instances are processed and output in the order they appear in the search area, from left to right.
<i>hsRightToLeft</i>	2	Instances are processed and output in the order they appear in the search area, from right to left.
<i>hsTopToBottom</i>	3	Instances are processed and output in the order they appear in the search area, from top to bottom.
<i>hsBottomToTop</i>	4	Instances are processed and output in the order they appear in the search area, from bottom to top.
<i>hsQuality</i>	5	All the instances are first processed and then they are output according to their Quality, beginning with the highest quality.
<i>hsDistanceImage</i>	6	Instances are processed and output according to their distance from a reference image coordinate, beginning with the closest.
<i>hsDistanceWorld</i>	7	Instances are processed and output according to their distance from a reference world coordinate, beginning with the closest.
<i>hsShadingConsistency</i>	8	Instances are processed and output according to their shading consistency with respect to the Model, beginning with the strongest hypothesis.
<i>hsLeftToRightPrecise</i>	9	Instances are processed and output in the order they appear in the search area, from left to right, in an exact manner.

Constant	Value	Description
<i>hsRightToLeftPrecise</i>	10	Instances are processed and output in the order they appear in the search area, from right to left, in an exact manner.
<i>hsTopToBottomPrecise</i>	11	Instances are processed and output in the order they appear in the search area, from top to bottom, in an exact manner.
<i>hsBottomToTopPrecise</i>	12	Instances are processed and output in the order they appear in the search area, from bottom to top, in an exact manner.
<i>hsDistanceImagePrecise</i>	13	Instances are processed and output according to their distance from a reference image coordinate, beginning with the closest, in an exact manner.
<i>hsDistanceWorldPrecise</i>	14	Instances are processed and output according to their distance from a reference world coordinate, beginning with the closest, in an exact manner.

Remark(s)

With the hsQuality 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. With the hsDistanceImage and hsDistanceWorld modes, the reference coordinate used to compute the distance is set with the **InstanceOrderingReferenceX** and **InstanceOrderingReferenceY** properties. While the directional ordering are done on a best-effort basis, to obtain the first instance immediately when available, the Precise modes are ordering the instances in an exact manner, when all the instances are located. In those latter modes, just as with the hsQuality mode, the return of located instances as they are found is not available.

Example

```
Dim CurrentOrder = Locator1.InstanceOrdering
Locator1.InstanceOrdering = hsLeftToRight
```

InstanceOrderingReferenceX

Description

Reference X coordinate used to compute the distance when either of the hsDistanceImage or hsDistanceWorld ordering modes are enabled using the **InstanceOrdering** property.

Syntax

```
control.InstanceOrderingReferenceX [= Value As Single]
```

Type

Single

Range

Not applicable.

Example

```
' Locate the instance closest to pixel coordinate
' X=150, Y=325
Locator1.InstanceOrdering = hsDistanceImage
```

```
Locator1.InstanceOrderingReferenceX = 150
Locator1.InstanceOrderingReferenceY = 325
Locator1.Execute
```

InstanceOrderingReferenceY

Description

Reference Y coordinate used to compute the distance when either of the `hsDistanceImage` or `hsDistanceWorld` ordering modes are enabled using the **InstanceOrdering** property.

Syntax

```
control.InstanceOrderingReferenceX [= Value As Single]
```

Type

Single

Range

Not applicable.

Example

```
' Locate the instance closest to pixel coordinate
' X=150, Y=325
Locator1.InstanceOrdering = hsDistanceImage
Locator1.InstanceOrderingReferenceX = 150
Locator1.InstanceOrderingReferenceY = 325
Locator1.Execute
```

InstanceReferencePointCount

Description

Number of reference points of the selected object instance. Read only.

Syntax

```
control.InstanceReferencePointCount(InstanceIndex As Integer)
```

Type

Integer

Range

Greater than or equal to zero.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim ReferencePointCount As Integer
ReferencePointCount = Locator1.InstanceReferencePointCount(0)
```

InstanceReferencePointPositionX

Description

X coordinate of the selected reference point of the selected object instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the calibrated units

specified by the **UnitsLength** property when the **CoordinateSystem** property is set to **hsLocatorWorld**. Expressed in pixels otherwise. Read only.

Syntax

control.InstanceReferencePointPositionX(*InstanceIndex* As Integer, *ReferencePointIndex* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1
<i>ReferencePointIndex</i>	Index of the reference point.	Integer	Minimum: 0 Maximum: InstanceReferencePointCount - 1

Example

```
Dim RefPtPositionX As Single
RefPtPositionX = Locator1.InstanceReferencePointPositionX(1, 0)
```

InstanceReferencePointPositionY

Description

Y coordinate of the selected reference point of the selected object instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the calibrated units specified by the **UnitsLength** property when the **CoordinateSystem** property is set to **hsLocatorWorld**. Expressed in pixels otherwise. Read only.

Syntax

control.InstanceReferencePointPositionY(*InstanceIndex* As Integer, *ReferencePointIndex* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1
<i>ReferencePointIndex</i>	Index of the reference point.	Integer	Minimum: 0 Maximum: InstanceReferencePointCount - 1

Example

```
Dim RefPtPositionY As Single
```

```
RefPtPositionY = Locator1.InstanceReferencePointPositionY(1, 0)
```

InstanceRotation

Description

Angle of rotation of the Object coordinate system of the selected object instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the orientation units specified by the **UnitsOrientation** property when the **CoordinateSystem** property is set to **hsLocatorWorld**. Unavailable otherwise. Read only.

Syntax

```
control.InstanceRotation(InstanceIndex As Integer)
```

Type

Single

Range

Minimum: **MinimumRotation** or **NominalRotation**

Maximum: **MaximumRotation** or **NominalRotation**

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Remark(s)

When the **NominalRotationEnabled** property is True, the rotation of the object instance is always equal to **NominalRotation**.

Example

```
Dim InstanceRotation
InstanceRotation = Locator1.InstanceRotation(1)
```

InstanceScaleFactor

Description

Scale factor of the selected object instance, giving its relative size with respect to its associated model. Available only when the **CoordinateSystem** property is set to **hsLocatorWorld**. Unavailable otherwise. Read only.

Syntax

```
control.InstanceScaleFactor(InstanceIndex As Integer)
```

Type

Single

Range

Minimum: **MinimumScaleFactor** or **NominalScaleFactor**

Maximum: **MaximumScaleFactor** or **NominalScaleFactor**

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Remark(s)

When the **NominalScaleFactorEnabled** property is True, the scale factor of the object instance is always equal to **NominalScaleFactor**.

Example

```
Dim ScaleFactor As Single
ScaleFactor = Locator1.InstanceScaleFactor(1)
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

control.**InstanceSelectionEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.InstanceSelectionEnabled
Locator2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.**InstanceSelectionIndex** [= Value As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
```

```
Ndx = Locator1.InstanceSelectionIndex
Locator2.InstanceSelectionIndex = 0
```

InstanceSymmetry

Description

Index of the object instance of which the selected object instance is a symmetry of. Read only.

Syntax

```
control.InstanceSymmetry(InstanceIndex As Integer)
```

Type

Integer

Range

Minimum: 0

Maximum: **InstanceCount** - 1

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Remark(s)

If **OutputSymmetricInstances** is set to False, **InstanceSymmetry** is always equal to the instance's index.

Example

```
Dim ObjectIsSymmetric As Boolean
Dim Index As Integer
Index = 2
If Locator1.InstanceSymmetry(Index) = Index Then
    ObjectIsSymmetric = False
Else
    ObjectIsSymmetric = True
End If
```

InstanceTime

Description

Time needed to recognize and locate the selected object instance, expressed in milliseconds.
Read only.

Syntax

```
control.InstanceTime(InstanceIndex As Integer)
```

Type

Single

Range

Greater than 0

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Remark(s)

The time needed to locate the first object instance is usually longer because it includes all of the low-level image preprocessing.

Example

```
Dim T As Single
T = Locator1.InstanceTime(3)
```

InstanceTranslationX

Description

X translation of the Object coordinate system of the selected object instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the calibrated units specified by the **UnitsLength** property when the **CoordinateSystem** property is set to **hsLocatorWorld**. Expressed in pixels otherwise. Read only.

Syntax

control.**InstanceTranslationX**(*InstanceIndex* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim TranslationX As Single
TranslationX = Locator1.InstanceTranslationX(0)
```

InstanceTranslationY

Description

Y translation of the Object coordinate system of the selected object instance, with respect to the coordinate system set by the **CoordinateSystem** property. Expressed in the calibrated units specified by the **UnitsLength** property when the **CoordinateSystem** property is set to **hsLocatorWorld**. Expressed in pixels otherwise. Read only.

Syntax

control.**InstanceTranslationY**(*InstanceIndex* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim TranslationY As Single
TranslationY = Locator1.InstanceTranslationY(0)
```

InstanceVisible

Description

Percentage of the selected object instance that fits within the boundary of the selected input image. Instance visible ranges from 0 to 1, with 1 being the indication of an object instance been totally enclosed within boundary of input image. This result value can be used to detect if an object instance touch image borders. In that case, returned value will be below 1.

Read only.

Syntax

control.**InstanceVisible**(*InstanceIndex* As Integer)

Type

Single

Range

Minimum: 0.0
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1

Example

```
Dim InstanceVisible As Single
InstanceVisible = Locator1.InstanceVisible(1)
```

LastModelName

Description

Name of the last model in the models database. Returns an empty string when the database is empty. Accessing this property resets the internal model name iterator to the last model in the database. This internal iterator can be used to browse the model name with the following properties: **FirstModelName**, **LastModelName**, **NextModelName** and **PreviousModelName**. Read only.

Syntax

control.**LastModelName**

Type

String

Range

Not applicable.

Example

```
Dim modelName As String
modelName = Locator1.LastModelName
```

LearnTime

Description

Time elapsed (in milliseconds) for the learn process during the last execution of the Locator tool. If the Learn process was not required, the LearnTime is time is 0. A Learn process is required and automatically called on the next execution of the Locator after modifications to models and/or changes in some search parameters. Read only.

Syntax

control.**LearnTime**

Type

Single

Range

Greater than 0

Example

```
Dim totalTime As Single
Dim learnTime As Single
Dim searchTime As Single
Locator1.Execute
totalTime = Locator1.ElapsedTime
learnTime = Locator1.LearnTime
searchTime = Locator1.SearchTime
```

MaskImageFileName

Description

Name of the mask image file currently loaded. If the image file name is changed using the **MaskImageFileName** property directly, the Locator will not load this image file. To load a new mask image file, the **LoadMaskImage** method should be used instead.

Syntax

control.**MaskImageFileName**

Type

String

Range

Not applicable.

Example

```
Dim fileName As String
fileName = Locator1.MaskImageFileName
```

MaximumConformityTolerance

Description

Maximum value allowed for the **ConformityTolerance** property. Read only.

Syntax

control.**MaximumConformityTolerance**

Type

Single

Range

Greater than **MinimumConformityTolerance**.

Remark(s)

This property is computed by the Locator by analyzing the calibration, the contour detection parameters, and the search parameters. See also **ConformityTolerance**, **DefaultConformityTolerance**, and **ConformityToleranceRangeEnabled**.

Example

```
MaxConformityTolerance As Single
MaxConformityTolerance = Locator1.MaximumConformityTolerance
```

MaximumInstanceCount

Description

Maximum number of object instances that are searched for in the input image. All of the object instances respecting the search constraints are output, up to a maximum of **MaximumInstanceCount**. They are ordered according to the **InstanceOrdering** property.

Syntax

control.**MaximumInstanceCount** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1
Maximum: 2000

Remark(s)

This property is applicable only if the **MaximumInstanceCountEnabled** property is set to True.

Example

```
Dim MaxInstances As Integer
MaxInstances = Locator1.MaximumInstanceCount
Locator2.MaximumInstanceCount = 10
```

MaximumInstanceCountEnabled

Description

When **MaximumInstanceCountEnabled** is True, the search is limited to the number of instances set by the **MaximumInstanceCount** property.

Syntax

control.**MaximumInstanceCountEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Search is limited to number of instances specified by MaximumInstanceCount .
False	Search is not limited to a set number of instances.

Example

```
Dim MaxInstanceCountEnabled As Boolean
MaxInstanceCountEnabled = Locator1.MaximumInstanceCountEnabled
Locator2.MaximumInstanceCountEnabled = False
```

MaximumRotation

Description

Maximum angle of rotation allowed for an object instance to be recognized. Expressed in the orientation units specified by the **UnitsOrientation** property.

Syntax

control.**MaximumRotation** [= Value As Single]

Type

Single

Range

Minimum: -180.0 degrees or -PI rad

Maximum: 180.0 degrees or PI rad

Remark(s)

This property is applicable only if the **NominalRotationEnabled** property is set to False.

When **MaximumRotation** is lower than **MinimumRotation**, the search range is equivalent to **MinimumRotation** to **MaximumRotation** + 360 degrees (or 2*PI rad).

Example

```
Dim MaxRotation As Single
MaxRotation = Locator1.MaximumRotation
Locator2.MaximumRotation = 180.0
```

MaximumScaleFactor

Description

Maximum scale factor allowed for an object instance to be recognized.

Syntax

control.**MaximumScaleFactor** [= Value As Single]

Type

Single

Range

Minimum: 0.1

Maximum: 10.0

Remark(s)

This property is applicable only if the **NominalScaleFactorEnabled** property is set to False.

Example

```
Dim MaximumScaleFactor As Single
MaximumScaleFactor = Locator1.MaximumScaleFactor
Locator2.MaximumScaleFactor = 10.0
```

MessageCount

Description

The number of messages issued during the last execution. The identification number and description of each message is retrieved using respectively the **MessageNumber** and **MessageDescription** properties. Read only.

Syntax

control.**MessageCount**

Type

Integer

Range

Minimum: 0

Maximum: Unlimited

Example

```
Dim m As Integer
For m = 0 To Locator1.MessageCount - 1
    Description.Text = Locator1.MessageDescription(m)
    Number.Text = Str(Locator1.MessageNumber(m))
Next m
```

MessageDescription

Description

The description of a message issued during the last execution. The identification number of the message corresponding to this description is retrieved using the **MessageNumber** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageDescription**(*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```
Dim m As Integer
For m = 0 To Locator1.MessageCount - 1
    Description.Text = Locator1.MessageDescription(m)
```

```

        Number.Text = Str(Locator1.MessageNumber(m))
    Next m

```

MessageNumber

Description

The identification number of a message issued during the last execution. The description of the message corresponding to this identification number is retrieved using the **MessageDescription** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageNumber**(*Index* As Integer)

Type

Integer

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```

Dim m As Integer
For m = 0 To Locator1.MessageCount - 1
    Description.Text = Locator1.MessageDescription(m)
    Number.Text = Str(Locator1.MessageNumber(m))
Next m

```

MinimumClearPercentage

Description

When **MinimumClearPercentageEnabled** is set to true, **MinimumClearPercentage** sets the minimum percentage of the model bounding box area that must be free of obstacles to consider an object instance as valid.

Syntax

control.**MinimumClearPercentage** [= *Value* As Single]

Type

Single

Range

Minimum: Greater than zero.
Maximum: 100.0

Example

```

Dim MinimumClear As Single
MinimumClear = Locator1.MinimumClearPercentage
Locator2.MinimumClearPercentage = 20.0

```

MinimumClearPercentageEnabled

Description

When **MinimumClearPercentageEnabled** is set to true, the **MinimumClearPercentage** constraint is applied to the search process. **MinimumClearPercentageEnabled** must be enable to calculate the **InstanceClearQuality**.

Syntax

control.**MinimumClearPercentageEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The MinimumClearPercentage constraint is enabled and applied to the Search process.
False	The MinimumClearPercentage constraint is not enabled.

Example

```
Dim Clear As Boolean
Clear = Locator.MinimumClearPercentageEnabled
```

MinimumConformityTolerance

Description

Minimum value allowed for the **ConformityTolerance** property. Read only.

Syntax

control.**MinimumConformityTolerance**

Type

Single

Range

Minimum: greater than zero.
Maximum: Lower than **MaskImageFileName**.

Remark(s)

This property is computed by the Locator by analyzing the calibration, the contour detection parameters, and the search parameters. See also **ConformityTolerance**, **DefaultConformityTolerance**, and **ConformityToleranceRangeEnabled**.

Example

```
Dim MinConformityTolerance As Single
MinConformityTolerance = Locator1.MinimumConformityTolerance
```

MinimumModelPercentage

Description

Minimum percentage of model contours that need to be matched in the input image in order to consider the object instance as valid.

Syntax

control.**MinimumModelPercentage** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: 100.0

Example

Dim MinModelPercent As Single

MinModelPercent = Locator1.**MinimumModelPercentage**Locator2.**MinimumModelPercentage** = 50.0

MinimumRequiredFeatures

Description

Minimum percentage of required features that must be recognized in order to consider the object instance as valid.

Syntax*control*.**MinimumRequiredFeatures** [= *Value* As Single]**Type**

Single

Range

Minimum: greater than zero.

Maximum: 100.0

Remark(s)

The minimum percentage of required features is expressed in terms of the number of required features in a model without considering the amount of contour each required feature represents in the model. For example, if the model contains 3 required features and **MinimumRequiredFeatures** is set to 50%, an instance of the object will be considered valid as long as 2 out of 3 required features are recognized.

Example

Dim MinRequiredFeatures As Single

MinRequiredFeatures = Locator1.**MinimumRequiredFeatures**Locator2.**MinimumRequiredFeatures** = 100.0

MinimumRotation

Description

Minimum angle of rotation allowed for an object instance to be recognized. Expressed in the orientation units specified by the **UnitsOrientation** property.

Syntax*control*.**MinimumRotation** [= *Value* As Single]**Type**

Single

Range

Minimum: -180.0 degrees or -PI rad

Maximum: 180.0 degrees or PI rad

Remark(s)

This property is applicable only if the **NominalRotationEnabled** property is set to False.
 When **MaximumRotation** is lower than **MinimumRotation**, the search range is equivalent to **MinimumRotation** to **MaximumRotation** + 360 degrees (or 2*PI rad).

Example

```
Dim MinRotation As Single
MinRotation = Locator1.MinimumRotation
Locator2.MinimumRotation = -180.0
```

MinimumScaleFactor

Description

Minimum scale factor allowed for an object instance to be recognized.

Syntax

control.**MinimumScaleFactor** [= *Value* As Single]

Type

Single

Range

Minimum: 0.1
 Maximum: 10.0

Remark(s)

This property is applicable only if the **NominalScaleFactorEnabled** property is set to False.

Example

```
Dim MinimumScaleFactor As Single
MinimumScaleFactor = Locator1.MinimumScaleFactor
Locator2.MinimumScaleFactor = 0.5
```

ModelAutomaticLevels

Description

When set to True, this property specifies that the Outline and Detail coarseness levels were automatically optimized while building the model. When False, the Outline and Detail coarseness levels were configured manually before building the model. Read only.

Syntax

control.**ModelAutomaticLevels** (*Model* As Variant)

Type

Boolean

Range

Value	Description
True	Detail and coarseness levels were automatically optimized while building the model.
False	Detail coarseness levels were manually configured before building the model.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Boolean
Value = LocatorTool.ModelAutomaticLevels("hook")
```

ModelBasedMaximumRotation

Description

Maximum angle of rotation allowed for the **TransformRotation** property when the **ModelTransformEnabled** is set to True (model-based mode is active) and the **ModelBasedRotationMode** is set to **hsRelative**. Expressed in the orientation units specified by the **UnitsOrientation** property.

Syntax

```
control.ModelBasedMaximumRotation [= Value As Single]
```

Type

Single

Range

Minimum: -180.0 degrees or -PI rad
Maximum: 180.0 degrees or PI rad

Remark(s)

See **ModelBasedRotationMode** property for more details.

Example

```
Dim AllowedRotation As Single
AllowedRotation = Locator1.ModelBasedMaximumRotation
Locator2.ModelBasedMaximumRotation = 180.0
```

ModelBasedMaximumScaleFactor

Description

Maximum scale allowed for the **TransformScale** property when the **ModelTransformEnabled** is set to True (model-based mode is active) and the **ModelBasedScaleFactorMode** is set to **hsRelative**.

Syntax

```
control.ModelBasedMaximumScaleFactor [= Value As Single]
```

Type

Single

Range

Minimum: 0.1
Maximum: 10.0

Remark(s)

See **ModelBasedScaleFactorMode** property for more details.

Example

```
Dim AllowedScale As Single
AllowedScale = Locator1.ModelBasedMaximumScaleFactor
Locator2.ModelBasedMaximumScaleFactor = 1.25
```

ModelBasedMinimumRotation

Description

Minimum angle of rotation allowed for the **TransformRotation** parameter when the **ModelTransformEnabled** property is set to True (model-based mode is active) and the **ModelBasedRotationMode** is set to **hsRelative**. Expressed in the orientation units specified by the **UnitsOrientation** property.

Syntax

control.**ModelBasedMinimumRotation** [= *Value* As Single]

Type

Single

Range

Minimum: -180.0 degrees or -PI rad
Maximum: 180.0 degrees or PI rad

Remark(s)

See **ModelBasedRotationMode** property for more details.

Example

```
Dim AllowedRotation As Single
AllowedRotation = Locator1.ModelBasedMinimumRotation
Locator2.ModelBasedMinimumRotation = -180.0
```

ModelBasedMinimumScaleFactor

Description

Minimum scale allowed for the **TransformScale** property when the **ModelTransformEnabled** is set to True (model-based mode is active) and the **ModelBasedScaleFactorMode** is set to **hsRelative**.

Syntax

control.**ModelBasedMinimumScaleFactor** [= *Value* As Single]

Type

Single

Range

Minimum: 0.1
Maximum: 10.0

Remark(s)

See **ModelBasedScaleFactorMode** property for more details.

Example

```
Dim AllowedScale As Single
AllowedScale = Locator1.ModelBasedMinimumScaleFactor
Locator2.ModelBasedMinimumScaleFactor = 0.75
```

ModelBasedRotationMode

Description

Selects the method used to manage the rotation parameters of the Locator's search (**MinimumRotation**, **MaximumRotation** and **NominalRotation**), when the **ModelTransformEnabled** property is set to True.

Syntax

control.**ModelBasedRotationMode**(= *Value* As hsModelBasedMode)

Type

hsModelBasedMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsAbsolute	0	Default Value. The Search directly applies the rotation parameters specified for the Locator search
hsRelative	1	The specified TransformRotation value modifies the rotation parameters of the Search process. For example, if the nominal Search is set to 45 degrees and the TransformRotation value is set to 10 degrees, the Locator will search for instances having a rotation of 55 degrees.

Remark(s)

hsAbsolute has no effect on rotation parameters of the Locator's search process. It is useful for positioning objects, based on the position of an object found by an initial Locator tool.
hsRelative optimizes search speed and robustness when you need to accurately position sub-parts of an object, based on the position of the source object. In the hsRelative mode, the Locator's Learn phase applies **ModelBasedMinimumRotation** and **ModelBasedMaximumRotation** as the allowed rotation range for the **TransformRotation** property.

Example

```
Locator.ModelBasedMaximumRotation = 10.00
Locator.ModelBasedMinimumRotation = -10.00
Mode = LocatorTool.ModelBasedRotationMode = hsRelative
Locator.NominalRotationEnabled = True
Locator.Nominal Rotation = 0
```

' Search for subparts on instances from 10 to -10 degrees

ModelBasedScaleFactorMode

Description

Selects the method used to manage the scale properties used by the Locator's search (**MinimumScaleFactor**, **MaximumScaleFactor** and **NominalScaleFactor**), when the **ModelTransformEnabled** property is set to True.

Syntax

control.**ModelBasedScaleFactorMode**(= *Value* As hsModelBasedMode)

Type

hsModelBasedMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsAbsolute	0	Default Value. The Search directly applies the scale parameters specified for the Locator search.
hsRelative	1	The specified TransformScale value modifies the scale parameters of the Search. For example, if the nominal scale value is 1.25 and the TransformScale value is set to 1.5, the Locator will search for instances with a scale factor of 1.5.

Remark(s)

hsAbsolute has no effect on the scale parameters of the Locator's search process. It is useful for positioning objects, based on the position of an object found by an initial Locator tool.

hsRelative optimizes search speed and robustness when you need to accurately position sub-parts of an object, based on the position of the source object. In the **hsRelative** mode, the Locator's Learn phase applies **ModelBasedMinimumScaleFactor** and **ModelBasedMinimumRotation** as the possible range in scale for the **TransformScale** property.

Example

```
Locator.ModelBasedMaximumScaleFactor = 2.0
Locator.ModelBasedMinimumScaleFactor = 0.5
Locator.ModelBasedScaleFactorMode = hsRelative
Locator.NominalScaleFactorEnabled = True
Locator.NominalScaleFactor = 1.0
```

```
' Search for subfeatures on instances
' varying in scale from .5 to 2
```

ModelBoundingAreaBottom

Description

The Y position in the image of the bottom of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used as an indication of the image area in which the contours used to construct the model were detected. **ModelBoundingAreaBottom** is expressed in pixels. Read only.

Syntax

control.**ModelBoundingAreaBottom** (*Model* As Variant)

Type

Integer

Range

Minimum: 0.

Maximum: height of the image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Integer
Value = LocatorTool.ModelBoundingAreaBottom(0)
```

ModelBoundingAreaLeft**Description**

The X position in the image of the left side of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used as an indication of the image area in which the contours used to construct the model were detected. **ModelBoundingAreaLeft** is expressed in pixels. Read only.

Syntax

```
control.ModelBoundingAreaLeft(Model As Variant)
```

Type

Integer

Range

Minimum: 0.
Maximum: width of the image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Integer
Value = LocatorTool.ModelBoundingAreaLeft(0)
```

ModelBoundingAreaRight**Description**

The X position in the image of the right side of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used as an indication of the image area in which the contours used to construct the model were detected. **ModelBoundingAreaRight** is expressed in pixels. Read only.

Syntax

control.**ModelBoundingAreaRight** (*Model* As Variant)

Type

Integer

Range

Minimum: 0.

Maximum: width of the image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Integer
```

```
Value = LocatorTool.ModelBoundingAreaRight(0)
```

ModelBoundingAreaTop

Description

The Y position in the image of the top of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used as an indication of the image area in which the contours used to construct the model were detected. **ModelBoundingAreaTop** is expressed in pixels. Read only.

Syntax

control.**ModelBoundingAreaTop** (*Model* As Variant)

Type

Integer

Range

Minimum: 0.

Maximum: height of the image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Integer
```

```
Value = LocatorTool.ModelBoundingAreaTop(0)
```


ModelContrastThreshold

Description

The contrast threshold value used to detect the contours from which the model features were selected. Read only.

Syntax

control.**ModelContrastThreshold** (*Model* As Variant)

Type

Integer

Range

Minimum: 1

Maximum: 255

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Integer
```

```
Value = LocatorTool.ModelContrastThreshold("hook")
```

ModelContrastThresholdMode

Description

The method used to compute the threshold used for detecting contours in the model image. Read only.

Syntax

control.**ModelContrastThresholdMode**(*Model* As Variant)

Type

hsContrastThresholdMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContrastThresholdAdaptiveLowSensitivity	0	Uses a low sensitivity adaptive threshold for detecting contours.
hsContrastThresholdAdaptiveNormalSensitivity	1	Uses a normal sensitivity adaptive threshold for detecting contours.
hsContrastThresholdAdaptiveHighSensitivity	2	Uses a high sensitivity adaptive threshold for detecting contours.
hsContrastThresholdFixedValue	3	Uses a fixed value threshold for detecting contours.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Mode As hsContrastThresholdMode
Mode = LocatorTool.ModelContrastThresholdMode("hook")
```

ModelCount

Description

Number of models in the models database. Read only.

Syntax

```
control.ModelCount
```

Type

Integer

Range

Greater than or equal to zero.

Example

```
Dim Number As Integer
Number = Locator1.ModelCount
```

ModelDatabase

Description

Handle to the current models database of the Locator. This handle can be used to share models databases between multiple instances of the Locator tool. Models database sharing can be used to reduce runtime memory usage in applications where the same models must be searched in different search areas or with different parameters. This property can only be set when **UseExternalModelDatabase** is set to True. Read only otherwise.

Syntax

```
control.ModelDatabase [= Value As OLE_HANDLE]
```

Type

OLE_HANDLE

Range

Not applicable.

Example

```
Locator2.UseExternalModelDatabase = True
Locator2.ModelDatabase = Locator1.ModelDatabase
Locator2.Execute
```

ModelDatabaseFileName

Description

Name of the models database file currently loaded. If the database file name is changed using the **ModelDatabaseFileName** property directly, the Locator will not load this database file. To load a new database file, the **LoadModelDatabase** method should be used instead.

Syntax

control.**ModelDatabaseFileName**

Type

String

Range

Not applicable.

Example

```
Dim FileName As String
FileName = Locator1.ModelDatabaseFileName
```

ModelDatabaseModified

Description

When True, indicates that the current models database has been modified. These modifications include edition of an existing model, addition of a new model to the database and deletion of an existing model from the database. Read only.

Syntax

control.**ModelDatabaseModified**

Type

Boolean

Range

Not applicable.

Example

```
If Locator1.ModelDatabaseModified Then
    Locator1.SaveModelDatabase "MyDatabaseFile"
End If
```

ModelDetailLevel

Description

The coarseness of the contours used to build the model at the Detail level. Read only.

Syntax

control.**ModelDetailLevel** (*Model* As Variant)

Type

Integer

Range

Minimum: 1
Maximum: 16

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As Integer
Value = LocatorTool.ModelDetailLevel("hook")
```

ModelDisambiguationEnabled

Description

When set to True (default), the Locator applies disambiguation to discriminate between similar models and between similar hypotheses of a single object. When set to False, the Locator does not apply disambiguation.

Syntax

```
control.ModelDisambiguationEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Locator applies disambiguation.
False	Locator does not apply disambiguation.

Example

```
Dim EnableDisambiguation As Boolean
EnableDisambiguation = Locator1.ModelDisambiguationEnabled
Locator2.ModelDisambiguationEnabled = False
```

ModelEditorActive

Description

Specifies if the Model Editor is currently active. **ModelEditorActive** can be used to check when model editing is completed. Read only.

Syntax

```
control.ModelEditorActive
```

Type

Boolean

Range

Not applicable.

Example

```
Do
    DoEvents
```

Loop While Locator1.**ModelEditorActive**

ModelEditorEnabled

Description

Specifies if the models can be edited interactively using the Model Editor. When set to True, the models are editable. When set to False, the models cannot be modified.

Syntax

control.**ModelEditorEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Models can be edited.
False	Models cannot be edited.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ModelEditorEnabled
Locator1.ModelEditorEnabled = False
```

ModelEnabled

Description

Specifies if the selected model is enabled, which means that the Locator will search for this model when the Locator is executed. Enabled models are output in the application runtime database, in the view set by the **OutputModelView** property.

Syntax

control.**ModelEnabled**(*Model* As Variant) [= *NewValue* As Boolean]

Type

Boolean

Range

Value	Description
True	Models is enabled.
False	Models is disabled.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ModelEnabled(0)
```

ModelFeatureSelection

Description

The mode used to select model features from the detected contours at both the Outline and Detail levels. Read only.

Syntax

```
control.ModelFeatureSelection(Model As Variant) [= Value as hsModelFeatureSelection]
```

Type

hsModelFeatureSelection

Range

The valid range for this property is as follows:

Constant	Value	Description
hsModelFeatureSelectionNone	0	No features were automatically selected from the contours.
hsModelFeatureSelectionLess	1	Fewer features than the optimal set were automatically selected from the contours.
hsModelFeatureSelectionNormal	2	The optimal features were automatically selected from the contours.
hsModelFeatureSelectionMore	3	More features than the optimal set were automatically selected from the contours.
hsModelFeatureSelectionAll	4	All of the contours were automatically selected as features.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim Value As hsModelFeatureSelection
Value = ModelFeatureSelection("hook")
```

ModelName

Description

Name of the selected model. Read only.

Syntax

```
control.ModelName(ModelIndex As Integer)
```

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>ModelIndex</i>	Index of the selected model.	Integer	Minimum: 0 Maximum: ModelCount - 1

Example

```
Dim Name As String
Name = Locator1.ModelName(0)
```

ModelOptimizerActive

Description

Specifies if the Model Optimizer is currently active. **ModelOptimizerActive** can be used to check when model optimizing is completed. Read only.

Syntax

control.**ModelOptimizerActive**

Type

Boolean

Range

Not applicable.

Example

```
Do
    DoEvents
Loop While Locator1.ModelOptimizerActive
```

ModelOptimizerEnabled

Description

Specifies if the models can be optimized interactively using the Model Optimizer. When set to True, the models can be optimized. When set to False, the models cannot be modified.

Syntax

control.**ModelOptimizerEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Models can be optimized.
False	Models cannot be optimized.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ModelOptimizerEnabled
Locator1.ModelOptimizerEnabled = False
```

ModelOriginPositionX

Description

The X position in the world coordinate system of the selected model's origin. The origin, defined by the **ModelOriginPositionX**, **ModelOriginPositionY**, and **ModelOriginRotation** properties, is used by the Locator to express the pose of instances of the model. The translation of the instance represents the position of the model's origin in the coordinate system selected by the **CoordinateSystem** property. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin also defines the object coordinate system that can be used to express results of model-based inspection tools. **ModelOriginPositionX** is expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**ModelOriginPositionX**(*Model* As Variant)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim X As Single
Dim Y As Single
Dim Rotation As Single
X = Locator1.ModelOriginPositionX(0)
Y = Locator1.ModelOriginPositionY(0)
Rotation = Locator1.ModelOriginRotation(0)
```

ModelOriginPositionY

Description

The Y position in the world coordinate system of the selected model's origin. The origin, defined by the **ModelOriginPositionX**, **ModelOriginPositionY**, and **ModelOriginRotation** properties, is used by the Locator to express the pose of instances of the model. The translation of the instance represents the position of the model's origin in the coordinate system selected by the **CoordinateSystem** property. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin also defines the object coordinate system that can be used to express results of model-based inspection tools. **ModelOriginPositionY** is expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**ModelOriginPositionY**(*Model* As Variant)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```

Dim X As Single
Dim Y As Single
Dim Rotation As Single
X = Locator1.ModelOriginPositionX(0)
Y = Locator1.ModelOriginPositionY(0)
Rotation = Locator1.ModelOriginRotation(0)

```

ModelOriginRotation

Description

The rotation in the world coordinate system of the selected model's origin. The origin, defined by the **ModelOriginPositionX**, **ModelOriginPositionY**, and **ModelOriginRotation** properties, is used by the Locator to express the pose of instances of the model. The translation of the instance represents the position of the model's origin in the coordinate system selected by the **CoordinateSystem** property. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin also defines the object coordinate system that can be used to express results of model-based inspection tools. **ModelOriginRotation** is expressed in the orientation units specified by the **UnitsOrientation** property of the associated Locator. Read only.

Syntax

control.**ModelOriginRotation**(*Model* As Variant)

Type

Single

Range

Minimum: -180.0
Maximum: 180.0

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```

Dim X As Single
Dim Y As Single
Dim Rotation As Single
X = Locator1.ModelOriginPositionX(0)
Y = Locator1.ModelOriginPositionY(0)
Rotation = Locator1.ModelOriginRotation(0)

```

ModelOutlineLevel

Description

The coarseness of the contours used to build the model at the Outline level. Read only.

Syntax

control.**ModelOutlineLevel** (*Model* As Variant)

Type

Integer

Range

Minimum: 1
Maximum: 16

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```

Dim Value As Integer
Value = LocatorTool.ModelOutlineLevel("hook")

```

ModelReferencePointCount

Description

Number of reference points defined on the selected model. Read only.

Syntax

control.**ModelReferencePointCount** (*Model* As Variant)

Type

Integer

Range

Greater than or equal to zero.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim NumPoints As Integer
NumPoints = Locator1.ModelReferencePointCount(0)
NumPoints = Locator1.ModelReferencePointCount("Cover")
```

ModelReferencePointPositionX**Description**

X coordinate of the selected reference point on the selected model. Expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**ModelReferencePointPositionX** (*Model* As Variant, *Index* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.
<i>Index</i>	Index of the selected reference point.	Integer	Minimum: 0 Maximum: ModelReferencePointCount (<i>Model</i>) - 1

Remark(s)

This property is expressed with respect to the model's origin.

Example

```
Dim XPos As Single
XPos = Locator1.ModelReferencePointPositionX(0,0)
XPos = Locator1.ModelReferencePointPositionX("PartModelA", 2)
```

ModelReferencePointPositionY**Description**

Y coordinate of the selected reference point on the selected model. Expressed in the calibrated units specified by the **UnitsLength** property. Read only.

Syntax

control.**ModelReferencePointPositionY**(*Model* As Variant, *Index* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.
<i>Index</i>	Index of the selected reference point.	Integer	Minimum: 0 Maximum: ModelReferencePointCount (<i>Model</i>) - 1

Remark(s)

This property is expressed with respect to the model's origin.

Example

```
Dim YPos As Single
```

```
YPos = Locator1.ModelReferencePointPositionY(0,0)
```

```
YPos = Locator1.ModelReferencePointPositionY("PartModelA", 2)
```

ModelShadingAreaBottom

Description

The Y position of the bottom of the shading area that is bounded by **ModelShadingAreaBottom**, **ModelShadingAreaLeft**, **ModelShadingAreaRight** and **ModelShadingAreaTop**. The shading area is used for shading consistency analysis when the **InstanceOrdering** property is set to **hsShadingConsistency**. Read only.

Syntax

control.**ModelShadingAreaBottom** [=*Model* As Variant]

Type

Integer

Range

Minimum: 0

Maximum: Height of the Model image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim lBottom As Integer
lBottom = Locator1.ModelShadingAreaBottom (0)
```

ModelShadingAreaLeft**Description**

The X position of the left side of the shading area that is bounded by **ModelShadingAreaBottom**, **ModelShadingAreaLeft**, **ModelShadingAreaRight** and **ModelShadingAreaTop**. The shading area is used for shading consistency analysis when the **InstanceOrdering** property is set to **hsShadingConsistency**. Read only.

Syntax

control.**ModelShadingAreaLeft** [=Model As Variant]

Type

Integer

Range

Minimum: 0

Maximum: Width of the Model image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim lLeft As Integer
lLeft = Locator1.ModelShadingAreaLeft (0)
```

ModelShadingAreaRight**Description**

The X position of the right side of the shading area that is bounded by **ModelShadingAreaBottom**, **ModelShadingAreaLeft**, **ModelShadingAreaRight** and **ModelShadingAreaTop**. The shading area is used for shading consistency analysis when the **InstanceOrdering** property is set to **hsShadingConsistency**. Read only.

Syntax

control.**ModelShadingAreaRight** [=Model As Variant]

Type

Integer

Range

Minimum: 0

Maximum: Width of the Model image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim lRight As Integer
lRight = Locator1.ModelShadingAreaRight (0)
```

ModelShadingAreaTop

Description

The Y position of the top of the shading area that is bounded by **ModelBoundingAreaBottom**, **ModelBoundingAreaLeft**, **ModelBoundingAreaRight** and **ModelShadingAreaTop**. The shading area is used for shading consistency analysis when the **InstanceOrdering** property is set to **hsShadingConsistency**. Read only.

Syntax

control.**ModelShadingAreaTop** [=*Model* As Variant]

Type

Integer

Range

Minimum: 0

Maximum: Height of the Model image.

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim lTop As Integer
lTop = Locator1.ModelShadingAreaTop (0)
```

ModelTrackingInertia

Description

The tracking inertia setting used to detect the contours of the model. Read only.

Syntax

control.**ModelTrackingInertia** (*Model* As Variant)

Type

Integer

Range

Minimum: 0

Maximum: 1

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the selected model.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Example

Dim Value As Integer

Value = LocatorTool.**ModelTrackingInertia**("hook")

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a Model Image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax*control*.**ModelTransformEnabled** [= *Value* As Boolean]**Type**

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

Dim Enabled As Boolean

Enabled = Locator2.**ModelTransformEnabled**Locator3.**ModelTransformEnabled** = True

Name

Description

Name used to access the Locator tool through the Process Manager.

Syntax*control*.**Name****Type**

String

Range

Not applicable.

Example

```
Dim LocatorName As String
LocatorName = Locator1.Name
Locator2.Name = "Locator"
```

NextModelName

Description

Moves the internal model name iterator to the next model and returns its name. Returns an empty string after the last model is reached. This internal iterator can be used to browse the model name with the following properties: **FirstModelName**, **LastModelName**, **NextModelName** and **PreviousModelName**. Read only.

Syntax

control.**NextModelName**

Type

String

Range

Not applicable.

Example

```
Dim Name As String
Name = Locator1.NextModelName
```

NominalRotation

Description

Required angle of rotation for an object instance to be recognized. Expressed in the orientation units specified by the **UnitsOrientation** property.

Syntax

control.**NominalRotation** [= Value As Single]

Type

Single

Range

Minimum: -180.0 degrees or -PI rad
Maximum: 180.0 degrees or PI rad

Remark(s)

This property is applicable only if the **NominalRotationEnabled** property is set to True.

Example

```
Dim NomRotThreshold As Single
NomRotThreshold = Locator1.NominalRotation
Locator2.NominalRotation = 0.0
```

NominalRotationEnabled

Description

Specifies whether the rotation of a recognized instance must fall within the range set by **MinimumRotation** and **MaximumRotation** or be equal to the nominal value set by the

NominalRotation property. When **NominalRotationEnabled** is set to True, the nominal value is applied, otherwise the range is used.

Syntax

control.**NominalRotationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Locator searches for instances that meet NominalRotation constraint.
False	Locator searches for instances within range set by MinimumRotation and MaximumRotation .

Example

```
Dim NomRotThresholdEnabled As Boolean
NomRotThresholdEnabled = Locator1.NominalRotationEnabled
Locator2.NominalRotationEnabled = True
```

NominalScaleFactor

Description

Required scale factor for an object instance to be recognized.

Syntax

control.**NominalScaleFactor** [= *Value* As Single]

Type

Single

Range

Minimum: 0.1
Maximum: 10.0

Remark(s)

This property is applicable only if the **NominalScaleFactorEnabled** property is set to True.

Example

```
Dim NominalScaleFactor As Single
NominalScaleFactor = Locator1.NominalScaleFactor
Locator2.NominalScaleFactor = 2.0
```

NominalScaleFactorEnabled

Description

Specifies whether the scale factor of a recognized instance must fall within the range set by **MinimumScaleFactor** and **MaximumScaleFactor** or be equal to the nominal value set by the **NominalScaleFactor** property. When **NominalScaleFactorEnabled** is set to True, the nominal value is applied, otherwise the range is used.

Syntax

control.**NominalScaleFactorEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Locator searches for instances that meet NominalScaleFactor constraint.
False	Locator searches for instances within range set by MinimumScaleFactor and MaximumScaleFactor .

Example

```
Dim NominalScaleFactorEnabled As Boolean
NominalScaleFactorEnabled = Locator1.NominalScaleFactorEnabled
Locator2.NominalScaleFactorEnabled = False
```

OutlineLevel

Description

The coarseness of the contours at the Outline level. This property can only be set when **ParametersBasedOn** is set to **hsParametersCustom**. Read only otherwise.

Syntax

control.**OutlineLevel**

Type

Integer

Range

Minimum: 1
Maximum: 16

Remark(s)

For most applications, the **ParametersBasedOn** property should be set to **hsParametersAllModels**. Custom contour detection should only be used when the default values do not work correctly.

Example

```
Locator2.OutlineLevel = 8
Locator2.Execute
```

OutputDetailScene

Description

Name of the output Detail Contour Scene. The Detail Contour Scene contains the vectorized description of the contours generated by the Locator at the Detail level.

Syntax

control.**OutputDetailScene** [= Value As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim DetailSceneName As String
```

```
DetailSceneName = Locator1.OutputDetailScene
Locator2.OutputDetailScene = "Scene"
```

OutputDetailSceneEnabled

Description

When **OutputDetailSceneEnabled** is set to True, the Detail Contour Scene, set by the **OutputDetailScene** and **OutputDetailSceneView** properties, is output to the runtime database.

Syntax

```
control.OutputDetailSceneEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Detail Contour Scene is output to runtime database.
False	Detail Contour Scene is not output to runtime database.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.OutputDetailSceneEnabled
Locator2.OutputDetailSceneEnabled = True
```

OutputDetailSceneView

Description

Name of the View in the runtime database to which the Detail Contour Scene is output. The **OutputDetailScene** property sets the name of the Detail Contour Scene.

Syntax

```
control.OutputDetailSceneView [= Value As String]
```

Type

String

Range

Not applicable.

Remark(s)

See **OutputDetailScene** and **OutputDetailSceneEnabled** for more details.

Example

```
Dim OutputDetailView As String
OutputDetailView = Locator1.OutputDetailSceneView
Locator2.OutputDetailSceneView = "Locator"
```

OutputInstanceScene

Description

Name of the scene containing the object instances located by the Locator tool. When **OutputInstanceSceneEnabled** is set to true, this scene is output in the runtime database, in the view set by the **OutputInstanceSceneView** property.

Syntax

control.**OutputInstanceScene** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim OutputInstanceScene As String
OutputInstanceScene = Locator1.OutputInstanceScene
Locator2.OutputInstanceScene = "Instance Scene"
```

OutputInstanceSceneEnabled

Description

When **OutputInstanceSceneEnabled** is set to True, the instance scene set by the **OutputInstanceScene** and **OutputInstanceSceneView** properties is output in the runtime database.

Syntax

control.**OutputInstanceSceneEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Instance Scene is output to runtime database.
False	Instance Scene is not output to runtime database.

Remark(s)

For normal searches using both the Outline and Detail levels, the models at the Detail level are used to draw the instances. When **SearchBasedOnOutlineLevelOnly** is True, the models at the Outline level are used.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.OutputInstanceSceneEnabled
Locator2.OutputInstanceSceneEnabled = True
```

OutputInstanceSceneView

Description

Name of the View in the runtime database to which the Instance Scene is output. The **OutputInstanceScene** property sets the name of the instance scene.

Syntax

control.**OutputInstanceSceneView** [= *Value* As String]

Type

String

Range

Not applicable.

Remark(s)

For normal searches using both the Outline and Detail levels, the models at the Detail level are used to draw the instances. When **SearchBasedOnOutlineLevelOnly** is True, the models at the Outline level are used.

Example

```
Dim OutputInstanceView As String
OutputInstanceView = Locator1.OutputInstanceSceneView
Locator2.OutputInstanceSceneView = "Locator"
```

OutputMode

Description

Mode used to output object instances in the Instance Scene.

Syntax

control.**OutputMode** [= *Value* As hsLocatorOutputMode]

Type

hsLocatorOutputMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsNoGraphics	0	The output Scene does not contain any graphical representation of object instances.
hsTransformedModel	1	In the output scene, an object instance is represented by transforming its associated model according the pose computed by the Locator.
hsMatchedModel	3	In the output scene, an object instance is represented by transforming the sections of its model that were matched to actual contours according the pose computed by the Locator.

Remark(s)

For normal searches using both the Outline and Detail levels, the models at the Detail level are used to draw the instances. When **SearchBasedOnOutlineLevelOnly** is True, the models at the Outline level are used. Setting this property to **hsMatchedModel** or **hsTransformedModel** will usually increase the processing time. It should be set to **hsNoGraphics** for optimal performance.

Example

```
Dim OutputMode As hsLocatorOutputMode
OutputMode = Locator1.OutputMode
Locator2.OutputMode = hsMatchedModel
```

OutputModelEnabled

Description

When **OutputModelEnabled** is set to True, the models enabled in the models database are output in the runtime database, in the view set by the **OutputModelView** property.

Syntax

control.**OutputModelEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Enabled Models are output to runtime database.
False	Enabled Models are not output to runtime database.

Remark(s)

For normal searches using both the Outline and Detail levels, the models at the Detail level are output in the model view. When **SearchBasedOnOutlineLevelOnly** is True, the models at the Outline level are output.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.OutputModelEnabled
Locator2.OutputModelEnabled = True
```

OutputModelView

Description

The View to which the enabled models are output in the runtime database when the **OutputModelEnabled** property is set to True.

Syntax

control.**OutputModelView** [= *Value* As String]

Type

String

Range

Not applicable.

Remark(s)

For normal searches using both the Outline and Detail levels, the models at the Detail level are output in the model view. When **SearchBasedOnOutlineLevelOnly** is True, the models at the Outline level are output.

Example

```
Dim OutputSceneView As String
OutputSceneView = Locator1.OutputModelView
Locator2.OutputModelView = "Locator"
```

OutputOutlineScene

Description

Name of the output Outline Contour Scene. The Outline Contour Scene contains the vectorized description of the contours generated by the Locator at the Outline level.

Syntax

control.**OutputOutlineScene** [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Example

```
Dim OutlineSceneName As String
OutlineSceneName = Locator1.OutputOutlineScene
Locator2.OutputOutlineScene = "Scene"
```

OutputOutlineSceneEnabled

Description

When **OutputOutlineSceneEnabled** is set to True, the Outline Contour Scene, set by the **OutputOutlineScene** and **OutputOutlineSceneView** properties, is output to the runtime database.

Syntax

control.**OutputOutlineSceneEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Outline Contour Scene is output to runtime database.
False	Outline Contour Scene is not output to runtime database.

Parameter(s)

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.OutputOutlineSceneEnabled
Locator2.OutputOutlineSceneEnabled = True
```

OutputOutlineSceneView

Description

Name of the View in the runtime database to which the Outline Contour Scene is output. The **OutputOutlineScene** property sets the name of the Outline Contour Scene.

Syntax

control.**OutputOutlineSceneView** [= *Value* As String]

Type

String

Range

Not applicable.

Parameter(s)

Not applicable.

Remark(s)

See **OutputOutlineScene** and **OutputOutlineSceneEnabled** for more details.

Example

```
Dim OutputSceneView As String
OutputSceneView = Locator1.OutputOutlineSceneView
Locator2.OutputOutlineSceneView = "Locator"
```

OutputSymmetricInstances

Description

When set to true, all the symmetric pose of the object instances are output. If False, only the best quality of the symmetric poses of the object instance is output.

Syntax

control.**OutputSymmetricInstances** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Locator outputs all symmetrical poses of an instance.
False	Locator outputs only single best pose of an instance.

Remark(s)

See also **InstanceSymmetry**.

Example

```
Dim OutputSymmetricInstances As Boolean
OutputSymmetricInstances = Locator1.OutputSymmetricInstances
Locator2.OutputSymmetricInstances = False
```

ParametersBasedOn

Description

Sets how the contour detection parameters are configured. When set to **hsContourParametersAllModels**, the contour detection parameters are optimized by analyzing the parameters used to build all the models. When set to **hsContourParametersCustom**, the contour detection parameters are set manually. When set to a value greater than **hsContourParametersCustom**, the contour detection parameters of a specific model are used. The contour detection parameters on which this property has an effect are **DetailLevel**, **OutlineLevel**, **ContrastThresholdMode**, **ContrastThreshold**, and **TrackingInertia**.

Syntax

control.**ParametersBasedOn** [= *Value* As hsContourParameters]

Type

hsContourParameters

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContourParametersAllModels	-2	The contour detection parameters are optimized by analyzing the parameters used to build all the models.
hsContourParametersCustom	-1	The contour detection parameters are set manually.
Integer	Greater than or equal to 0.	The contour detection parameters of a specific model are used.

Remark(s)

For most applications, the **ParametersBasedOn** property should be set to **hsParametersAllModels**. Custom contour detection should only be used when the default values do not work correctly.

Example

```
Dim Parameters As hsContourParameters
Parameters = Locator1.ParametersBasedOn
Locator2.ParametersBasedOn = hsContourParametersAllModels
Locator2.ParametersBasedOn = 2
```

PercentageOfPointsToAnalyze

Description

The percentage of points on a model contour that are actually used for the optimization process. For example, when **PercentageOfPointsToAnalyze** is set to the default 50% value, one out of two points are used. Increasing this value can increase the accuracy of the optimized model but incurs a longer optimization time.

Syntax

control.**PercentageOfPointsToAnalyze** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: 100.0

Example

```
Dim PointPercentage As Single
PointPercentage = Locator1.PercentageOfPointsToAnalyze
Locator2.PercentageOfPointsToAnalyze = 50.0
```

PositioningLevel

Description

Configurable effort level of the instance positioning process. A value of 0 will provide coarser positioning and lower execution time. Conversely, a value of 10 will provide high accuracy positioning of object instances.

Syntax

control.**PositioningLevel** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 10

Example

```
Dim PositioningLevel As Integer
PositioningLevel = Locator1.PositioningLevel
Locator2.PositioningLevel = 5
```

PreviousModelName

Description

Moves the internal model name iterator to the previous model and returns its name. Returns an empty string after the first model is reached. This internal iterator can be used to browse the model name with the following properties: **FirstModelName**, **LastModelName**, **NextModelName** and **PreviousModelName**. Read only.

Syntax

```
control.PreviousModelName
```

Type

String

Range

Not applicable.

Example

```
Dim Name As String
Name = Locator1.PreviousModelName
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

```
control.ProcessingFormat [= Value As hsProcessingFormat]
```

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be performed using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.

Constant	Value	Description
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

RecognitionLevel

Description

Configurable effort level of the search process. A value of 0 will lead to a faster search that may miss instances that are partly occluded. Conversely, a value of 10 is useful for finding partly occluded objects in cluttered or noisy images, or for models made up of small features at the Outline Level.

Syntax

control.**RecognitionLevel** [= Value As Integer]

Type

Integer

Range

Minimum: 0
Maximum: 10

Example

```
Dim RecognitionLevel As Integer
RecognitionLevel = Locator1.RecognitionLevel
Locator2.RecognitionLevel = 5
```

RespectContrastPolarity

Description

Note: HexSight supports the **RespectContrastPolarity** property for compatibility with applications built in previous versions. You should use the **ContrastPolarity** property to set polarity. **RespectContrastPolarity** will be handled as follows: if **RespectContrastPolarity** is True, the polarity is set to **hsContrastPolarityNormal**, if False, polarity is set to **hsContrastPolarityDontCare**.

Syntax

control.**RespectContrastPolarity** [= Value As Boolean]

Type

Boolean

Range

True, False.

Remark(s)

Setting **RespectContrastPolarity** to True makes the search faster.

Example

```
Dim RespectContrastPol As Boolean
RespectContrastPol = Locator1.RespectContrastPolarity
Locator2.RespectContrastPolarity = True
```

SearchAreaBottom

Description

Note: HexSight supports the **SearchAreaBottom** property for compatibility with applications built in previous versions. You should define the tool's region of interest with **ToolHeight**, **ToolWidth**, **ToolPositionX**, **ToolPositionY**, **ToolRotation**, and **ToolUseEntireImage**.

SearchAreaLeft

Description

Note: HexSight supports the **SearchAreaLeft** property for compatibility with applications built in previous versions. You should define the tool's region of interest with: **ToolHeight**, **ToolWidth**, **ToolPositionX**, **ToolPositionY**, **ToolRotation**, and **ToolUseEntireImage**.

SearchAreaRight

Description

Note: HexSight supports the **SearchAreaRight** property for compatibility with applications built in previous versions. You should define the tool's region of interest with: **ToolHeight**, **ToolWidth**, **ToolPositionX**, **ToolPositionY**, **ToolRotation**, and **ToolUseEntireImage**.

SearchAreaTop

Description

Note: HexSight supports the **SearchAreaTop** property for compatibility with applications built in previous versions. You should define the tool's region of interest with: **ToolHeight**, **ToolWidth**, **ToolPositionX**, **ToolPositionY**, **ToolRotation**, and **ToolUseEntireImage**.

SearchAreaUseEntireImage

Description

Note: HexSight supports the **SearchAreaUseEntireImage** property for compatibility with applications built in previous versions. You should define the tool's region of interest with: **ToolHeight**, **ToolWidth**, **ToolPositionX**, **ToolPositionY**, **ToolRotation**, and **ToolUseEntireImage**.

SearchBasedOnOutlineLevelOnly

Description

When set to True, the Locator does not use the models at the Detail level for the positioning process. This mode can be used to improve the speed when only a coarse positioning of object instances is required.

Syntax

```
control.SearchBasedOnOutlineLevelOnly [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Only Outline Level Models are used to position instances.
False	Both Outline Level and Detail Level Models are used to position instances.

Example

```
Dim UseOutlineOnly As Boolean
UseOutlineOnly = Locator1.SearchBasedOnOutlineLevelOnly
Locator2.SearchBasedOnOutlineLevelOnly = True
```

SearchTime

Description

Time elapsed (in milliseconds) for the search process during the last execution of the Locator tool. Read only.

Syntax

control.**SearchTime**

Type

Single

Range

Greater than 0

Example

```
Dim TotalTime As Single
Dim LearnTime As Single
Dim SearchTime As Single
Locator1.Execute
TotalTime = Locator1.ElapsedTime
LearnTime = Locator1.LearnTime
SearchTime = Locator1.SearchTime
```

ShowContourSceneDetail

Description

Specifies if the contours at the Detail level are shown in the Results Viewer.

Syntax

control.**ShowContourSceneDetail** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Detail Level contours are displayed in the Results Viewer window.
False	Detail Level contours are not displayed in the Results Viewer window.

Example

```
Dim Enabled As Boolean
```

```
Enabled = Locator1.ShowContourSceneDetail
Locator1.ShowContourSceneDetail = False
```

ShowContourSceneOutline

Description

Specifies if the contours at the Outline level are shown in the Results Viewer.

Syntax

```
control.ShowContourSceneOutline [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Outline Level contours are displayed in the Results Viewer window.
False	Outline Level contours are not displayed in the Results Viewer window.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ShowContourSceneOutline
Locator1.ShowContourSceneOutline = False
```

ShowCoordinateSystem

Description

Specifies if the object coordinate system markers are shown in the Results Viewer.

Syntax

```
control.ShowCoordinateSystem [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Coordinate System marker is displayed in the Results Viewer window.
False	Coordinate System marker is not displayed in the Results Viewer window.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ShowCoordinateSystem
Locator1.ShowCoordinateSystem = False
```

ShowImage

ShowGreyScaleImage

Description

Specifies if the input image is shown in the Results Viewer.

Syntax

control.**ShowImage** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Image is displayed in the Results Viewer window.
False	Image is not displayed in the Results Viewer window.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ShowImage
Locator1.ShowImage = False
```

ShowInstanceScene

Description

Specifies if object instances are shown in the Results Viewer.

Syntax

control.**ShowInstanceScene** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Instance Scene is displayed in the Results Viewer window.
False	Instance Scene image is not displayed in the Results Viewer window.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ShowInstanceScene
Locator1.ShowInstanceScene = False
```

ShowInstanceSceneAs

Description

Mode used to draw object instances in the Results Viewer.

Syntax

control.**ShowInstanceSceneAs** [= *Value* As hsLocatorOutputMode]

Type

hsLocatorOutputMode

Range

The valid range for this property is as follows:

hsLocatorOutputMode	Value	Description
hsTransformedModel	1	In the Results display, the found arc is represented by the arc hypotheses fitted to match actual edges.
hsMatchedModel	2	In the Results display, the found arc is represented by only the sections of the arc hypotheses that were matched to actual edges.

Example

```
Dim DisplayMode As hsLocatorOutputMode
DisplayMode = Locator1.ShowInstanceSceneAs
Locator2.ShowInstanceSceneAs = hsMatchedModel
```

ShowReferencePoints

Description

Specifies if reference points target markers are shown in the Results Viewer.

Syntax

control.**ShowReferencePoints** [= *Value* As Boolean]

Type

Value	Description
True	Reference Point markers are displayed in the Results Viewer window.
False	Reference Point markers are not displayed in the Results Viewer window.

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ShowReferencePoints
Locator1.ShowReferencePoints = False
```

ShowSearchArea

Description

Specifies if the Search Area bounding box is shown in the Results Viewer.

Syntax

control.**ShowSearchArea** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Rectangle bounding box is displayed in the Results Viewer window.
False	Rectangle bounding box is not displayed in the Results Viewer window.

Example

```
Dim Enabled As Boolean
Enabled = Locator1.ShowSearchArea
Locator1.ShowSearchArea = False
```

Timeout

Description

Time (in milliseconds) after which the Locator tool aborts its search process. This timeout period does not include the model learning phase. When the timeout is reached and **TimeoutEnabled** is set to True, the instances recognized up to the timeout are output by the Locator and the search is aborted.

Syntax

control.Timeout [= Value As Integer]

Type

Integer

Range

Greater than zero.

Remark(s)

It may take the Locator a few milliseconds more than the time specified by **Timeout** to abort due to internal and operating system latencies. This property is applied only when **TimeoutEnabled** is set to True.

Example

```
Dim Timeout As Integer
Timeout = Locator1.Timeout
Locator2.Timeout = 150
```

TimeoutEnabled

Description

Specifies if the timeout period set by the **Timeout** property is used to limit the search time.

Syntax

control.TimeoutEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Search time is limited by Timeout property.
False	Search time is not limited.

Example

```
Dim TimeoutEnable As Boolean
TimeoutEnable = Locator1.TimeoutEnable
Locator2.TimeoutEnable = False
```

ToolHeight

Description

Height of the tool rectangle.

Syntax

control.**ToolHeight** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height as Single
Height = Locator2.ToolHeight
Locator3.ToolHeight = 15.2
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionX** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim X_Pos As Single
X_Pos = Locator2.ToolPositionX
Locator3.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = Locator2.ToolPositionY
Locator3.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = Locator2.ToolRotation
Locator3.ToolRotation = -10.0
```

ToolUseEntireImage

Description

Specifies whether When set to True, the entire input image is used for the search. When set to False, the tool searches only within the tool rectangle defined by **ToolPositionX**, **ToolPositionY**, **ToolWidth**, **ToolHeight** and **ToolRotation**.

Syntax

control.**SearchAreaUseEntireImage** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The Locator uses the entire image as the search area.
False	The Locator uses the Rectangle region of interest as search area.

Example

```
Dim UseEntireImage As Boolean
UseEntireImage = Locator1.ToolUseEntireImage
Locator2.ToolUseEntireImage = True
```

ToolWidth

Description

Width of the tool rectangle.

Syntax

control.**ToolWidth** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = Locator2.ToolWidth
Locator3.ToolWidth = 23.4
```

TrackingInertia

Description

The inertia of the edge tracking phase where individual edge elements are connected to form the contours. It corresponds to the maximum distance, in pixels, between two edge elements that can be connected to form a continuous contour. This property can only be set when **ParametersBasedOn** is set to **hsParametersCustom**. Read only otherwise.

Syntax

control.TrackingInertia

Type

Integer

Range

Minimum: 0
Maximum: 1

Remark(s)

For most applications, the **ParametersBasedOn** property should be set to **hsParametersAllModels**. Custom contour detection should only be used when the default values do not work correctly.

Example

```
Dim Inertia As Integer
Inertia = Locator1.TrackingInertia
Locator2.TrackingInertia = 1
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = Locator1.TransformRotation
Locator2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: Greater than zero.
Maximum: Unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = Locator1.TransformScale
Locator2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = Locator1.TransformTranslationX
Locator2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = Locator1.TransformTranslationY
Locator2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the Locator tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSLocator".

Example

```
Dim TypeName As String
TypeName = Locator1.TypeName
```

UnitsLength

Description

Unit used to express calibrated lengths and positions.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnits	8	generic calibration units

Example

```
Dim UnitsLength As hsUnitsLength
UnitsLength = Locator1.UnitsLength
Locator2.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express orientation.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	Radian
hsDegree	1	Degree

Example

```
Dim UnitsOrientation As hsUnitsOrientation
UnitsOrientation = Locator1.UnitsOrientation
Locator2.UnitsOrientation = hsRadian
```

UseDefaultConformityTolerance

Description

Specifies whether the default conformity tolerance returned by the **DefaultConformityTolerance** property is used instead of the user-defined conformity tolerance set by the **ConformityTolerance** property. When set to True, the default conformity tolerance is used.

Syntax

control.**UseDefaultConformityTolerance** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The Locator uses the entire image as the search area.
False	The Locator uses the Rectangle region of interest as search area.

Example

```
Dim UseDefaultCT As Boolean
UseDefaultCT = Locator1.UseDefaultConformityTolerance
Locator2.UseDefaultConformityTolerance = False
```

UseExternalModelDatabase

Description

Specifies if an external models database provided by another instance of the Locator tool should be used to provide models instead of the normal internal models database. The handle to the external models database is provided by the **ModelDatabase** property. Models database sharing can be used to reduce runtime memory usage in applications where the same models must be searched in different search areas or with different parameters.

Syntax

control.**UseExternalModelDatabase** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
False	Models are provided by the current Locator tool.
True	Models are provided by an external Models database.

Example

```
Locator2.UseExternalModelDatabase = True
Locator2.ModelDatabase = Locator1.ModelDatabase
Locator2.Execute
```

UseMinimumClearPercentage

Description

Specifies whether the tool uses the **MinimumClearPercentage** property.

Syntax

```
control.UseMinimumClearPercentage[= Value As Boolean]
```

Type

Boolean

Range

Not applicable.

Example

```
Dim UseClearPercentage As Boolean
UseClearPercentage = Locator1.UseMinimumClearPercentage
Locator2.UseMinimumClearPercentage = False
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

```
control.Version
```

Type

String

Range

Not applicable.

Remark(s)

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Locator1.Version
```

Methods

AboutBox

Description

Displays the Locator tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

Locator1.**AboutBox**

AddModel

Description

Adds a new blank model to the models database. Note that the newly created model does not contain any features, it only contains a copy of the current input image of the Locator. Adding features to this blank model can only be done through the Model Editor, either interactively or programmatically. The **ShowModelEditor** method and the **HSMModelEditorInterface** object can be used, to programmatically add features to the model.

Syntax

control.**AddModel**(*ModelName* As String) As Integer

Parameter(s)

Name	Description	Type	Range
<i>ModelName</i>	Name of the Model to be added.	String	Not applicable.

Return

Type	Description	Range
Integer	Index of the newly created model.	Successful: Greater than or equal to 0 Failed: -1

Example

```
Dim NewIndex As Integer
NewIndex = Locator1.AddModel("ObjectA")
```

ClearMaskImage

Description

Clears the current mask image.

Syntax

control.**ClearMaskImage**

Parameter(s)

None

Return

No return value.

ExampleLocator1.**ClearMaskImage**

CompactMemory

Description

Compacts all the internal structures used by the Locator's search process to their minimal size, without unlearning the models. Unlike the **Unlearn** method, the next execution of the Locator is not slowed down since models do not have to be re-learned. **CompactMemory** can be used to reduce memory usage in the following cases: after the size search area has been reduced, after the size of the input image has been reduced, after processing a highly cluttered image containing a very large number of instances. Compacting the memory will only reduce memory usage, it will not make the search process execute faster.

Syntax*control*.**CompactMemory****Parameter(s)**

None

Return

None

Example

```
' Search the entire image
Locator1.SearchAreaUseEntireImage = True
Locator1.Execute
' Reduce the search area
Locator1.SearchAreaLeft = 100
Locator1.SearchAreaRight = 150
Locator1.SearchAreaBottom = 200
Locator1.SearchAreaTop = 250
' Compact the search structures to reduce memory usage
Locator1.CompactMemory
Locator1.Execute
```

ControlPropertyExchange

Description

Reserved for internal use.

CopyModel

Description

Copies the selected model to a new model in the models database.

Syntax*control*.**CopyModel**(*Model* As Variant, *NewName* As String) As Integer

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the model to be copied (or source model).	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.
<i>NewName</i>	Name of the new model.	String	Not applicable.

Return

Type	Description	Range
Integer	Index of the newly created model.	Successful: Greater than or equal to 0 Failed: -1

Example

```
Dim NewIndex As Integer
NewIndex = Locator1.CopyModel("modelA", "modelA_copy")
NewIndex = Locator2.CopyModel(2, "modelB_copy")
```

Execute

Description

Executes the Locator process.

Syntax

control.**Execute**() As Single

Parameter(s)

None

Return

Type	Description	Range
Single	Execution time in milliseconds.	Greater than 0.0

Example

```
Time As Single
Time = Locator1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetInstanceIntrinsicBoundingBox

Description

Returns the bounding box of the selected object instance. Bounding box is returned as four points representing the four corners of the bounding box. Read only

Syntax

control.**GetInstanceIntrinsicBoundingBox**(*InstanceIndex* As Integer,
P0 As HSPoint *, *P1* As HSPoint *, *P2* As HSPoint *, *P3* As HSPoint *)

Parameter(s)

Name	Description	Type	Range
<i>InstanceIndex</i>	Index of the instance.	Integer	Minimum: 0 Maximum: InstanceCount - 1
<i>P0</i>	Reference to HSPoint Bottom-Left corner	HSPoint *	
<i>P1</i>	Reference to HSPoint Bottom-Right corner	HSPoint *	
<i>P2</i>	Reference to HSPoint Top-Left corner	HSPoint *	
<i>P3</i>	Reference to HSPoint Top-Right corner	HSPoint *	

Example

```
Dim P(4) As HSPoint
Set P(0) = Database.AddPoint("Boundary", "P0")
Set P(1) = Database.AddPoint("Boundary", "P1")
Set P(2) = Database.AddPoint("Boundary", "P2")
Set P(3) = Database.AddPoint("Boundary", "P3")
Locator1.GetInstanceIntrinsicBoundingBox 0, P(0), P(1), P(2), P(3)
```

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

control.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see ProcessingFormat property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideModelEditor

Description

Hides and terminates the Model Editor dialog.

Syntax

control.**HideModelEditor** As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Locator1.HideModelEditor
```

HideModelOptimizer

Description

Hides and terminates the Model Optimizer dialog.

Syntax

control.**HideModelOptimizer** As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim OptimizerWasDisplayed As Boolean
OptimizerWasDisplayed = Locator1.HideModelOptimizer
```

HideProperties

Description

Hides the Locator properties interface.

Syntax*control.HideProperties***Parameter(s)**

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Locator1.HideProperties
```

HideResultsViewer

Description

Hides the results viewer.

Syntax*control.HideResultsViewer***Return**

Boolean

Range

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Locator1.HideResultsViewer
```

ImportModel

Description

Imports a model to the current model database from another model database file.

Syntax*control.ImportModel([FileName As Variant], [Model As Variant])***Parameter(s)**

Name	Description	Type	Range
<i>FileName</i>	Name of the file to load. Optional.	Variant	Must be a string.
<i>Model</i>	Index or name of the model to import. Optional.	Variant	Integer: String: Range not applicable. Other types not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

Since the range cannot be found for the external model database, if the parameter is an index, the number of the model must be known beforehand.

If no FileName is specified, the default File Open dialog is displayed to allow the selection of the model database file. If no Model is specified, an Import Model dialog is displayed to allow the selection of the model(s) to be imported.

Example

```
Locator1.ImportModel "myModelDatabase.hdb", "Model2"
Locator1.ImportModel "myModelDatabase.hdb", 2
```

Learn

Description

Forces the Locator to learn the models. After models are modified and/or some of the search parameters are changed, the Locator must learn the models to optimize the search process. This learn process is automatically called the first time the Locator is executed after models and/or parameters are changed. To avoid the extra processing time added by the learn process at the first execution, the **Learn** method can be called explicitly.

Syntax

control.**Learn**() As Single

Parameter(s)

None

Return

Type	Description	Range
Single	Learn time in milliseconds.	Greater than 0.0

Example

```
Dim Time As Single
' Load a new models database and learn the models
Locator1.LoadModelDatabase "PartModel.hdb"
Time = Locator1.Learn
' This execution will not include a learn process
Locator1.Execute
```

LoadConfiguration

Description

Reserved for internal use.

LoadMaskImage

Description

Loads a new mask image file.

Syntax

control.**LoadMaskImage**([*FileName* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of the file to load. Optional.	Variant	Must be a string.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If no *FileName* parameter is provided, a file open dialog appears.

Example

```
Dim Loaded As Boolean
Loaded = Locator.LoadMaskImage
Locator1.LoadMaskImage "... \MaskImage"
```

LoadModelDatabase

Description

Loads a new models database file.

Syntax

control.**LoadModelDatabase**([*FileName* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of the file to load. Optional.	Variant	Must be a string.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If no *FileName* parameter is provided, a file open dialog appears.

Example

```
Dim Loaded As Boolean
Loaded = Locator.LoadModelDatabase
Locator1.LoadModelDatabase "... \Models\EngineCover"
```

ModelOptimizerReset

Description

Clears all optimized data recorded so far.

Syntax

control.**ModelOptimizerReset**

Parameter(s)

None

Return

No return value.

Example

Locator.**ModelOptimizerReset**

NewModelDatabase

Description

Clears the current models database.

Syntax

control.**NewModelDatabase**

Parameter(s)

None

Return

No return value.

Example

Locator1.**NewModelDatabase**

RemoveModel

Description

Removes a model from the models database.

Syntax

control.**RemoveModel**(*Model* As Variant) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the model to be removed.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

Locator1.**RemoveModel** "Model0"

Locator1.**RemoveModel** 0

RenameModel

Description

Renames a model in the models database.

Syntax

control.**RenameModel**(*Model* As Variant, *ModelName* As String) As Integer

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Index or name of the model to be renamed.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.
<i>ModelName</i>	New name of the model	String	Not applicable.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeeded As Boolean
Succeeded = Locator1.RenameModel("Model0", "NewModel0")
Succeeded = Locator1.RenameModel(0, "NewModel0")
```

SaveConfiguration

Description

Reserved for internal use.

SaveModelDatabase

Description

Saves the models database file.

Syntax

control.**SaveModelDatabase**(*FileName* As Variant, [*SaveImages* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of the file to save. Optional.	Variant	Must be a string.
<i>SaveImages</i>	True to include the images in the Model Database (default). Optional.	Variant	Must be a boolean. True, False

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If no *FileName* parameter is provided, a file save dialog appears. *SaveImages* is set to True by default. If *SaveImages* is set to False, the images will be removed from the database; although it takes less disk/memory space, it eliminates the possibility to eventually modify or rebuild the model.

Example

```
Dim Succeeded As Boolean
Succeeded = Locator1.SaveModelDatabase("..\Parts.hdb", True)
Succeeded = Locator2.SaveModelDatabase
```

SetDebugParameter

Description

Reserved for internal use.

ShowModelEditor

Description

Opens the Model Editor interface in a modeless window. A reference to an **HSMModelEditorInterface** object is returned for programmatic access to the dialog. The Model Editor can be opened but left invisible to edit models programmatically without showing the interactive user interface. When programmatic access to an **HSMModelEditorInterface** is not required, the Model Editor can be opened in a modal window using the **ShowModelEditorModal** method.

Syntax

control.**ShowModelEditor**(*Visible* As Boolean, *Model* As Variant, [*DisplayTopMost* As Variant])
As HSMModelEditorInterface

Parameter(s)

Name	Description	Type	Range
<i>Visible</i>	Specifies if the Model Editor interface will be shown.	Boolean	True, False
<i>Model</i>	Name or index of the model to be edited.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.
<i>DisplayTopMost</i>	When True, the ModelEditor interface window is displayed topmost.	Variant	True, False

Return

Type	Description	Range
HSMModelEditorInterface	Reference to a HSMModelEditorInterface object that can be used to access the Model Editor programmatically.	Not applicable.

Example

```
Dim ModelEditor As HSModelEditorInterface
' Create invisible Model Editor
Set ModelEditor = Locator1.ShowModelEditor(False, 0)
Locator1.ShowModelEditor True, "BrakePad"
```

ShowModelEditorModal

Description

Opens the Model Editor interface in a modal window. Unlike the **ShowModelEditor** method, the main thread of the application is blocked while the Model Editor is active. A reference to an **HSModelEditorInterface** object is therefore not returned.

Syntax

control.**ShowModelEditorModal**(*Model* As Variant) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Name or index of the model to be edited.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Locator1.ShowModelEditorModal "MyModel"
```

ShowModelOptimizer

Description

Opens the Model Optimizer interface in a modeless window. A reference to an **HSModelOptimizerInterface** object is returned for programmatic access to the dialog. The Model Optimizer can be opened but left invisible to edit models programmatically without showing the interactive user interface. When programmatic access to an **HSModelOptimizerInterface** is not required, the Model Optimizer can be opened in a modal window using the **ShowModelOptimizerModal** method.

Syntax

control.**ShowModelOptimizer**(*Visible* As Boolean, *Model* As Variant, [*DisplayTopMost* As Variant]) As HSModelOptimizerInterface

Parameter(s)

Name	Description	Type	Range
<i>Visible</i>	Specifies if the Model Optimizer interface will be shown.	Boolean	True, False

Name	Description	Type	Range
<i>Model</i>	Name or index of the model to be optimized.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.
<i>DisplayTopMost</i>	When True, the ModelOptimizer interface window is displayed topmost.	Variant	True, False

Return

Type	Description	Range
HSMModelOptimizerInterface	Reference to a HSMModelOptimizerInterface object that can be used to access the Model Optimizer programmatically.	Not applicable.

Example

```
Dim Optimizer as HSMModelOptimizerInterface
' Create invisible Model Optimizer
Set Optimizer = Locator.ShowModelOptimizer (False, "Model")
' Adjust parameters
Optimizer.LinkOutlineWithDetail = True
' Show the Model Optimizer in a modeless window
Optimizer.Show
```

ShowModelOptimizerModal

Description

Opens the Model Optimizer interface in a modal window. Unlike the **ShowModelOptimizer** method, the main thread of the application is blocked while the Model Optimizer is active. A reference to an **HSMModelOptimizerInterface** object is therefore not returned.

Syntax

control.**ShowModelOptimizerModal**(*Model* As Variant)

Parameter(s)

Name	Description	Type	Range
<i>Model</i>	Name or index of the model to be edited.	Variant	Integer: Minimum: 0 Maximum: ModelCount - 1 String: Range not applicable. Other types not supported.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Locator1.ShowModelOptimizerModal "MyModel"
```

ShowProperties

Description

Displays the Locator properties interface.

Syntax

```
control.ShowProperties(Modal As Boolean, [Option As Variant], [DisabledTabs As Variant],  
[DefaultTab As Variant]) As Boolean)
```

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	True, False
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface, if the window should be displayed topmost, and what the behavior of the control is when the OK button is clicked. When this parameter is not provided, all the elements of the bitfield are set except <code>hsExecuteOnOK</code> , <code>hsExecuteRemaining</code> and <code>hsDisplayTopMost</code> .	Variant	Bitfield elements shown in table below.
<i>DisabledTabs</i>	Optional bitfield parameter. It specifies the tabs that should be removed (not displayed) from the interface. When this parameter is not provided, no element of the bitfield is set.	Variant	Bitfield elements shown in table below.
<i>DefaultTab</i>	When the properties window opens, displays the default number of tabs in the interface.	Variant	Minimum: 1 Maximum: TabCount - 1

<i>Option Field</i>	Value	Description
<code>hsShowApply</code>	0x01	Displays APPLY button
<code>hsShowReset</code>	0x02	Displays RESET button
<code>hsShowDefault</code>	0x04	Displays DEFAULT button
<code>hsShowExecute</code>	0x08	Displays EXECUTE button
<code>hsShowOK</code>	0x10	Displays OK button
<code>hsExecuteOnOK</code>	0x20	The tool is executed when the OK button is clicked on the interface.

Option Field	Value	Description
hsExecuteRemaining	0x40	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the bitfields hsExecuteOnOK and hsExecuteRemaining are both set.
hsDisplayTopMost	0x80	Displays the dialog as topmost.
hsDisableSaveModelDialogOnOK	0x100	Disables the Save Models Database dialog that opens when changes to models have been made through the user interface.

DisableTabs Field	Value	Description
hsDisableTab1	0x01	Removes tab 1 from the interface
hsDisableTab2	0x02	Removes tab 2 from the interface
hsDisableTab3	0x04	Removes tab 3 from the interface
hsDisableTab4	0x08	Removes tab 4 from the interface
hsDisableTab5	0x10	Removes tab 5 from the interface
hsDisableTab6	0x20	Removes tab 6 from the interface
hsDisableTab7	0x40	Removes tab 7 from the interface
hsDisableTab8	0x80	Removes tab 8 from the interface
hsDisableTab9	0x100	Removes tab 9 from the interface

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeed As Boolean
Succeed = Locator1.ShowProperties(True, hsShowOK + hsShowExecute)
```

ShowResultsViewer

Description

Shows the Results Viewer.

Syntax

control.**ShowResultsViewer**(*Modal* As Boolean, [*DisplayTopMost* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	True, False

Name	Description	Type	Range
<i>DisplayTopMost</i>	When True, the Results Viewer interface window is displayed topmost. Optional	Variant	True, False

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Status As Boolean
Status = Locator1.ShowResultsViewer(True)
```

Start

Description

Initiates a multi-threaded execution of the search process. In this mode, the Locator runs the search process in its own thread without blocking the main thread of the application. The **Wait** method can be used to wait for the completion of the search process. To abort the search process, the **Stop** method can be used.

Syntax

control.**Start** () As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful	True, False

Example

```
Dim X As Single
Dim Y As Single
Dim InstanceIndex As Integer
Dim Running As Boolean
' Limit the search to 10 instances
Locator1.MaximumInstanceCount = 10
' Learn the models
Locator1.Learn
' Start the search process thread
Locator1.Start
' Reset the instance index
InstanceIndex = 0
' Wait until 1 instance is found
Do While ( Wait( 1, -1, Running ) > 0 )
    ' Retrieve the results for the located instance
```

```

    X = Locator1.InstanceTranslationX(InstanceIndex)
    Y = Locator1.InstanceTranslationY(InstanceIndex)
    ' Pick the instance on the conveyor
    PickInstance( X, Y )
    ' Increment the instance index
    InstanceIndex = InstanceIndex + 1
Loop

```

Stop

Description

Stops the search process thread initiated with the **Start** method.

Syntax

control.**Stop** () As Boolean

Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the operation was successful	True, False

Example

```

Dim IncorrectType As Boolean
Dim InstanceType As Integer
Dim X As Single
Dim Y As Single
Dim InstanceIndex As Integer
Dim Running As Boolean
' Limit the search to 10 instances
Locator1.MaximumInstanceCount = 10
' Learn the models
Locator1.Learn
' Start the search process thread
Locator1.Start
' Reset the instance index
InstanceIndex = 0
' Wait until 1 instance is found
Do While ( Wait( 1, -1, Running ) > 0 )
    ' Retrieve the results for the located instance
    InstanceType = Locator1.InstanceModel(InstanceIndex)
    X = Locator1.InstanceTranslationX(InstanceIndex)
    Y = Locator1.InstanceTranslationY(InstanceIndex)
    If InstanceType = 0

```

```

        ' Pick the instance on the conveyor
        PickInstance( X, Y )
        ' Increment the instance index
        InstanceIndex = InstanceIndex + 1
    Else
        ' A part of the incorrect type has been found
        Locator1.Stop
        IncorrectType = True
        Exit Do
    End if
Loop

```

Unlearn

Description

Clears all the internal structures used by the Locator, including the learned models. The next execution of the Locator will be longer because the models must be re-learned.

Syntax

control.**Unlearn**

Parameter(s)

None

Return

None

Example

```

' Search the image
Locator1.Execute
' Free all the memory used by the Locator
Locator1.Unlearn

```

Wait

Description

Blocks the main thread of the application until the search process thread initiated with the **Start** method has found a given number of instances or a given timeout is reached. The timeout parameter of the **Wait** method must not be confused with the Locator's **Timeout** property. When the wait timeout is reached and no instances are found yet, the call will return, but the search process thread will continue until the number of instances to find is reached or the timeout set with the **Timeout** property is reached. The Running parameter of the **Wait** method can be used to check if the search process is still active.

Syntax

control.**Wait**(InstanceCount As Integer, Timeout As Integer, Running As Boolean) As Integer

Parameter(s)

Name	Description	Type	Range
<i>InstanceCount</i>	The number of instances to wait for.	Integer	0: Wait for completion of the search process. Greater than 0: Wait until a given number of instances are found
<i>Timeout</i>	A timeout after which the call to Wait will return.	Integer	-1: The call will not return until InstanceCount is reached or the search process is complete. 0: The call will return immediately. Greater than 0: The call will return when Timeout is reached.
<i>Running</i>	A Boolean variable passed by reference. Set to True if the search process thread is still running after the call to wait returns.	Boolean	True: The search process is still running. False: The search process is complete.

Return

Type	Description	Range
Integer	Returns the actual number of instances located during the wait period. This number never exceeds the value of the <i>InstanceCount</i> parameter.	Greater than or equal to 0

Example

```

Dim Running As Boolean
Dim AbortFlag As Boolean
Dim InstancesFound As Integer
Dim i As Integer
Dim X As Single
Dim Y As Single
' Learn the models
Locator1.Learn
' Start the search process thread
Locator1.Start
' Loop while the search process thread is running
Do
    Wait( 0, 10, Running )
    ' The user has decided to abort the search
    If AbortFlag Then

```

```

        Locator1.Stop
        Running = False
    End If
Loop While Running
' Process the instances
InstancesFound = Locator1.InstanceCount
For i = 0 To InstancesFound - 1
    ' Retrieve the results for the located instance
    X = Locator1.InstanceTranslationX(i)
    Y = Locator1.InstanceTranslationY(i)
    ' Pick the instance on the conveyor
    PickInstance( X, Y )
Loop

```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

HSModelEditorInterface

Description

The HSModelEditorInterface control provides programmatic access to the Model Editor dialog of the Locator. It can be used to open the dialog and configure model teaching parameters to propose default values customized for a specific application. A Model can also be built programmatically. The Model Editor dialog can also be hidden so that the entire model teaching procedure can be carried out without having to show the interactive user interface.

Properties

AutomaticLevels

Description

When set to True, this property specifies that the coarseness levels used for contour detection will be automatically optimized while building the model. When False, the contour coarseness levels must be configured manually before building the model using the **OutlineLevel** and **DetailLevel** properties.

Syntax

control.**AutomaticLevels** [= *Value* As Boolean]

Type

Boolean

Range

True, False

Example

`ModelEditor.AutomaticLevels = True`

BoundingBoxBottom

Description

The Y position in the image of the bottom of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used in the Model Editor only as an indication of the image area in which the contours used to construct the model will be detected. **BoundingBoxBottom** is expressed in pixels.

Syntax

control.**BoundingBoxBottom** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0.

Maximum: height of the image.

Example

`ModelEditor.BoundingBoxLeft = 100`

```

ModelEditor.BoundingBoxAreaRight = 540
ModelEditor.BoundingBoxAreaBottom = 50
ModelEditor.BoundingBoxAreaTop = 430
ModelEditor.BuildModel

```

BoundingBoxAreaLeft

Description

The X position in the image of the left side of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used in the Model Editor only as an indication of the image area in which the contours used to construct the model will be detected. **BoundingBoxAreaLeft** is expressed in pixels.

Syntax

control.**BoundingBoxAreaLeft** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0.

Maximum: width of the image.

Example

```

ModelEditor.BoundingBoxAreaLeft = 100
ModelEditor.BoundingBoxAreaRight = 540
ModelEditor.BoundingBoxAreaBottom = 50
ModelEditor.BoundingBoxAreaTop = 430
ModelEditor.BuildModel

```

BoundingBoxAreaRight

Description

The X position in the image of the right side of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used in the Model Editor only as an indication of the image area in which the contours used to construct the model will be detected. **BoundingBoxAreaRight** is expressed in pixels.

Syntax

control.**BoundingBoxAreaRight** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0.

Maximum: width of the image.

Example

```

ModelEditor.BoundingBoxAreaLeft = 100
ModelEditor.BoundingBoxAreaRight = 540
ModelEditor.BoundingBoxAreaBottom = 50
ModelEditor.BoundingBoxAreaTop = 430

```



```
ModelEditor.BuildModel
```

BoundingBoxAreaTop

Description

The Y position in the image of the top of the model's bounding box. Unlike other parameters of a model such as its origin or reference points, the bounding box is not a calibrated property. It is used in the Model Editor only as an indication of the image area in which the contours used to construct the model will be detected. **BoundingBoxAreaTop** is expressed in pixels.

Syntax

```
control.BoundingBoxAreaTop [= Value As Integer]
```

Type

Integer

Range

Minimum: 0.

Maximum: height of the image.

Example

```
ModelEditor.BoundingBoxAreaLeft = 100
ModelEditor.BoundingBoxAreaRight = 540
ModelEditor.BoundingBoxAreaBottom = 50
ModelEditor.BoundingBoxAreaTop = 430
ModelEditor.BuildModel
```

ContrastThreshold

Description

Defines the minimum contrast needed for a contour to be detected in the model image. This threshold is expressed in terms of step in greylevel values. This property can only be set when **ContrastThresholdMode** is set to **hsContrastThresholdFixedValue**. Read only otherwise.

Syntax

```
control.ContrastThreshold [= Value As Integer]
```

Type

Integer

Range

Minimum: 1

Maximum: 255

Example

```
ModelEditor.ContrastThresholdMode = _
    hsContrastThresholdModeFixedValue
ModelEditor.ContrastThreshold = 10
```

ContrastThresholdMode

Description

Selects the method used to compute the threshold used for detecting contours in the model image.

Syntax

```
control.ContrastThresholdMode [= Value As hsContrastThresholdMode]
```

Type

hsContrastThresholdMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContrastThresholdAdaptiveLowSensitivity	0	Uses a low sensitivity adaptive threshold for detecting contours.
hsContrastThresholdAdaptiveNormalSensitivity	1	Uses a normal sensitivity adaptive threshold for detecting contours.
hsContrastThresholdAdaptiveHighSensitivity	2	Uses a high sensitivity adaptive threshold for detecting contours.
hsContrastThresholdFixedValue	3	Uses a fixed value threshold for detecting contours.

Example

```
ModelEditor.ContrastThresholdMode = hsContrastThresholdModeFixedValue
ModelEditor.ContrastThreshold = 10
```

Database

Description

Reserved for internal use.

DetailLevel

Description

The coarseness of the contours used to build the model at the Detail Level.

Syntax*control*.**DetailLevel** [= *Value* As Integer]**Type**

Integer

Range

Minimum: 1

Maximum: 8

Example

```
ModelEditor.AutomaticContourDetection = False
ModelEditor.OutlineLevel = 8
ModelEditor.DetailLevel = 1
ModelEditor.BuildModel
```

DisplayCalibrationMode

Description

Calibration mode used to display the model in the Model Editor.

Syntax*control*.**DisplayCalibrationMode** [= *Value* As hsDisplayCalibrationMode]**Type**

hsEditorDisplayCalibrationMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsDisplayUncalibrated	0	The model can only be displayed in the uncalibrated mode. The Calibrated button is not shown in the display toolbar.
hsDisplayCalibrated	1	The model can only be displayed in the calibrated mode. The Calibrated button is not shown in the display toolbar.
hsDisplayUserSelected	2	The model can be displayed in the uncalibrated or calibrated mode by clicking on the Calibrated button in the display toolbar.

Example

```
' Deactivate the calibrated button and show the model
' in the uncalibrated mode
ModelEditor.DisplayCalibrationMode = hsEditorDisplayUncalibrated
```

FeatureSelection

Description

This property controls the amount of features that will be selected from the detected contours at both the Outline and Detail levels. At its normal setting, the optimal amount of features are selected to build the model. Increasing **FeatureSelection** will add more features to the model. To manually select the features of the model using the interactive Model Editor, **FeatureSelection** can be set to **none** so that no features will be automatically selected after the call to **BuildModel**.

Syntax

```
control.FeatureSelection [= Value As hsModelFeatureSelection]
```

Type

```
hsModelFeatureSelection
```

Range

The valid range for this property is as follows:

Constant	Value	Description
hsFeatureSelectionNone	0	No features are selected from the contours.
hsFeatureSelectionLess	1	Fewer features than the optimal set will be selected from the contours.
hsFeatureSelectionNormal	2	The optimal features are selected from the contours
hsFeatureSelectionMore	3	More features than the optimal set will be selected from the contours.
hsFeatureSelectionAll	4	All of the contours are selected as features.

Example

```
ModelEditor.AutomaticContourDetection = False
ModelEditor.FeatureSelection = hsFeatureSelectionNormal
ModelEditor.BuildModel
```

IconVisible

Description

Controls whether the window icon of the Model Editor is visible or invisible.

Syntax

`control.IconVisible [= Value As Boolean]`

Type

Boolean

Range

Not applicable.

Example

```
' Hide the Model Editor's windows icon
ModelEditor.IconVisible = False
```

ImageOriginY

Description

Note: This property is obsolete and is supported only for compatibility with applications built in previous versions. To set the reference frame that used to express image coordinates, use the **ImageCoordinateSystemType** property of the *HSImage* object.

MessageCount

Description

The number of messages issued during the last call to **BuildModel** or **AnalyzeModel**. The identification number and description of each message is retrieved using respectively the **MessageNumber** and **MessageDescription** properties. Read only.

Syntax

`control.MessageCount`

Type

Integer

Range

Minimum: 0

Maximum: Unlimited

Example

```
Dim m As Integer
For m = 0 To ModelEditor.MessageCount - 1
    Description.Text = ModelEditor.MessageDescription(m)
    Number.Text = Str(ModelEditor.MessageNumber(m))
Next m
```

MessageDescription

Description

The description of a message issued during the last call to **BuildModel** or **AnalyzeModel**. The identification number of the message corresponding to this description is retrieved using the **MessageNumber** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageDescription**(*Index* As Integer)

Type

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```
Dim m As Integer
For m = 0 To ModelEditor.MessageCount - 1
    Description.Text = ModelEditor.MessageDescription(m)
    Number.Text = Str(ModelEditor.MessageNumber(m))
Next m
```

MessageNumber

Description

The identification number of a message issued during the last call to **BuildModel** or **AnalyzeModel**. The description of the message corresponding to this identification number is retrieved using the **MessageDescription** property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageNumber**(*Index* As Integer)

Type

Integer

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```
Dim m As Integer
For m = 0 To ModelEditor.MessageCount - 1
    Description.Text = ModelEditor.MessageDescription(m)
    Number.Text = Str(ModelEditor.MessageNumber(m))
Next m
```

ModelEdgePathCount

Description

Number of Model edge paths in the Model contour scene. Read only. A model edge path is a group of connected edge elements in a Model contour scene. It is a structure used to select features from contours detected in a Model image. The model edge path selection structure is used to implement interactive Model edition functionalities. As opposed to a regular edge path, the edge elements that form a model edge path are not necessarily stored sequentially in the scene's edge elements array. A given model edge path *i* always starts at index

Scene.ModelEdgePathFirstEdgeIndex(*i*) and ends at index

Scene.ModelEdgePathLastEdgeIndex(*i*).

Syntax

control.**ModelEdgePathCount**(ModelScene As hsModelScene)

Type

Long

Range

Greater than or equal to 0.

Parameter(s)

Parameter	Description	Type	Range
<i>ModelScene</i>	Selected Model contour scene.	hsModelScene	See table below.
<i>ModelEdgePathIndex</i>	Index of the Model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount (ModelScene) -1

Example

See example from **HSScene** at **ModelEdgePathFirstEdgeIndex**.

ModelEdgePathLocation

Description

Specifies if the selected Model edge path is enabled for Location.

Syntax

control.**ModelEdgePathLocation**(ModelScene As hsModelScene, ModelEdgePathIndex As Long) [= Value As Boolean]

Type

Boolean

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ModelScene</i>	Selected Model contour scene.	hsModelScene	See table below.
<i>ModelEdgePathIndex</i>	Index of the Model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount (ModelScene) -1

Constant	Value	Description
hsModelSceneOutline	0	Model outline contour scene.
hsModelSceneDetail	1	Model outline detail scene.

Example

```
Dim UsedForLocation As Boolean
UsedForLocation = ModelEditor.ModelEdgePathLocation(hsModelSceneDetail,0)
ModelEditor.ModelEdgePathLocation(hsModelSceneOutline,0) = False
```

ModelEdgePathRequired

Description

Specifies if the selected Model edge path is enabled as Required.

Syntax

```
control.ModelEdgePathRequired(ModelScene As hsModelScene, ModelEdgePathIndex As Long) [= Value As Boolean]
```

Type

Boolean

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>ModelScene</i>	Selected Model contour scene.	hsModelScene	See table below.
<i>ModelEdgePathIndex</i>	Index of the Model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount (ModelScene) -1

Constant	Value	Description
hsModelSceneOutline	0	Model outline contour scene.
hsModelSceneDetail	1	Model outline detail scene.

Example

```
Dim Required As Boolean
Required = ModelEditor.ModelEdgePathRequired(hsModelSceneDetail,0)
ModelEditor.ModelEdgePathRequired(hsModelSceneOutline,0) = True
```

ModelImage

Description

ModelImage provides access to the HSImage object that contains the Model image. Read only.

Syntax

```
control.ModelImage
```

Type

HSImage

Range

Not applicable.

Example

```
Dim ModelImage As HSImage
Set ModelImage = ModelEditor.ModelImage
```

ModelMask

Description

ModelMask provides access to the HSImage object that contains the Mask image. Read only. A Mask must be built programmatically with HSImage object drawing methods. The mask is initially created by the **CreateMask** method and is destroyed by calling the **RemoveMask** method.

The purpose of the Mask is to exclude parts of an image from the model-building process. Pixels with a value greater than 0 are marked as masked and pixels with a value of 0 are marked as unmasked. When the **BuildModel** command is executed, all masked pixels are excluded from the process: no edges are detected for the masked pixels. The mask image is saved in the Model Database and is used only for model generation. Although a mask can only be created and edited programmatically, if a mask image exists for a model, it will be displayed into the Model Editor interface. When changes are made to a mask, calling **RefreshDisplay** will update the Model Editor display to show the modified Mask.

Syntax*control*.**ModelMask****Parameter(s)**

Not applicable

Type

HSImage

Example

```
' Show Model Editor and Add Mask Image
Dim Locator As HSLocator
Dim ModelInterface As HSModelEditorInterface
Set Locator = ApplicationControl.ProcessManager.Process("Locator")
If Not Locator Is Nothing Then
    Set ModelInterface = Locator.ShowModelEditor(True, 0)
If Not ModelInterface Is Nothing Then
    ModelInterface.CreateMask
End If

' Draw Mask Image
Dim ModelMask As HSImage
Set ModelMask = ModelInterface.ModelMask

If Not ModelMask Is Nothing Then
    ModelMask.BackgroundColor = 1
```



```

    ModelMask.AddRectangleShape 100, 100, 300, 300, 0
    ModelMask.AddCircleShape 400, 250, 50, 0
    ModelMask.Render
    ModelInterface.RefreshDisplay
End If

```

ModelSceneDetail

Description

ModelSceneDetail provides access to the HSScene object that contains the Model Detail Contour Scene. Read only.

Syntax

control.**ModelSceneDetail**

Type

HSScene

Range

Not applicable.

Example

```

Dim DetailScene as HSScene
Set DetailScene = ModelEditor.ModelSceneDetail

```

ModelSceneOutline

Description

ModelSceneOutline provides access to the HSScene object that contains the Model Outline Contour Scene. Read only.

Syntax

control.**ModelSceneOutline**

Type

HSScene

Range

Not applicable.

Example

```

Dim OutlineScene as HSScene
Set OutlineScene = ModelEditor.ModelSceneOutline

```

Name

Description

The name of the currently model edited. Read only.

Syntax

control.**Name**

Type

String

Range

Not applicable.

Example

```
Dim modelName As String
modelName = ModelEditor.Name
```

OriginPositionX

Description

The X position in the World coordinate system of the model's origin. The origin, defined by the **OriginPositionX**, **OriginPositionY**, and **OriginRotation** properties, is used by the Locator to express the pose of instances of the model. The translation of the instance represents the position of the model's origin in the World coordinate system, as defined by the calibration of the input image. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin also defines the Object coordinate system that can be used to express results of model-based inspection tools. **OriginPositionX** is expressed in the calibrated units specified by the **UnitsLength** property of the associated Locator.

Syntax

```
control.OriginPositionX [= Value As Single]
```

Type

Single

Range

Not applicable.

Example

```
ModelEditor.OriginPositionX = 235.5
ModelEditor.OriginPositionY = 127.2
ModelEditor.OriginRotation = 45.0
ModelEditor.Apply
```

OriginPositionY

Description

The Y position in the World coordinate system of the model's origin. The origin, defined by the **OriginPositionX**, **OriginPositionY**, and **OriginRotation** properties, is used by the Locator to express the pose of instances of the model. The translation of the instance represents the position of the model's origin in the World coordinate system, as defined by the calibration of the input image. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin also defines the Object coordinate system that can be used to express results of model-based inspection tools. **OriginPositionY** is expressed in the calibrated units specified by the **UnitsLength** property of the associated Locator.

Syntax

```
control.OriginPositionY [= Value As Single]
```

Type

Single

Range

Not applicable.

Example

```
ModelEditor.OriginPositionX = 235.5
ModelEditor.OriginPositionY = 127.2
```

```
ModelEditor.OriginRotation = 45.0
ModelEditor.Apply
```

OriginRotation

Description

The rotation in the World coordinate system of the model's origin. The origin, defined by the **OriginPositionX**, **OriginPositionY**, and **OriginRotation** properties, is used by the Locator to express the pose of instances of the model. The translation of the instance represents the position of the model's origin in the World coordinate system, as defined by the calibration of the input image. The model's origin is also used as the pivot point around which the rotation of the instance is measured. This origin also defines the Object coordinate system that can be used to express results of model-based inspection tools. **OriginRotation** is expressed in the orientation units specified by the **UnitsOrientation** property of the associated Locator.

Syntax

control.**OriginRotation** [= Value As Single]

Type

Single

Range

Minimum: -180.0

Maximum: 180.0

Example

```
ModelEditor.OriginPositionX = 235.5
ModelEditor.OriginPositionY = 127.2
ModelEditor.OriginRotation = 45.0
ModelEditor.Apply
```

OutlineLevel

Description

The coarseness of the contours used to build the model at the Outline level. Read only when the **AutomaticLevels** property is True.

Syntax

control.**OutlineLevel**

Type

Integer

Range

Minimum: 1

Maximum: 8

Example

```
ModelEditor.AutomaticContourDetection = False
ModelEditor.OutlineLevel = 8
ModelEditor.DetailLevel = 1
ModelEditor.BuildModel
```

ReferencePointCount

Description

Number of reference points defined in the current Model. Read only.

Syntax

control.**ReferencePointCount**

Type

Integer

Range

Greater than or equal to 0.

Example

```
Dim ReferencePointCount As Integer
ReferencePointCount = ModelEditor.ReferencePointCount
```

ReferencePointPositionX

Description

X coordinate of the selected Reference Point, expressed in calibrated units. Read only.

Syntax

control.**ReferencePointPositionX**(*ReferencePointIndex* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>ReferencePointIndex</i>	Index of the reference point.	Integer	Minimum: 0 Maximum: ReferencePointCount - 1

Example

```
Dim XPos As Single
Xpos = ModelEditor.ReferencePointPositionX(0)
```

ReferencePointPositionY

Description

Y coordinate of the selected Reference Point, expressed in calibrated units. Read only.

Syntax

control.**ReferencePointPositionY**(*ReferencePointIndex* As Integer)

Type

Single

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
<i>ReferencePointIndex</i>	Index of the reference point.	Integer	Minimum: 0 Maximum: ReferencePointCount - 1

Example

```
Dim YPos As Single
YPos = ModelEditor.ReferencePointPositionY(0)
```

ShadingAreaBottom

Description

Sets the Y position of the bottom of the shading area. A custom shading area bounded by **ShadingAreaBottom**, **ShadingAreaLeft**, **ShadingAreaRight** and **ShadingAreaTop** is used for shading consistency analysis when the **InstanceOrdering** property of the Locator is set to **hsShadingConsistency** and the **UseCustomShadingArea** is set to True.

Syntax

```
control.ShadingAreaBottom [=Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Height of the Model image.

Example

```
ModelEditor.ShadingAreaLeft = 100
ModelEditor.ShadingAreaRight = 540
ModelEditor.ShadingAreaBottom = 50
ModelEditor.ShadingAreaTop = 430
```

ShadingAreaLeft

Description

Sets the X position of the left side of the shading area. A custom shading area bounded by **ShadingAreaBottom**, **ShadingAreaLeft**, **ShadingAreaRight** and **ShadingAreaTop** is used for shading consistency analysis when the **InstanceOrdering** property of the Locator is set to **hsShadingConsistency** and the **UseCustomShadingArea** is set to True.

Syntax

```
control.ShadingAreaLeft [=Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Width of the Model image.

Example

```
ModelEditor.ShadingAreaLeft = 100
ModelEditor.ShadingAreaRight = 540
```

```
ModelEditor.ShadingAreaBottom = 50
ModelEditor.ShadingAreaTop = 430
```

ShadingAreaRight

Description

Sets the X position of the right side of the shading area. A custom shading area bounded by **ShadingAreaBottom**, **ShadingAreaLeft**, **ShadingAreaRight** and **ShadingAreaTop** is used for shading consistency analysis when the **InstanceOrdering** property of the Locator is set to **hsShadingConsistency** and the **UseCustomShadingArea** is set to True.

Syntax

control.**ShadingAreaRight** [=Value As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Width of the Model image.

Example

```
ModelEditor.ShadingAreaLeft = 100
ModelEditor.ShadingAreaRight = 540
ModelEditor.ShadingAreaBottom = 50
ModelEditor.ShadingAreaTop = 430
```

ShadingAreaTop

Description

Sets the Y position of the top of the shading area. A custom shading area bounded by **ShadingAreaBottom**, **ShadingAreaLeft**, **ShadingAreaRight** and **ShadingAreaTop** is used for shading consistency analysis when the **InstanceOrdering** property of the Locator is set to **hsShadingConsistency** and the **UseCustomShadingArea** is set to True.

Syntax

control.**ShadingAreaTop** [=Value As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Height of the Model image.

Example

```
ModelEditor.ShadingAreaLeft = 100
ModelEditor.ShadingAreaRight = 540
ModelEditor.ShadingAreaBottom = 50
ModelEditor.ShadingAreaTop = 430
```

ShowBars

Description

When **ShowBars** is set to True, the display's tool bar, scroll bar, status bar and rulers are visible in the Model Editor.

Syntax

control.**ShowBars** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

ModelEditor.ShowBarse = True

ShowImage

ShowGreyScaleImage

Description

When **ShowImage** is set to True, the image is shown in background in the display of the Model Editor.

Syntax

control.**ShowImage** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

ModelEditor.ShowImage = True

ShowImageMode

Description

Mode used to show the image in the Model Editor display. With the default normal setting the image is shown using its actual brightness level. The brightness of the image in the display can be respectively increased or decreased with the bright and dim settings. The **ShowImageMode** property only affect the appearance of the image in the display. It does not modify the actual greylevel values of the image.

Syntax

control.**ShowImageMode** [= *Value* As hsShowImageMode]

Type

hsShowImageMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsShowImageModeNormal	0	Image is displayed using its normal brightness level.
hsShowImageModeDim	1	Image is displayed with a reduced brightness level.
hsShowImageModeBright	2	Image is displayed with an increased brightness level.

Example

ModelEditor.ShowImageMode

TrackingInertia

Description

The inertia of the edge tracking phase where individual edge elements are connected to form the contours from which the model features are selected. It corresponds to the maximum distance, in pixels, between two edge elements that can be connected to form a continuous contour.

Syntax

control.**TrackingInertia** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: 1

Example

```
ModelEditor.TrackingInertia = 1
ModelEditor.BuildModel
```

UseCustomShadingArea

Description

When **UseCustomShadingArea** is set to True, shading consistency analysis is carried out on the custom shading area defined by **ShadingAreaBottom**, **ShadingAreaLeft**, **ShadingAreaRight** and **ShadingAreaTop**. When False, shading consistency analysis is carried out on the entire Model boundary area. Shading consistency analysis is carried out when the **InstanceOrdering** property of the **Locator** is set to **hsShadingConsistency**.

Syntax

control.**UseCustomShadingArea** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
ModelEditor.UseCustomShadingArea = True
```

Methods

AddModelEdgePath

Description

Adds a new Model edge path from specified parameters. The specified edgel indexes must come from the same **Model Scene Edge Path(i)**.

Syntax

control.**AddModelEdgePath**(*ModelScene* As hsModelScene, *FirstEdgelIndex* As Long, *LastEdgelIndex* As Long) As Long

Parameter(s)

Parameter	Description	Type	Range
<i>ModelScene</i>	Selected Model contour scene.	hsModelScene	See table below.
FirstEdgeIndex	Index of the first edgel.	Long	Minimum: Scene.EdgePathFirstEdgeIndex(i) Maximum: Scene.EdgePathLastEdgeIndex(i)
LastEdgeIndex	Index of the last edgel.	Long	Minimum: Scene.EdgePathFirstEdgeIndex(i) Maximum: Scene.EdgePathLastEdgeIndex(i)

Constant	Value	Description
hsModelSceneOutline	0	Model outline contour scene.
hsModelSceneDetail	1	Model outline detail scene.

Return

Long

Range

Not applicable

Example

```
ModelEditor.AddModelEdgePath(hsModelSceneOutline, 0, 10)
```

AddReferencePoint

Description

Adds a new Reference Point to the currently edited model.

Syntax

```
control.AddReferencePoint (X As Single, Y As Single) As Integer
```

Parameter(s)

Parameter	Description	Type	Range
X	X coordinate of the reference point	Single	Not applicable.
Y	Y coordinate of the reference point	Single	Not applicable.

Return

Integer

Range

Not applicable.

Example

```
ModelEditor.AddReferencePoint(100.0, 200.0)
```

AnalyzeModel

Description

Analyzes the current model. The result of this analysis is retrieved using the **MessageCount**, **MessageNumber**, and **MessageDescription** properties.

Syntax

control.**AnalyzeModel**

Parameter(s)

None

Return

No return value.

Example

ModelEditor.**AnalyzeModel**

Apply

Description

Copies the edited model to the current models database of the Locator, without closing the Model Editor. Calling **Apply** does not implicitly save the modified models database file. To do so, the Locator's **SaveModelDatabase** method must be used.

Syntax

control.**Apply**

Parameter(s)

None

Return

No return value.

Example

ModelEditor.**BuildModel**

ModelEditor.**Apply**

BuildModel

Description

Builds a new model using the image area delimited by the **BoundingBoxBottom**, **BoundingBoxTop**, **BoundingBoxLeft**, and **BoundingBoxRight** properties. If **AutomaticLevels** is True, the contour detection parameters are optimized, otherwise they are used as is to detect the contours at both the Outline and Detail levels. If **FeatureSelection** is set to a value other than none, features are selected from the detected contours. An analysis of the newly built model can be retrieved using the **MessageCount**, **MessageNumber**, and **MessageDescription** properties.

Syntax

control.**BuildModel**

Parameter(s)

None

Return

No return value.

Remark(s)

If a mask exists for the model, **BuildModel** will excluded all masked pixels. No edges are detected for the masked pixels. The mask image is saved in the Model Database and is used only for model generation. Although a mask can only be created and edited programmatically, if a mask image exists for a model, it will be displayed into the Model Editor interface. When changes are made to a mask, calling **RefreshDisplay** will update the Model Editor display to show the modified Mask.

Example

```
ModelEditor.BuildModel
```

CreateMask

Description

CreateMask creates a Mask image in which, by default, every pixel is set to 0. If successful, **CreateMask** returns an HSImage object in which the mask image can be drawn. The mask can be destroyed by the **RemoveMask** method. The **ModelEdgePathRequired** property returns the HSImage object in which the Mask image is created by means of the HSImage's object drawing capabilities.

The Mask is an 8-bit grey-scale image that is used to exclude parts of an image from the model-building process. Pixels with a value greater than 0 are marked as masked and pixels with a value of 0 are marked as unmasked. When the **BuildModel** command is executed, all masked pixels are excluded from the process: no edges or features are created from the masked pixels. The mask image is saved in the Model Database and is used only for model generation. Although a mask can only be created and edited programmatically, if a mask image exists for a model, it will be displayed into the Model Editor interface. When changes are made to a mask, calling **RefreshDisplay** will update the Model Editor display to show the modified Mask.

Syntax

```
control.CreateMask()
```

Type

Boolean

Range

Not applicable.

Example

```
' Show Model Editor and Add Mask Image
Dim Locator As HSLocator
Dim ModelInterface As HSModelEditorInterface
Set Locator = ApplicationControl.ProcessManager.Process("Locator")
If Not Locator Is Nothing Then
    Set ModelInterface = Locator.ShowModelEditor(True, 0)
If Not ModelInterface Is Nothing Then
    ModelInterface.CreateMask
End If

' Draw Mask Image
Dim ModelMask As HSImage
```

```
Set ModelMask = mModelInterface.ModelMask
```

```
If Not ModelMask Is Nothing Then
    ModelMask.BackgroundColor = 1
    ModelMask.AddRectangleShape 100, 100, 300, 300, 0
    ModelMask.AddCircleShape 400, 250, 50, 0
    ModelMask.Render
    ModelInterface.RefreshDisplay
End If
```

CropModel

Description

Crops the image according to the bounding area used to build the model.

Syntax

control.**CropModel**

Parameter(s)

None

Return

No return value.

Example

Locator1.**CropModel**

EndDialog

Description

Terminates the Model Editor dialog. Depending on the value of the **Result** parameter, the modified model is either discarded or copied to the current models database of the Locator. Calling **EndDialog** does not implicitly save the modified models database file. To do so, the Locator's **SaveModelDatabase** method must be used.

Syntax

control.**EndDialog**(*Result* As hsModelDialogResult)

Parameter(s)

Parameter	Description	Type	Range
<i>Result</i>	Specifies whether the model will be discarded or copied to the current models database.	hsModelDialogResult	Enum. See table below.

hsModelDialogResult	Value	Description
hsResultCancel	2	The model is discarded
hsResultOk	1	The model is copied to the current models database

Return

No return value.

Example

```

If DiscardModel = True Then
    ModelEditor.EndDialog hsResultCancel
Else
    ModelEditor.EndDialog hsResultOk
End If

```

GetDispatchPointer

Description

Reserved for internal use.

Hide

Description

Hides the Model Editor dialog. Unlike **EndDialog**, calling **Hide** hides the dialog but leaves it active without copying the model to the current models database. To terminate the dialog, **EndDialog** must be called.

Syntax

control.**Hide**

Parameter(s)

None

Return

No return value.

Example

```

ModelEditor.Hide

```

RefreshDisplay

Description

Updates the display in the Model Editor interface, to show any changes that have been made to the model, including the addition of a mask or changes made to an existing mask.

Syntax

control.**RefreshDisplay**

Parameter(s)

None

Return

No return value.

Range

Not applicable.

Example

```

' Show Model Editor and Add Mask Image
Dim Locator As HSLocator
Dim ModelInterface As HSModelEditorInterface
Set Locator = ApplicationControl.ProcessManager.Process("Locator")
If Not Locator Is Nothing Then
    Set ModelInterface = Locator.ShowModelEditor(True, 0)

```

```

If Not ModelInterface Is Nothing Then
    ModelInterface.CreateMask
End If

' Draw Mask Image
Dim ModelMask As HSImage
Set ModelMask = mModelInterface.ModelMask

If Not ModelMask Is Nothing Then
    ModelMask.BackgroundColor = 1
    ModelMask.AddRectangleShape 100, 100, 300, 300, 0
    ModelMask.AddCircleShape 400, 250, 50, 0
    ModelMask.Render
    ModelInterface.RefreshDisplay
End If

```

RemoveModelEdgePath

Description

Destroys an existing Model edge path.

Syntax

control.**RemoveModelEdgePath**(*ModelScene* As hsModelScene, *ModelEdgePathIndex* As Long) As Long

Parameter(s)

Parameter	Description	Type	Range
<i>ModelScene</i>	Selected Model contour scene.	hsModelScene	See table below.
<i>ModelEdgePathIndex</i>	Index of the Model edge path.	Long	Minimum: 0 Maximum: ModelEdgePathCount (ModelScene) -1

Constant	Value	Descriptions
hsModelSceneOutline	0	Model outline contour scene.
hsModelSceneDetail	1	Model outline detail scene.

Return

No return value.

Example

```
ModelEditor.RemoveModelEdgePath(hsModelSceneOutline, 0)
```

RemoveReferencePoint

Description

Destroys an existing Reference Point.

Syntax

control.**RemoveReferencePoint** (*ReferencePointIndex* As Integer)

Parameter(s)

Parameter	Description	Type	Range
ReferencePointIndex	Index of the reference point.	Integer	Minimum: 0 Maximum: ReferencePointCount - 1.

Return

No return value.

Example

```
ModelEditor.RemoveReferencePoint(2)
```

RemoveMask

Description

RemoveMask destroys an existing Mask image. For more detail on Masks see the **ModelMask** property and **CreateMask** method.

Syntax

control.**RemoveMask**()

Type

Boolean

Range

Not applicable.

Example

```
' Remove Mask
If Not ModelInterface Is Nothing Then
    ModelInterface.RemoveMask
    ModelInterface.RefreshDisplay
End If
```

Revert

Description

Undoes all modifications made to the model since the last call to **Apply**.

Syntax

control.**Revert**

Parameter(s)

None

Return

No return value.

Example

```
Locator1.Revert
```

Show

Description

Shows the Model Editor interface in a modeless window. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameter settings can then be changed before making the dialog visible by calling **Show**. To show the Model Editor interface in a modal window, use the **ShowModal** method.

Syntax

control.**Show**

Parameter(s)

None

Return

No return value.

Example

```
Dim ModelEditor As HSModelEditorInterface
' Create a new model
Locator1.AddModel("MyModel")
' Create an invisible Model Editor
ModelEditor = Locator1.ShowModelEditor(False, "MyModel")
' Adjust the parameters and build the model
ModelEditor.FeatureSelection = hsFeatureSelectionAll
ModelEditor.BuildModel
' Show the Model Editor in a modeless window
ModelEditor.Show
```

ShowModal

Description

Shows the Model Editor interface in a modal window. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameter settings can then be changed before making the dialog visible by calling **ShowModal**. To show the Model Editor interface in a modeless window, use the **Show** method.

Syntax

control.**ShowModal**

Parameter(s)

None

Return

No return value.

Example

```
Dim ModelEditor As HSModelEditorInterface
' Create a new model
Locator1.AddModel("MyModel")
' Create an invisible Model Editor
ModelEditor = Locator1.ShowModelEditor(False, "MyModel")
' Adjust the parameters and build the model
ModelEditor.FeatureSelection = hsFeatureSelectionAll
```



```

ModelEditor.BuildModel
' Show the Model Editor in a modal window
ModelEditor.ShowModal

```

Events

AnalyzeModelClick

Description

Occurs when the Analyze Model button is clicked in the Model Editor.

Parameter(s)

None

ApplyClick

Description

Occurs when the Apply button is clicked in the Model Editor.

Parameter(s)

None

BoundingBoxChange

Description

Occurs when the bounding area is changed in the Model Editor.

Parameter(s)

None

BuildModelClick

Description

Occurs when the Build Model button is clicked in the Model Editor.

Parameter(s)

None

CancelClick

Description

Occurs when the Cancel button is clicked in the Model Editor.

Parameter(s)

None

ContourDetectionParametersChange

Description

Occurs when the contour detection parameters are changed in the Model Editor.

Parameter(s)

None

CoordinateSystemChange

Description

Occurs when the Object coordinate system is changed in the Model Editor.

Parameter(s)

None

FeatureSelectionChange

Description

Occurs when features are added or deleted in the Model Editor.

Parameter(s)

None

FeatureSelectionSliderChange

Description

Occurs when the feature selection slider is changed in the Model Editor.

Parameter(s)

None

OKClick

Description

Occurs when the OK button is clicked in the Model Editor.

Parameter(s)

None

ReferencePointsChange

Description

Occurs when reference points are changed in the Model Editor.

Parameter(s)

None

HSMModelOptimizerInterface

Description

The HSMModelOptimizerInterface control provides programmatic access to the Model Optimizer dialog of the Locator. It can be used to open the dialog and optimize a Model. This can also be done programmatically. The Model Optimizer dialog can also be hidden so that the entire model optimizing procedure can be carried out without having to show the interactive user interface.

Properties

Database

Description

Reserved for internal use.

DisplayCalibrationMode

Description

Calibration mode used to display the model in the Model Optimizer.

Syntax

control.**DisplayCalibrationMode** [= *Value* as hsDisplayCalibrationMode]

Type

Integer

Range

The valid range for this property is as follows:

Constant	Value	Description
hsDisplayUncalibrated	0	The model can only be displayed in the uncalibrated mode. The Calibrated button is not shown in the display toolbar.
hsDisplayCalibrated	1	The model can only be displayed in the calibrated mode. The Calibrated button is not shown in the display toolbar.
hsDisplayUserSelected	2	The model can be displayed in the uncalibrated or calibrated mode by clicking on the Calibrated button in the display toolbar.

Example

```
ModelOptimizer.DisplayCalibrationMode = _
    hsDisplayUncalibrated
```

IconVisible

Description

Controls whether the window icon of the Model Optimizer is visible or invisible.

Syntax

control.**IconVisible** [= *Value* as Boolean]

Type

Boolean

Range

Not applicable.

Example

```
' Hide the Model Optimizer's windows icon
ModelOptimizer.IconVisible = False
```

InstancesAnalyzed

Description

Number of instances that have been processed and used to optimize the current model. Read only.

Syntax

control.**InstancesAnalyzed**

Type

Integer v

Range

Greater than or equal to zero.

Example

```
Dim InstancesAnalyzed as Integer
InstancesAnalyzed = ModelOptimizer1.InstancesAnalyzed
```

LinkOutlineWithDetail

Description

When **LinkOutlineWithDetail** is set to true, **MinimumRecurrenceOutline** and **MorphingOutline** are automatically set, based on the **MinimumRecurrenceDetail** and **MorphingDetail** properties. To set a value for **MinimumRecurrenceOutline**, **LinkOutlineWithDetail** must be set to False.

Syntax

control.**LinkOutlineWithDetail** [= Value as Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Optimizer as HSMModelOptimizerInterface
' Create invisible Model Optimizer
Optimizer = Locator.ShowModelOptimizer (False, "Model")
' Adjust parameters
Optimizer.LinkOutlineWithDetail = True
' Show the Model Optimizer in a modeless window
Optimizer.Show
```

MinimumRecurrenceDetail

Description

At the Detail level of the model to be optimized, **MinimumRecurrenceDetail** sets the minimum recurrence required for a feature to be considered a strong feature. Strong features are those that are retained as valid and used to build the optimized Model. This value represents a percentage; a value of 50 means that a feature will be retained for the optimized model if it occurs in 50% or more of the object instances retained for the optimization.

Syntax

control.**MinimumRecurrenceDetail** [= *Value* as Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: 100.0

Example

```
Dim OccurrenceThresholdDetail as Single
OccurrenceThresholdDetail = _
    Optimizer1.MinimumRecurrenceDetail
Optimizer1.MinimumRecurrenceDetail = 100.0
```

MinimumRecurrenceOutline

Description

At the Outline level of the model to be optimized, **MinimumRecurrenceOutline** sets the minimum frequency of recurrence that is required for a feature to be considered a strong feature. Strong features are those that are retained as valid and used to build the optimized Model. This value represents a percentage; a value of 50 means that a feature will be retained for the optimized model if it occurs in 50% or more of the object instances retained for the optimization.

Syntax

control.**MinimumRecurrenceOutline** [= *Value* as Single]

Type

Single

Range

Minimum: greater than 0. See Remarks.
Maximum: 100.0

Remark(s)

The value of this property can be determined by the **MinimumRecurrenceDetail** value, when **LinkOutlineWithDetail** is set to True. In such a case, the value of **MinimumRecurrenceOutline** is equal to: $\text{Max}([\text{MinimumRecurrenceDetail} - 10], 1)$.

Example

```
Dim OccurrenceThresholdOutline as Single
OccurrenceThresholdOutline = _
    Optimizer1.MinimumRecurrenceOutline
Optimizer2.MinimumRecurrenceOutline = 100.0
```

MorphingDetail

Description

When **MorphingDetail** is set to True, morphing is applied when optimizing Model features at the Detail Level. Morphing allows the optimization process to add recurring features to the optimized model even if these features were absent from the initial model proposed for the optimization process. These recurring features are appended to the model only if their frequency of recurrence is equal to or above the threshold set by **MinimumRecurrenceDetail**.

Syntax

control.**MorphingDetail** [= Value as Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled as Boolean
Enabled = ModelOptimizer1.MorphingDetail
ModelOptimizer1.MorphingDetail = False
```

MorphingOutline

Description

When **MorphingOutline** is set to True, morphing is applied to the model features at the Outline Level. Morphing allows the optimization process to add recurring features to the optimized model even if these features were absent from the initial model proposed for the optimization process. These recurring features are appended to the model only if their frequency of recurrence is equal to or above the threshold set by **MinimumRecurrenceOutline**.

Syntax

control.**MorphingOutline** [= Value as Boolean]

Type

Boolean

Range

Not applicable.

Remark(s)

The value of this property can be set by **MorphingDetail**. When **LinkOutlineWithDetail** is set to True, the value of **MorphingOutline** is the same as **MorphingDetail**.

Example

```
Dim Enabled as Boolean
Enabled = ModelOptimizer1.MorphingOutline
ModelOptimizer1.MorphingOutline = False
```

Name

Description

The name of the model currently optimized. Read only.

Syntax*control*.**Name****Type**

String

Range

Not applicable.

Example

```
Dim ModelName as String
ModelName = ModelOptimizer.Name
```

ShowBars

Description

When **ShowBars** is set to True, the display's tool bar, scroll bar, status bar and rulers are visible in the Model Optimizer.

Syntax*control*.**ShowBars** [= *Value* as Boolean]**Type**

Boolean

Range

Not applicable.

Example

```
ModelOptimizer.ShowBars = True
```

ShowImage

ShowGreyScaleImage

Description

When **ShowImage** is set to True, the image is shown in background in the display of the Model Optimizer.

Syntax*control*.**ShowImage** [= *Value* as Boolean]**Type**

Boolean

Range

Not applicable.

Example

```
ModelOptimizer.ShowImage = True
```

ShowImageMode

Description

Sets the mode used to show the image in the Model Optimizer display. With the default normal setting the image is shown using its actual brightness level. The brightness of the image in the display can be respectively increased or decreased with the bright and dim settings. The **ShowImageMode** property only affects the appearance of the image in the display. It does not modify the actual greylevel values of the image.

Syntax

control.**ShowImageMode** [= *Value* as hsShowImageMode]

Type

hsShowImageMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsShowImageModeNormal	0	Image is displayed using its normal brightness level.
hsShowImageModeDim	1	Image is displayed with a reduced brightness level.
hsShowImageModeBright	2	Image is displayed with an increased brightness level.

Example

ModelOptimizer.**ShowImageMode**

Methods

Apply

Description

Copies the optimized model to the current models database of the Locator, without closing the Model Optimizer.

Syntax

control.**Apply**

Parameter(s)

None

Return

No return value.

Example

ModelOptimizer.**Apply**

EndDialog

Description

Terminates the Model Optimizer dialog.

Syntax

control.**EndDialog**

Parameter(s)

None

Return

No return value.

Example

ModelOptimizer.**EndDialog**

GetDispatchPointer

Description

Reserved for internal use.

Hide

Description

Hides the Model Optimizer dialog. Unlike **EndDialog**, calling **Hide** hides the dialog but leaves it active without copying the model to the current models database. To terminate the dialog, **EndDialog** must be called.

Syntax

control.**Hide**

Parameter(s)

None

Return

No return value.

Example

`ModelOptimizer.Hide`

Reset

Description

Clears all optimized data recorded so far and resets **InstancesAnalyzed** to zero.

Syntax

control.**Reset**

Parameter(s)

None

Return

No return value.

Example

`Optimizer.Reset`

Show

Description

Shows the Model Optimizer interface in a modeless window. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameter settings can then be changed before making the dialog visible by calling **Show**. To show the Model Optimizer interface in a modal window, use the **ShowModal** method.

Syntax

control.**Show**

Parameter(s)

None

Return

No return value.

Example

```
Dim Optimizer as HSModelOptimizerInterface
' Create invisible Model Optimizer
```

```

Optimizer = Locator.ShowModelOptimizer (False, "Model")
' Adjust parameters
Optimizer.LinkOutlineWithDetail = True
' Show the Model Optimizer in a modeless window
Optimizer.Show

```

ShowModal

Description

Shows the Model Optimizer interface in a modal window. To modify the default settings of the dialog before it is shown to the user, the dialog can be created as invisible. Parameter settings can then be changed before making the dialog visible by calling **ShowModal**. To show the Model Optimizer interface in a modeless window, use the **Show** method.

Syntax

control.**ShowModal**

Parameter(s)

None

Return

No return value.

Example

```

Dim Optimizer as HSModelOptimizerInterface
' Create invisible Model Optimizer
Optimizer = Locator.ShowModelOptimizer (False, "Model")
' Adjust parameters
Optimizer.LinkOutlineWithDetail = True
' Show the Model Optimizer in a modal window
Optimizer.ShowModal

```

Update

Description

Updates the ModelOptimizer user interface dialog and currently recorded ModelOptimizer data.

Syntax

control.**Update**

Parameter(s)

None

Return

No return value.

Example

```

Locator.Execute
Optimizer.Update

```

Events

ApplyClick

Description

Occurs when the Apply button is clicked in the Model Optimizer.

DetailParametersChange

Description

Occurs each time any detail parameters are changed.

DoneClick

Description

Occurs when the Done button is clicked in the Model Optimizer.

OutlineParametersChange

Description

Occurs each time any outline parameters are changed.

ResetClick

Description

Occurs when the Reset button is clicked in the Model Optimizer.

ResultSelectionChange

Description

Occurs when the Result Display selection changes.

OutputTool

Description

The Output Tool can schedule a digital output of a Gocator based on the result of another HexSight tool, at a specific moment in time. The decision to schedule the digital output is usually the result of an Evaluator Tool. Only a True result (or a value different than zero) will schedule the event. To obtain the proper decision, it is possible to use multiple Evaluator Tool for more complex operations (see Evaluator Tool).

Properties

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Output1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Output1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

Handler:

```
MsgBox "Error: " & Output1.ErrorObject.Description
```

InputImage

Description

Name of the image used as reference for the OutputTool. This image provides information for the OutputTool such as the image **TimeStamp** and **EncoderPosition**.

Syntax

control.**InputImage** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim MyOutput As HSOutputTool
MyOutput.InputView = "Acquisition"
MyOutput.InputImage = "Image"
MyOutput.Execute
```

InputToolFunction

Description

Name of the function used to extract a value from another tool.

Syntax

control.**InputToolFunction** [= *Value* As String]

Type

String

Range

Any available property from the tool referenced by the InputToolName property.

Example

```
Output1.InputToolName = "Evaluator1"
Output1.InputToolFunction = "Result"
```

InputToolFunctionIndex

Description

Index used as first parameter for the property referenced by InputToolFunction. This parameter is optional, only used if the InputToolFunction is indexed.

Syntax

control.**InputToolFunctionIndex** [= *Index* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Maximum number of index for InputToolFunction

Example

```
Output1.InputToolName = "HSLocator"
Output1.InputToolFunction = "InstanceMatchQuality"
Output1.InputToolFunctionIndex = 3
```

InputToolName

Description

Name of the tool used to extract a value from a specific function.

Syntax

control.**InputToolName** [= *Value* As String]

Type

String

Range

Any available tool from the Process Manager. The tool should provide at least one property visible to the OutputTool.

Example

```
Output1.InputToolName = "Evaluator1"
Output1.InputToolFunction = "Result"
```

InputView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=*Name* As String]

Type

String

Range

Not applicable.

Example

```
Dim MyOutput AS HSOutputTool
MyOutput.InputView = "Acquisition"
MyOutput.InputImage = "Image"
MyOutput.Execute
```

LastResult

Description

The result of the previous scheduling process. See the Result property for the result of the current execution.

Syntax

control.**LastResult**

Type

hsResult

Range

The valid range for this property is as follows:

Constant	Value	Description
hsResultError	-1	An error occurred preventing the scheduling: event scheduled in the past or bad connection.
hsResultNotProcessed	0	No new incoming data.
hsResultPulsedUnscheduledOnFailFalse	1	The scheduling condition was False for a schedule on Fail, an event was successfully scheduled.
hsResultPulsedUnscheduledOnFailTrue	2	The scheduling condition was True for a schedule on Fail, an event was not scheduled.
hsResultPulsedUnscheduledOnPassFalse	3	The scheduling condition was False for a schedule on Pass, an event was not scheduled.
hsResultPulsedUnscheduledOnPassTrue	4	The scheduling condition was True for a schedule on Pass, an event was successfully scheduled.
hsResultPulsedScheduledTimeOnFailFalse	5	The scheduling condition was False for a schedule on Fail, an event was successfully scheduled based on Time.
hsResultPulsedScheduledTimeOnFailTrue	6	The scheduling condition was True for a schedule on Fail, an event was not scheduled.
hsResultPulsedScheduledTimeOnPassFalse	7	The scheduling condition was False for a schedule on Pass, an event was not scheduled.
hsResultPulsedScheduledTimeOnPassTrue	8	The scheduling condition was True for a schedule on Pass, an event was successfully scheduled based on Time.
hsResultPulsedScheduledEncoderOnFailFalse	9	The scheduling condition was False for a schedule on Fail, an event was successfully scheduled based on Encoder.

Constant	Value	Description
hsResultPulsedScheduledEncoderOnFailTrue	10	The scheduling condition was True for a schedule on Fail, an event was not scheduled.
hsResultPulsedScheduledEncoderOnPassFalse	11	The scheduling condition was False for a schedule on Pass, an event was not scheduled.
hsResultPulsedScheduledEncoderOnPassTrue	12	The scheduling condition was True for a schedule on Pass, an event was successfully scheduled based on Encoder.
hsResultContinuousOnFailFalse	13	The scheduling condition was False for a schedule on Fail, output was set to high value.
hsResultContinuousOnFailTrue	14	The scheduling condition was True for a schedule on Fail, output was set to low value.
hsResultContinuousOnPassFalse	15	The scheduling condition was False for a schedule on Pass, output was set to low value.
hsResultContinuousOnPassTrue	16	The scheduling condition was True for a schedule on Pass, output was set to high value.
hsResultErrorSchedulingOffset	128	Offset for error results
hsResultErrorPulsedUnscheduledOnFailFalse	129	The scheduling condition was False for a schedule on Fail, but scheduling failed.
hsResultErrorPulsedUnscheduledOnPassTrue	132	The scheduling condition was True for a schedule on Pass, but scheduling failed.
hsResultErrorPulsedScheduledTimeOnFailFalse	133	The scheduling condition was False for a schedule on Fail, but scheduling failed.
hsResultErrorPulsedScheduledTimeOnPassTrue	136	The scheduling condition was True for a schedule on Pass, but scheduling failed.
hsResultErrorPulsedScheduledEncoderOnFailFalse	137	The scheduling condition was False for a schedule on Fail, but scheduling failed.
hsResultErrorPulsedScheduledEncoderOnPassTrue	140	The scheduling condition was True for a schedule on Pass, but scheduling failed.
hsResultErrorContinuousOnFailFalse	141	The scheduling condition was False for a schedule on Fail, but output set to high has failed.
hsResultErrorContinuousOnFailTrue	142	The scheduling condition was True for a schedule on Fail, but output set to low has failed.
hsResultErrorContinuousOnPassFalse	143	The scheduling condition was False for a schedule on Pass, but output set to low has failed.
hsResultErrorContinuousOnPassTrue	144	The scheduling condition was True for a schedule on Pass, but output set to high has failed.

Example

```
Dim Result As hsResult
Result = Output1.LastResult
```


LastScheduledEncoder

Description

The encoder position in ticks at which the scheduling occurred, if the Gocator was configured for encoder input, in the previous execution. Read only.

Syntax

control.**LastScheduledEncoder**

Type

Long

Range

Not applicable.

Example

```
Dim ScheduledEncoder As Long
ScheduledEncoder = Output1.LastScheduledEncoder
```

LastScheduledTime

Description

The time in microseconds at which the scheduling occurred, if the Gocator was configured for time input, in the previous execution. Read only.

Syntax

control.**LastScheduledTime**

Type

ULong

Range

Not applicable.

Example

```
Dim ScheduledTime As ULong
ScheduledTime = Output1.LastScheduledTime
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= *Value* As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcessName As String
ProcessName = Output1.Name
Output2.Name = "OutputPort1"
```

Offset

Description

The offset in ticks or delay in microseconds, from the InputImage stamps, to apply to the scheduling. The units are deduced from the current Gocator settings.

Syntax

control.**Offset** [= *Value* As Long]

Type

Long

Range

Not applicable.

Example

Output1.**Offset** = 1000000

OutputAssertOn

Description

The schedule decision can be based either on a True or a False from the evaluated result. If OutputAssertOn is set to True, the output is scheduled on a True result, if set to False, it is scheduled on a False result.

Syntax

control.**OutputAssertOn** [= *Value* As Boolean]

Type

Boolean

Range

True or False.

Example

Output1.**OutputAssertOn** = True

OutputIndex

Description

The digital output of the Gocator to schedule.

Syntax

control.**OutputIndex** [= *Value* As Integer]

Type

Integer

Range

Greater than or equal to 0.

Example

```
Dim OutputIndex As Integer
OutputIndex = Output1.OutputIndex
```

OutputIP

Description

The IP address of the Gocator to schedule.

Syntax

control.**OutputIP** [= *Value* As String]

Type

String

Range

Dot formatted IP address.

ExampleOutput1.**OutputIP** = "192.168.1.10"

OutputUnits

Description

Current Gocator schedule units. Read only.

Syntax*control*.**OutputUnits****Type**

hsUnits

Range

The valid range for this property is as follows:

Constant	Value	Description
hsUnitsUnknown	0	Unknown units
hsUnitsTicks	1	Encoder ticks
hsUnitsMicroSeconds	2	Micro seconds

Example

Dim Unit As hsUnits

Unit = Output1.**OutputUnits**

ProcessManager

Description

Reserved for internal use.

Result

Description

The result of the scheduling process.

Syntax*control*.**Result****Type**

hsResult

Range

The valid range for this property is as follows:

Constant	Value	Description
hsResultError	-1	An error occurred preventing the scheduling: event scheduled in the past or bad connection.
hsResultNotProcessed	0	No new incoming data.

Constant	Value	Description
hsResultPulsedUnscheduledOnFailFalse	1	The scheduling condition was False for a schedule on Fail, an event was successfully scheduled.
hsResultPulsedUnscheduledOnFailTrue	2	The scheduling condition was True for a schedule on Fail, an event was not scheduled.
hsResultPulsedUnscheduledOnPassFalse	3	The scheduling condition was False for a schedule on Pass, an event was not scheduled.
hsResultPulsedUnscheduledOnPassTrue	4	The scheduling condition was True for a schedule on Pass, an event was successfully scheduled.
hsResultPulsedScheduledTimeOnFailFalse	5	The scheduling condition was False for a schedule on Fail, an event was successfully scheduled based on Time.
hsResultPulsedScheduledTimeOnFailTrue	6	The scheduling condition was True for a schedule on Fail, an event was not scheduled.
hsResultPulsedScheduledTimeOnPassFalse	7	The scheduling condition was False for a schedule on Pass, an event was not scheduled.
hsResultPulsedScheduledTimeOnPassTrue	8	The scheduling condition was True for a schedule on Pass, an event was successfully scheduled based on Time.
hsResultPulsedScheduledEncoderOnFailFalse	9	The scheduling condition was False for a schedule on Fail, an event was successfully scheduled based on Encoder.
hsResultPulsedScheduledEncoderOnFailTrue	10	The scheduling condition was True for a schedule on Fail, an event was not scheduled.
hsResultPulsedScheduledEncoderOnPassFalse	11	The scheduling condition was False for a schedule on Pass, an event was not scheduled.
hsResultPulsedScheduledEncoderOnPassTrue	12	The scheduling condition was True for a schedule on Pass, an event was successfully scheduled based on Encoder.
hsResultContinuousOnFailFalse	13	The scheduling condition was False for a schedule on Fail, output was set to high value.
hsResultContinuousOnFailTrue	14	The scheduling condition was True for a schedule on Fail, output was set to low value.
hsResultContinuousOnPassFalse	15	The scheduling condition was False for a schedule on Pass, output was set to low value.
hsResultContinuousOnPassTrue	16	The scheduling condition was True for a schedule on Pass, output was set to high value.
hsResultErrorSchedulingOffset	128	Offset for error results
hsResultErrorPulsedUnscheduledOnFailFalse	129	The scheduling condition was False for a schedule on Fail, but scheduling failed.
hsResultErrorPulsedUnscheduledOnPassTrue	132	The scheduling condition was True for a schedule on Pass, but scheduling failed.
hsResultErrorPulsedScheduledTimeOnFailFalse	133	The scheduling condition was False for a schedule on Fail, but scheduling failed.

Constant	Value	Description
hsResultErrorPulsedScheduledTimeOnPassTrue	136	The scheduling condition was True for a schedule on Pass, but scheduling failed.
hsResultErrorPulsedScheduledEncoderOnFailFalse	137	The scheduling condition was False for a schedule on Fail, but scheduling failed.
hsResultErrorPulsedScheduledEncoderOnPassTrue	140	The scheduling condition was True for a schedule on Pass, but scheduling failed.
hsResultErrorContinuousOnFailFalse	141	The scheduling condition was False for a schedule on Fail, but output set to high has failed.
hsResultErrorContinuousOnFailTrue	142	The scheduling condition was True for a schedule on Fail, but output set to low has failed.
hsResultErrorContinuousOnPassFalse	143	The scheduling condition was False for a schedule on Pass, but output set to low has failed.
hsResultErrorContinuousOnPassTrue	144	The scheduling condition was True for a schedule on Pass, but output set to high has failed.

Example

```
Dim Result As hsResult
Result = Output1.Result
```

ScheduledEncoder

Description

The encoder position in ticks at which the scheduling occurred, if the Gocator was configured for encoder input. Read only.

Syntax

control.**ScheduledEncoder**

Type

Long

Range

Not applicable.

Example

```
Dim ScheduledEncoder As Long
ScheduledEncoder = Output1.ScheduledEncoder
```

ScheduledTime

Description

The time in microseconds at which the scheduling occurred, if the Gocator was configured for time input. Read only.

Syntax

control.**ScheduledTime**

Type

ULong

Range

Not applicable.

Example

```
Dim ScheduledTime As ULong
ScheduledTime = Output1.ScheduledTime
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.**TypeName**

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSOutputTool".

Example

```
Dim ControlType As String
ControlType = Output1.TypeName
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Output1.Version
```

Methods

AboutBox

Description

Displays the tool **About** box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Return

No return value.

Example

Output1.AboutBox

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the OutputTool to schedule an event at the specified offset.

Syntax*control*.Execute()**Parameter(s)**

None

Return

Single: Processing time in milliseconds.

Example

```
Dim Output1 AS HSOutputTool
Output1.InputView = "Acquisition"
Output1.InputImage = "Image"
Output1.Execute
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the tool properties interface.

Syntax*control*.HideProperties()**Parameter(s)**

None

Return

Boolean

Range

Value	Description
True	The interface was displayed when the method was called.
False	The interface was not displayed.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Output1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.ShowProperties(Modal As Boolean, [Variant Option]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the table below.

<i>Option</i> parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button

Option parameter: bitfield elements		
Element	Value	Description
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool process is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool process and the subsequent tools in the ProcessManager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = Output1.ShowProperties _
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Pattern Locator

Description
The Pattern Locator finds and locates instances of a pattern occurring within an Image.

Properties

AutoCoarsenessSelectionEnabled

Description
When **AutoCoarsenessSelectionEnabled** is True, the values of **SearchCoarseness** and **PositioningCoarseness** are automatically determined by the Pattern Locator process when the pattern is learned.

Syntax
`control.AutoCoarsenessSelectionEnabled [=Value As Boolean]`

Type
Boolean

Range

Value	Description
True	Automatic subsampling is enabled.
False	Automatic subsampling is disabled.

Example
`Dim Result As Boolean`
`Result = PatternLocator1.AutoCoarsenessSelectionEnabled`
`PatternLocator2.AutoCoarsenessSelectionEnabled = True`

BilinearInterpolationEnabled

Description
Specifies if bilinear interpolation is used to sample the pattern and search Images.

Syntax
`control.BilinearInterpolationEnabled [=Value C]`

Type
Boolean

Range
The valid range for this property is as follows:

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

```
Dim Result As Boolean
Result = PatternLocator1.BilinearInterpolationEnabled
PatternLocator2.BilinearInterpolationEnabled = True
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [=Value As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is Model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = PatternLocator1.CalibratedUnitsEnabled
PatternLocator2.CalibratedUnitsEnabled = False
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [=Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.

Constant	Value	Description
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input Image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```
Dim Result As hsCoordinateSystem
Result = PatternLocator1.CoordinateSystem
PatternLocator1.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = PatternLocator1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax*control*.**ElapsedTime****Type**

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent PatternLocator1.ElapsedTime
```

ErrorObject

DescriptionReturns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.**Syntax***control*.**ErrorObject****Type**

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

InputImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax*control*.**InputImage** [=Name As String]**Type**

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = PatternLocator1.InputImage
PatternLocator2.InputImage = "image1"
```

InputGreyScaleImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

control.**InputImage** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = PatternLocator1.InputImage
PatternLocator2.InputImage = "image1"
```

InputView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = PatternLocator1.InputView
PatternLocator2.InputView = "view0"
```

InputGreyScaleView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = PatternLocator1.InputView
PatternLocator2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the **Locator** tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = PatternLocator1.InputInstanceScene
PatternLocator2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.**InputInstanceView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = PatternLocator1.InputInstanceView
PatternLocator2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.**InputModelScene** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
```



```
ModelSceneName = PatternLocator1.InputModelScene
PatternLocator2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View that provides the Model Scene. See **InputModelScene** for more details.

Syntax

```
control.InputModelView [=Name As String]
```

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = PatternLocator1.InputModelView
PatternLocator2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

Specifies if the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property.

Syntax

```
control.InstanceSelectionEnabled [=Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Remark(s)

This property is only used if the **ModelTransformEnabled** is set to True.

Example

```
Dim Enabled As Boolean
Enabled = PatternLocator1.InstanceSelectionEnabled
PatternLocator2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **InstanceSelectionEnabled** property is set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.**InstanceSelectionIndex** [=Value As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
```

```
Ndx = PatterLocator1.InstanceSelectionIndex
```

```
PatterLocator2.InstanceSelectionIndex = 0
```

MatchCount

Description

Number of pattern matches found. Read only.

Syntax

control.**MatchCount**

Type

Integer

Range

Minimum: 0

Maximum: unlimited

Example

```
Dim NMatch As Integer
```

```
NMatch = PatternLocator1.MatchCount
```

MatchPositionX

Description

X coordinate of a matched pattern in the currently selected coordinate system. Read only.

Syntax

control.**MatchPositionX**(Index As Integer)

Type

Single

Range

Boundaries of the input image.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the match of interest.	Integer	Minimum: 0 Maximum: MatchCount - 1

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystem Image**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim XPos As Single
XPos = PatternLocator1.MatchPositionX(0)
```

MatchPositionY

Description

Y coordinate of a matched pattern in the currently selected coordinate system. Read only.

Syntax

control.MatchPositionY(Index As Integer)

Type

Single

Range

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the match of interest.	Integer	Minimum: 0 Maximum: MatchCount - 1

Example

```
Dim YPos As Single
YPos = PatternLocator1.MatchPositionY(0)
```

MatchRotation

Description

Rotation of the selected matched pattern in the currently selected coordinate system. Read only.

Syntax

control.MatchRotation(Index As Integer)

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	MatchRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the match of interest.	Integer	Minimum: 0 Maximum: MatchCount - 1

Remark(s)

The rotation is defined as the angle between the x axis of the active coordinate system, specified by the **CoordinateSystem** property, and the selected matched pattern, expressed using the units specified in **UnitsOrientation**.

Example

```
Dim Rotation As Single
Rotation = PatternLocator1.MatchRotation(0)
```

MatchStrength

Description

Strength from the match matrix for the selected matched pattern. Read only.

Syntax

```
control.MatchStrength(Index As Integer)
```

Type

Single

Range

Minimum: **MatchThreshold**
Maximum: 1.0

Parameter(s)

Name	Description	Type	Range
<i>Index</i>	Index of the match of interest.	Integer	Minimum: 0 Maximum: MatchCount - 1

Example

```
Dim Strength As Single
Strength = PatternLocator1.MatchStrength(0)
```

MatchThreshold

Description

Minimum match strength required by the tool to accept a match when searching the match matrix.

Syntax

```
control.MatchThreshold [=Value As Single]
```

Type

Single

Range

Minimum: 0.0 (weak match)
Maximum: 1.0 (strong match)

Example

```
Dim Threshold As Single
Threshold = PatternLocator1.MatchThreshold
PatternLocator2.MatchThreshold = 0.78
```

MaximumInstanceCount

Description

Maximum number of pattern instances to find. All pattern instances respecting the constraints are output, up to a maximum of **MaximumInstanceCount**. They are sorted in descending order of quality.

Syntax

control.**MaximumInstanceCount** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1

Remark(s)

This property is applicable only if the **MaximumInstanceCountEnabled** property is set to True.

Example

```
Dim MaxPatterns As Integer
MaxPatterns = PatternLocator1.MaximumInstanceCount
PatternLocator2.MaximumInstanceCount = 10
```

MaximumInstanceCountEnabled

Description

When **MaximumInstanceCountEnabled** is True, the **MaximumInstanceCount** property is applied.

Syntax

control.**MaximumInstanceCountEnabled** [= *Value* As Boolean]

Type

Boolean

Range

True, False.

Example

```
Dim MaxInstanceCountEnabled As Boolean
MaxInstanceCountEnabled = _
    PatternLocator1.MaximumInstanceCountEnabled
PatternLocator2.MaximumInstanceCountEnabled = False
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a Model Image and a transform is applied according to the object to be inspected in the input image. This

transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

`control.ModelTransformEnabled [=Value As Boolean]`

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = PatternLocator1.ModelTransformEnabled
PatternLocator2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

`control.Name [=Value As String]`

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = PatternLocator1.Name
PatternLocator2.Name = "PatLocA"
```

OverrideModelImageEnabled

Description

For a model-based tool only (**ModelTransformEnabled**), setting **OverrideModelImageEnabled** to True makes it possible to define a pattern from image other than the model image. For a non-model-based tool this property is invalid.

Syntax

`control.OverrideModelImageEnabled [=Value As Boolean]`

Type

Boolean

Range

Value	Description
True	Allows the pattern to be defined from an image other than the model image.

Value	Description
False	Pattern can only be defined in the model image.

Example

```
Dim Result As Boolean
Result = PatternLocator1.OverrideModelImageEnabled
PatternLocator2.OverrideModelImageEnabled = True
```

PatternDatabaseFileName

Description

Name of the database file where the pattern image is saved. Valid if the tool is non-model-based or if the tool is model-based and **OverrideModelImageEnabled** is True.

Syntax

control.**PatternDatabaseFileName**

Type

String

Range

Not applicable.

Example

```
Dim FileName As String
FileName = PatternLocator1.PatternDatabaseFileName
```

PatternHeight

Description

Height of the pattern rectangle. Can be defined in either the pattern image or the model image, depending of the state of **ModelTransformEnabled** and the **OverrideModelImageEnabled**.

Syntax

control.**PatternHeight** [=Value As Single]

Type

Single

Range

Minimum: Greater than 0.0

Maximum: boundaries of the image

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
Height = PatternLocator1.PatternHeight
PatternLocator2.PatternHeight = 23.4
```

PatternPositionX

Description

X coordinate of the center of the pattern rectangle. Can be defined in either the pattern image or the model image, depending of the state of **ModelTransformEnabled** and **OverrideModelImageEnabled**.

Syntax

control.**PatternPositionX** [=Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim X_Pos As Single
X_Pos = PatternLocator1.PatternPositionX
PatternLocator2.ToolPositionX = 23.2
```

PatternPositionY

Description

Y coordinate of the center of the pattern rectangle. Can be defined in either the pattern image or the model image, depending of the state of **ModelTransformEnabled** and **OverrideModelImageEnabled**.

Syntax

control.**PatternPositionY** [=Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the model)

Example

```
Dim Y_Pos As Single
Y_Pos = PatternLocator1.PatternPositionY
PatternLocator2.ToolPositionY = 23.2
```

PatternRotation

Description

Rotation of the pattern rectangle. Can be defined in either the pattern image or the model image, depending of the state of **ModelTransformEnabled** and **OverrideModelImageEnabled**.

Syntax

control.**ToolRotation** [=Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = PatternLocator1.PatternRotation
PatternLocator2.PatternRotation = -10.0
```

PatternWidth

Description

Width of the pattern rectangle. Can be defined in either the pattern image or the model image, depending of the state of **ModelTransformEnabled** and **OverrideModelImageEnabled**.

Syntax

control.**PatternWidth** [=Value As Single]

Type

Single

Range

Minimum: Greater than 0.0

Maximum: boundaries of the image

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

Dim Width As Single

Width =PatternLocator1.**PatternWidth**

PatternLocator2.**PatternWidth** = 23.4

PositioningCoarseness

Description

Subsampling level at which the hypothesis refinement ends. High values provide coarser positioning and lower execution time than higher values. Read only if **AutoCoarsenessSelectionEnabled** is True.

Syntax

control.**PositioningCoarseness**[= Value As Integer]

Type

Short

Range

[1,2,4]

Remarks

Can only be less than or equal to the **SearchCoarseness** value.

Example

Dim PositionLev As Single

PositionLev = PatternLocator1.**PositioningCoarseness**

PatternLocator2.**PositioningCoarseness** = 2

SamplingStep

Description

Step used in the tool's last execution to sample the pattern and search rectangles from the pattern Image and the input images. When the tool is model-based this step is expressed in calibrated units, otherwise it is expressed either in pixels or calibrated units as defined by the **CalibratedUnitsEnabled** property. All pixels in the sampled rectangle are square and of the

same size. The sampling step represents the height and the width of a sampled pixel. Read only.

Syntax

control.**SamplingStep**

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
```

```
Step = PatternLocator1.SamplingStep
```

SamplingStepCustom

Description

Sampling step factor defined by the user to override the default sampling step that can be computed by the tool. This value is used only if the **SamplingStepCustomEnabled** property is set to True.

Syntax

control.**SamplingStepCustom** [=Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
```

```
Step = PatternLocator1.SamplingStepCustom
```

```
PatternLocator2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

Value that specifies if the default sampling step is overridden by the **SamplingStepCustom** property.

Syntax

control.**SamplingStepCustomEnabled** [=Value As Boolean]

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom .

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Enabled As Boolean
Enabled = PatternLocator1.SamplingStepCustomEnabled
PatternLocator2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Default sampling step for the current input and pattern Images. Read only.

Syntax

control.**SamplingStepDefault**

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = PatternLocator.SamplingStepDefault
```

SearchCoarseness

Description

Subsampling level used to find pattern match hypotheses. High values provide a coarser search and lower execution time than lower values. Read only if

AutoCoarsenessSelectionEnabled is True.

Syntax

control.**SearchCoarseness** [= Value As Integer]

Type

short

Range

[1,2,4,8,16,32]

Remarks

Can only be higher than or equal to the **PositioningCoarseness** value.

Example

```
Dim SearchLev As Single
SearchLev = PatternLocator1.SearchCoarseness
PatternLocator2.SearchCoarseness = 8
```

ToolHeight

Description

Height of the tool rectangle. Can be defined in either the input image or the model image, depending of the state of **ModelTransformEnabled**.

Syntax

control.**ToolHeight** [=Value As Single]

Type

Single

Range

Minimum: greater than 0.0

Maximum: boundaries of the image

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Height As Single
```

```
Height = PatternLocator1.ToolHeight
```

```
PatternLocator2.ToolHeight = 23.4
```

ToolPositionX

Description

X coordinate of the center of the tool rectangle.

Syntax

control.**ToolPositionX** [=Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim X_Pos As Single
```

```
X_Pos = PatternLocator1.ToolPositionX
```

```
PatternLocator2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the position of the tool rectangle.

Syntax

control.**ToolPositionY** [=Value As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = PatternLocator1.ToolPositionY
PatternLocator2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool rectangle.

Syntax

control.**ToolRotation** [=Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = PatternLocator1.ToolRotation
PatternLocator2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool rectangle. Can be defined in either the input image or the model image, depending of the state of **ModelTransformEnabled**.

Syntax

control.**ToolWidth** [=Value As Single]

Type

Single

Range

Minimum: greater than 0.0
Maximum: boundaries of the image

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = PatternLocator1.ToolWidth
PatternLocator2.ToolWidth = 23.4
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [=Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
```

```
Angle = PatternLocator1.TransformRotation
```

```
PatternLocator2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

```
control.TransformScale [=Value As Single]
```

Type

Single

Range

Minimum: greater than zero.

Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
```

```
Scale = PatternLocator1.TransformScale
```

```
PatternLocator2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

```
control.TransformTranslationX [=Value As Single]
```


Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = PatternLocator1.TransformTranslationX
PatternLocator2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [=Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = PatternLocator1.TransformTranslationY
PatternLocator2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSPatternLocator".

Example

```
Dim ControlType As String
ControlType = PatternLocator1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = PatternLocator1.UnitsLength
PatternLocator2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = PatternLocator1.UnitsOrientation
PatternLocator2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = PatternLocator1.Version
```

Methods

AboutBox

Description

Displays the Pattern Locator tool About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

```
PatternLocator1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

CreatePatternFromInput

Description

Creates a pattern image from the input image. The pattern used for the search is a selection of that pattern image.

Syntax

control.**CreatePatternFromInput()**

Parameter(s)

None

Return

None

Example

PatternLocator1.CreatePatternFromInput

Execute

Description

Executes the Pattern Locator process.

Syntax

control.**Execute()**

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Range

Minimum: 0.0

Maximum: unlimited

Example

PatternLocator1.**Execute**

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the tool properties interface.

Syntax

control.**HideProperties()**

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = PatternLocator1.HideProperties
```

ImportPatternFromDatabase

Description

Creates a pattern image from an image imported from a database file. This method requires manual interaction of the user to select both the database and the image from the database. The pattern used for the search is a selection defined on this pattern image.

Syntax

```
control.ImportPatternFromDatabase()
```

Parameter(s)

None

Return

None

Example

```
PatternLocator1.ImportPatternFromDatabase
```

LoadConfiguration

Description

Reserved for internal use.

LoadPatternDatabase

Description

Loads a pattern database.

Syntax

```
control.LoadPatternDatabase([FileName As Variant])
```

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of the file to load. Optional.	Variant	Must be a string.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If no *FileName* parameter is provided, a file open dialog appears.

Example

```
Dim Loaded As Boolean
Loaded = PatternLocator1.LoadPatternDatabase
```

`PatternLocator1.LoadPatternDatabase ".\Pattern\Logo"`

NewPatternDatabase

Description

Creates a new empty pattern database and clears the current pattern database.

Syntax

`control.NewPatternDatabase`

Parameter(s)

None.

Return

No return value.

Example

`PatternLocator1.NewPatternDatabase`

SaveConfiguration

Description

Reserved for internal use.

SavePatternDatabase

Description

Saves a pattern database file.

Syntax

`control.SavePatternDatabase([FileName As Variant]) As Boolean`

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of the file to save. Optional.	Variant	Must be a string.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Remark(s)

If no *FileName* parameter is provided, a file save dialog appears.

Example

```
Dim Succeeded As Boolean
Succeeded = _
    PatternLocator1.SavePatternDatabase("..\\Patterns.hdb")
Succeeded = PatternLocator2.SavePatternDataBase
```

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool process is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool process and the subsequent tools in the ProcessManager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Dim Success As Boolean
Success = PatternLocator1.ShowProperties _
(True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

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.

Properties

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

control.**CalibratedUnitsEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is Model- based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = PointFinder1.CalibratedUnitsEnabled
PointFinder2.CalibratedUnitsEnabled = False
```

Connectivity

Description

Defines a minimum number of connected edges required to generate a point hypothesis from a specific found edge that satisfies search constraints.

Syntax

control.**Connectivity** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1
Maximum: 20

Example

```
Dim Connectivity As Integer
```

```

PointFinder1.ConnectivityEnabled = True
Connectivity = PointFinder1.Connectivity

```

ConnectivityEnabled

Description

When **ConnectivityEnabled** is set to True, the tool uses the value of the **Connectivity** property to generate a point hypothesis.

Syntax

```
control.ConnectivityEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Connectivity is enabled.
False	Connectivity is disabled.

Example

```

Dim Enabled As Boolean
Enabled = PointFinder1.ConnectivityEnabled
PointFinder2.ConnectivityEnabled = False

```

ContrastThreshold

Description

Defines the minimum contrast needed for an edge to be detected in the input image and used for point computation. This threshold is expressed in terms of a step in greylevel values. The property can only be set when **ContrastThresholdMode** is set to **hsContrastThresholdFixedValue**. Read only otherwise.

Syntax

```
control.ContrastThreshold [= Value As Integer]
```

Type

Integer

Range

Minimum: 1
Maximum: 255

Remark(s)

By default, the tool selects a **ContrastThresholdMode** based on image content to provide flexibility to variations in image lighting conditions and contrast. Adaptive threshold modes are generally recommended. A fixed value Contrast Threshold should only be used when adaptive values do not provide satisfactory results.

Example

```

Dim CurrentThreshold As Integer
CurrentThreshold = LineFinder1.ContrastThreshold
PointFinder1.ContrastThresholdMode = _
    hsContrastThresholdModeFixedValue

```

```
PointFINDER2.ContrastThreshold = 10
```

ContrastThresholdMode

Description

Selects the method used to compute the threshold used for detecting edges in the input image.

Syntax

```
control.ContrastThresholdMode [= Value As hsContrastThresholdMode]
```

Type

hsContrastThresholdMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsContrastThresholdAdaptiveLowSensitivity	0	Uses a low sensitivity adaptive threshold for detecting edges. Adaptive Low Sensitivity reduces the amount of noisy edges but may also cause significant edges to be undetected.
hsContrastThresholdAdaptiveNormalSensitivity	1	Uses a normal sensitivity adaptive threshold for detecting edges.
hsContrastThresholdAdaptiveHighSensitivity	2	Uses a high sensitivity adaptive threshold for detecting edges. Adaptive High Sensitivity can help detect weak-contrast edges but also increases the amount of noisy edges.
hsContrastThresholdFixedValue	3	Uses a fixed value threshold for detecting edges.

Example

```
Dim CurrentMode As hsContrastThresholdMode
CurrentMode = PointFINDER1.ContrastThresholdMode
PointFINDER1.ContrastThresholdMode = _
    hsContrastThresholdModeFixedValue
PointFINDER2.ContrastThresholdMode = 10
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

```
control.CoordinateSystem [= Value As hsCoordinateSystem]
```

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input Image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to the Tool coordinate system.

Example

```

Dim CurrentSystem As hsCoordinateSystem
' Get the current coordinate system
CurrentSystem = PointFinder1.CoordinateSystem
' Express the results in the Object coordinate system
PointFinder1.CoordinateSystem = hsCoordinateSystemObject

```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = PointFinder1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: Unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = PointFinder1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & PointFinder.ErrorObject.Description
```

InputImage

InputGreyScaleImage

Description

Name of the image within which the operation of the tool is carried out.

Syntax

control.**InputImage** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = PointFinder1.InputImage
PointFinder2.InputImage = "image1"
```

InputView

InputGreyScaleView

Description

Name of the View that provides the image. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = PointFinder1.InputView
PointFinder2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Scene — created by the **Locator** tool — that is used to store instances of located objects along with their transform. This Scene is only used in the model-based mode (**ModelTransformEnabled** property set to True) when the **InstanceSelectionEnabled** property is set to True. Along with the **InstanceSelectionIndex** property, the Instance Scene is used to automatically apply a transform to the tool to inspect a selected object instance.

Syntax

control.**InputInstanceScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = PointFinder1.InputInstanceScene
PointFinder2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View that provides the Instance Scene. See **InputInstanceScene** for more details.

Syntax

control.InputInstanceView [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = PointFinder1.InputInstanceView
PointFinder2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Model Scene. The tool retrieves the Object coordinate system information from the Model Scene.

Syntax

control.InputModelScene [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = PointFinder1.InputModelScene
PointFinder2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View that provides the Model Scene. See **InputModelScene** for more details.

Syntax

control.InputModelView [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = PointFinder1.InputModelView
PointFinder2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

When True, specifies that the tool transform parameters are automatically applied from an object instance appearing in the Instance Scene set by the **InputInstanceScene** property. The instance is selected using the **InstanceSelectionIndex** property. When False, the tool transform must be specified manually using the **TransformRotation**, **TransformScale**, **TransformTranslationX** and **TransformTranslationY** properties.

Syntax

control.**InstanceSelectionEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = PointFinder1.InstanceSelectionEnabled
PointFinder2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the selected instance from which the tool transform parameters are applied when the **ModelTransformEnabled** and **InstanceSelectionEnabled** property are set to True. The Instance Scene is set using the **InputInstanceScene** property.

Syntax

control.**InstanceSelectionIndex** [= *Value* As Integer]

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator, minus 1

Example

```
Dim Ndx As Integer
Ndx = PointFinder1.InstanceSelectionIndex
PointFinder2.InstanceSelectionIndex = 0
```

InterpolatePositionMode

Description

Sets the mode used by the tool to compute a point hypothesis.

Syntax

control.**InterpolatePositionMode** [= *Value* As hsInterpolatePositionMode]

Type

hsInterpolatePositionMode

Range

The valid range for this property is as follows:

Constant	Value	Description
hsInterpolateCorner	0	The tool will compute a hypothesis that fits a corner point to interpolated lines from connected edges.
hsInterpolateIntersection	1	The tool will compute a hypothesis that is an intersection between the search axis and connected edges of an interpolated line.

Example

```
Dim CurrentMode As hsInterpolatePositionMode
CurrentMode = LineFinder2.hsInterpolatePositionMode
PointFinder2.InterpolatePositionMode = _
    hsInterpolateCorner
LineFinder2.hsInterpolatePositionMode = 10
```

InterpolatePositionModeEnabled

Description

When **InterpolatePositionModeEnabled** is set to True, the tool uses the value set by the **InterpolatePositionMode** property to compute a point hypothesis. Otherwise, point hypothesis coordinates are taken directly from a specific found edge that satisfies search constraints.

Syntax

```
control.InterpolatePositionModeEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Interpolate position is enabled.
False	Interpolate position is disabled.

Example

```
Dim Enabled As Boolean
Result = PointFinder.InterpolatePositionModeEnabled
PointFinder2.InterpolatePositionModeEnabled = False
```

MessageCount

Description

The number of messages issued during the last execution. The identification number and description of each message is retrieved using respectively the **MessageNumber** and **MessageDescription** properties. Read only.

Syntax*control*.**MessageCount****Type**

Integer

Range

Minimum: 0

Maximum: Unlimited

Example

```

Dim m As Integer
For m = 0 To PointFinder1.MessageCount - 1
    Description.Text = PointFinder1.MessageDescription(m)
    Number.Text = Str(PointFinder1.MessageNumber(m))
Next m

```

MessageDescription

Description

The description of a message issued during the last execution. The identification number of the message corresponding to this description is retrieved using the **MessageNumber** property.

MessageCount returns the total number of messages issued. Read only.

Syntax*control*.**MessageDescription**(*Index* As Integer)**Type**

String

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```

Dim m As Integer
For m = 0 To PointFinder1.MessageCount - 1
    Description.Text = PointFinder1.MessageDescription(m)
    Number.Text = Str(PointFinder1.MessageNumber(m))
Next m

```

MessageNumber

Description

The identification number of a message issued during the last execution. The description of the message corresponding to this identification number is retrieved using the

MessageDescription property. **MessageCount** returns the total number of messages issued. Read only.

Syntax

control.**MessageNumber**(*Index* As Integer)

Type

Integer

Range

Not applicable.

Parameter(s)

Name	Description	Type	Range
Index	Index of the message.	Integer	Minimum: 0 Maximum: MessageCount - 1

Example

```
Dim m As Integer
For m = 0 To PointFinder1.MessageCount - 1
    Description.Text = PointFinder1.MessageDescription(m)
    Number.Text = Str(PointFinder1.MessageNumber(m))
Next m
```

ModelTransformEnabled

Description

Specifies if the tool is model-based. This means that the tool is defined on a Model Image and a transform is applied according to the object to be inspected in the input image. This transform can be supplied manually using **TransformScale**, **TransformRotation**, **TransformTranslationX** and **TransformTranslationY** or automatically using an Object instance set by the **InputInstanceScene** and **InstanceSelectionIndex** properties.

Syntax

control.**ModelTransformEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = PointFinder1.ModelTransformEnabled
PointFinder2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.Name [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = PointFinder1.Name
PointFinder2.Name = "PointFindA"
```

OutputEntity

Description

Name of the entity created by the tool. This entity is optionally output according to the value of the **OutputEntityEnabled** property.

Syntax

control.OutputEntity [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim Point As String
Point = PointFinder1.OutputEntity
PointFinder1.OutputEntity = "Point0"
```

OutputEntityEnabled

Description

Specifies if a found entity will be output to the runtime database.

Syntax

control.OutputEntityEnabled [= Value As Boolean]

Type

Boolean

Range

Value	Description
False	The entity will not be output.
True	The entity will be output.

Example

```
Dim Enabled As Boolean
Enabled = PointFinder1.OutputEntityEnabled
PointFinder1.OutputEntityEnabled = False
```

OutputEntityView

Description

Name of the View to which the Output Entity is output. See **OutputEntity** for more details.

Syntax

control.**OutputEntityView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = PointFinder1.OutputEntityView
PointFinder2.OutputEntityView = "PointView0"
```

PointAverageContrast

Description

Average contrast between light and dark pixels on either side of the found point, expressed in greylevel values. Read only.

Syntax

control.**PointAverageContrast**

Type

Integer

Range

Minimum: greater than 0.
Maximum: 255

Example

```
Dim AverageContrast As Integer
PointFinder1.Execute
AverageContrast = PointFinder1.PointAverageContrast
```

PointFound

Description

When True, indicates that a Point was found. Read only.

Syntax

control.**PointFound**

Type

Boolean

Range

Not applicable.

Example

```
Dim PointFound as Boolean
PointFound = PointFinder1.PointFound
```

PointPositionX

Description

X coordinate of the found Point, expressed in the currently selected coordinate system. Read only.

Syntax

control.**PointPositionX**

Type

Single

Range

Boundaries of the input image.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim PointX as Single
PointX = PointFinder1.PointPositionX
```

PointPositionY

Description

Y coordinate of the found point, expressed in the currently selected coordinate system. Read only.

Syntax

control.**PointCenterPointPositionY**

Type

Single

Range

Boundaries of the input image.

Remark(s)

In addition to **CoordinateSystem**, this property expression is dependent on both **CalibratedUnitsEnabled** and **ModelTransformEnabled** properties. If the tool is calibrated or model-based and the coordinate system is not **hsCoordinateSystemImage**, the property is expressed as **UnitsLength**. Otherwise, the property is expressed in pixels.

Example

```
Dim PointY as Single
PointY = PointFinder1.PointPositionY
```

PolarityMode

Description

Selects the type of polarity accepted for finding a point. Polarity identifies the change in greylevel values along the tool's X axis, in the positive direction.

Syntax

control.**PolarityMode** [= *Value* As hsPolarityMode]

Type

hsPolarityMode

Range

The valid range for this property is as follows:

hsPolarityMode	Value	Description
<i>hsDarkToLight</i>	0	The Point Finder searches only for point instances occurring at a dark to light transition in greylevel values.
<i>hsLightToDark</i>	1	The Point Finder searches only for point instances occurring at a light to dark transition in greylevel values.
<i>hsEither</i>	2	The Point Finder searches only for point instances occurring either at a light to dark or dark to light transition in greylevel values.
<i>hsDontCare</i>	3	The Point Finder searches only for point instances occurring at any transition in greylevel values including reversals in contrast at the point, for example on an unevenly colored background.

Remark(s)Enabling **hsEither** can significantly increase the tool's execution time.**Example**

```
' Search for a point at a light to dark transition
PointFinder1.PolarityMode = hsLightToDark
PointFinder1.Execute
```

PositioningLevel

Description

Configurable effort level of the instance positioning process. A value of 0 will provide coarser positioning and lower execution time. Conversely, a value of 10 will provide high accuracy positioning of Point instances.

Syntax

```
control.PositioningLevel [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: 10

Example

```
Dim PositioningLevel As Integer
PositioningLevel = PointFinder1.PositioningLevel
PointFinder2.PositioningLevel = 5
```

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax

control.**ProcessingFormat** [= *Value* As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Edge extraction will be perform using color information if a color input image is selected. Edge extraction will be performed using intensity information if a grey scale image is selected.
hsProcessingFormatGreyScale	1	Force grey scale processing. If a color input image is selected, only grey scale information will be used for edge extraction. Selecting this mode can decrease processing time of color image when appropriate.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SearchMode

Description

Specifies the method used by the tool to select a hypothesis.

Syntax

control.**SearchMode** [= *Value* as hsSearchMode]

Type

hsSearchMode

Range

The valid range for this property is as follows:

hsSearchMode	Value	Description
hsPointClosestToGuideline	1	Selects the point hypothesis closest to the Guideline.
hsPointWithMaximumNegativeXOffset	2	Selects the point hypothesis closest to the Search Area bound that is at maximum negative X offset.
hsPointWithMaximumPositiveXOffset	3	Selects the point hypothesis closest to the Search Area bound that is at maximum positive X offset.

Example

```
' Search for Point closest to Guideline
Dim Mode As hsSearchMode
PointFnder2.SearchMode = hsPointClosestToGuideline
```


Mode = PointFinder2.**SearchMode**

ShowEdges

Description

If true, edges in the Search Area are displayed in the Results display. Candidate edges for hypothesis generation are displayed in a different color (green) than other edges (blue).

Syntax

*control.***ShowEdges** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = PointFinder1.ShowEdges
PointFinder1.ShowEdges = False
```

ShowEntityFound

Description

When **ShowEntityFound** is set to true, the entity found by the tool is displayed in the Results display.

Syntax

*control.***ShowEntityFound** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = PointFinder1.ShowEntityFound
PointFinder1.ShowEntityFound = False
```

ShowImage

ShowGreyScaleImage

Description

When **ShowImage** is set to True, the input image is shown in the Results display.

Syntax

*control.***ShowImage** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
PointFinder1.ShowImage = True
```

ShowSearchArea

Description

Specifies if the Search Area is shown in the Results Viewer.

Syntax

control.**ShowSearchArea** [= *Value* As Boolean]

Type

Boolean

Range

Not applicable.

Example

```
Dim Enabled As Boolean
Enabled = PointFinder1.ShowSearchArea
PointFinder1.ShowSearchArea = False
```

SubsamplingLevel

Description

Subsampling level used to detect edges that are used by the tool to generate hypotheses. High values provide a coarser search and lower execution time than lower values.

Syntax

control.**SubsamplingLevel** [= *Value* As Integer]

Type

Integer

Range

Minimum: 1
Maximum: 8

Example

```
Dim SearchLevel As Single
SearchLevel = PointFinder1.SubsamplingLevel
PointFinder2.SubsamplingLevel = 4
```

ToolGuidelineOffset

Description

The X-axis offset of the Guideline marker from the center of the tool Search Area.

Syntax

control.**ToolGuidePointOffset** [= *Value* As Single]

Type

Single

Range

Maximum: $+0.5 \times \text{ToolWidth}$
Minimum: $-0.5 \times \text{ToolWidth}$

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```

PointFinder1.ToolWidth = 25
PointFinder1.ToolGuidelineOffset = -5
PointFinder1.Execute

```

ToolHeight

Description

Height of the tool Search Area, defined along the tool Y-axis.

Syntax

control.**ToolHeight** [= *Value* as Single]

Type

Single

Range

Greater than zero.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```

Dim Height as Single
Height = PointFinder1.ToolHeight
PointFinder2.ToolHeight = 15.2

```

ToolPositionX

Description

X coordinate of the position of the tool Search Area.

Syntax

control.**ToolPositionX** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```

Dim X_Pos As Single

```

```
X_Pos = PointFinder1.ToolPositionX
PointFinder2.ToolPositionX = 23.2
```

ToolPositionY

Description

Y coordinate of the position of the tool Search Area.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Remark(s)

The property is expressed in length units and with respect to a coordinate system specified in the following table:

Mode	Length unit	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Pixel	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	UnitsLength property	World
ModelTransformEnabled = True	UnitsLength property	Object (of the Model)

Example

```
Dim Y_Pos As Single
Y_Pos = PointFinder1.ToolPositionY
PointFinder2.ToolPositionY = 23.2
```

ToolRotation

Description

Angle of rotation of the tool Search Area.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the Model)

Example

```
Dim Angle As Single
Angle = PointFinder1.ToolRotation
PointFinder2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool Search Area, defined along the X-axis.

Syntax

control.**ToolWidth** [= Value As Single]

Type

Single

Range

Greater than 0.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width as Single
Width = PointFinder1.ToolWidth
PointFinder2.ToolWidth = 15.2
```

TransformRotation

Description

Rotation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.**TransformRotation** [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = PointFinder1.TransformRotation
PointFinder2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: Greater than zero.
Maximum: Unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = PointFinder1.TransformScale
PointFinder2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationX** property is expressed in length units, which are determined by the value of **UnitsLength** property.

Example

```
Dim X_Trans As Single
X_Trans = PointFinder1.TransformTranslationX
PointFinder2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the Object coordinate system transform into the World coordinate system of the input Image. This transformation is used to fit the tool to an object in the input Image. The different components of this transform are applied in the following order: Scale, Rotation and Translation. This property is handled as read only by the tool if the **InstanceSelectionEnabled** property is set to True.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = PointFinder1.TransformTranslationY
PointFinder2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSPointFinder".

Example

```
Dim ControlType As String
ControlType = PointFinder1.TypeName
```

UnitsLength

Description

Active length unit.

Syntax

*control.***UnitsLength** [= *Value* As hsUnitsLength]

Type

hsUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = PointFinder1.UnitsLength
PointFinder2.UnitsLength = hsInch
```

UnitsOrientation

Description

Active orientation unit.

Syntax

*control.***UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

hsUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
Unit = PointFinder1.UnitsOrientation
PointFinder2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = PointFinder1.Version
```

Methods

AboutBox

Description

Displays the Point Finder tool **About** box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

```
PointFinder1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the Point Finder process.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example

PointFinder1.**Execute**

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

GetProcessingFormatEnabled

Description

Returns whether or not specified processing format is enabled and available for selection.

Syntax

control.**GetProcessingFormatEnabled**(*ProcessingFormat* As hsProcessingFormat

Parameter(s)

Name	Description	Type	Range
<i>ProcessingFormat</i>	Format tested for availability.	hsProcessingFormat	see <i>ProcessingFormat</i> property

Return

Boolean

Range

Returns True if format enabled, otherwise False is returned.

Example

```
Dim Available As Boolean
Available = Tool.GetProcessingFormatEnabled _
    (hsProcessingFormatGreyScale)
```

HideProperties

Description

Hides the tool properties interface.

Syntax

```
control.HideProperties()
```

Parameter(s)

None

Return

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = PointFinder1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Displays the tool properties interface.

Syntax

```
control.ShowProperties(Modal As Boolean, [Variant Option])
```

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements		
Element	Value	Description
hsShowApply	1	Displays APPLY button
hsShowReset	2	Displays RESET button
hsShowDefault	4	Displays DEFAULT button
hsShowExecute	8	Displays EXECUTE button
hsShowOK	16	Displays OK button
hsExecuteOnOK	32	The tool process is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool process and the subsequent tools in the ProcessManager sequence are executed when the Execute button is clicked. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Dim Success As Boolean
Success = PointPointFinder1.ShowProperties _
    (True, hsShowOK + hsShowExecute)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Process Manager

Description

The Process Manager control is used to create, remove, re-order and execute sequences of HexSight processing controls. These processing controls cannot be executed outside of the Process Manager.

Properties

Database

Description

Reserved for internal use.

Description

Description

String that describes the standard usage of the control. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = ProcessManager1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the sequence of process. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = ProcessManager1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

On Error GoTo Handler

Handler:

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

FirstProcessName

Description

Name of the first process in the process list of the control. Returns an empty string when the process list is empty. Accessing this property resets the internal process name iterator to the first process in the list. This internal iterator can be used to browse the process names with the following properties: **FirstProcessName**, **LastProcessName**, **NextProcessName** and **PreviousProcessName**. Read only.

Syntax

control.**FirstProcessName**

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = ProcessManager1.FirstProcessName
ProcessManager1.FirstProcessName = "Process0"
```

InteractiveConfigurationEnabled

Description

Specifies if the properties of the instantiated processes can be modified by clicking the **Edit Process** button. Setting this property to True also generates the same effects As Setting the **ProcessListEditable** property to True regardless of the status of this latter property.

Syntax

control.**InteractiveConfigurationEnabled** [=Value As Boolean]

Type

Boolean

Range

True, False

Example

```
Dim Flag As Boolean
Flag = ProcessManager1.InteractiveConfigurationEnabled
ProcessManager1.InteractiveConfigurationEnabled = True
```

InterfaceVisible

Description

Specifies if the ProcessMananager interface is visible.

Syntax*control*.**InterfaceVisible** [= *Value* As Boolean]**Type**

Boolean

Range

True, False

Example

```
Dim VisibleFlag As Boolean
VisibleFlag = ProcessManager1.InterfaceVisible
ProcessManager1.InterfaceVisible = True
```

LastProcessName

Description

Name of the last process in the process list of the control. Returns an empty string when the process list is empty. Accessing this property resets the internal process name iterator to the last process in the list. This internal iterator can be used to browse the process names with the following properties: **FirstProcessName**, **LastProcessName**, **NextProcessName** and **PreviousProcessName**. Read only.

Syntax*control*.**LastProcessName****Type**

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = ProcessManager1.LastProcessName
```

Mode

Description

Mode of the ProcessManager. The Process Manager has two modes of operation. In the interactive mode, the process management interface is set according to the status of the **ProcessListEditable** and **InteractiveConfigurationEnabled** properties. In the running mode, this interface is disabled.

Syntax

control.**Mode** [=Value As hsProcessManagerMode]

Type

hsProcessManagerMode

Range

This accepted values for this property are:

Constant	Description	Value
hsModeRunning	ProcessListEditable property is set to False.	0
hsModeInteractive	All interactions with the control are enabled.	1

Remark(s)

When this property is first set to the running mode and later to the interactive mode, the **ProcessListEditable** property reverts to its original status.

Example

```
Dim ProcManMode As hsProcessManagerMode
ProcManMode = ProcessManager1.Mode
ProcessManager1.Mode = hsModeRunning
```

Name

Description

Name used to access the control.

Syntax

control.**Name** [=Value As String]

Type

String

Range

Not applicable.

Example

```
Dim CtrlName As String
CtrlName = ProcessManager1.Name = "ProcManA"
ProcessManager1.Name = "ProcManA"
```

NextProcessName

Description

Moves the internal process name iterator to the next process in the list and returns its name. Returns an empty string after the last process is reached. This internal iterator can be used to browse the process names with the following properties: **FirstProcessName**, **LastProcessName**, **NextProcessName** and **PreviousProcessName**. Read only.

Syntax

control.**NextProcessName**

Type

String

Range

Not applicable.

Remark(s)

This property returns an empty string if it is placed at the end of the process list.

Example

```
Dim ProcName As String
ProcName = ProcessManager1.NextProcessName
```

PreviousProcessName

Description

Moves the internal process name iterator to the previous process in the list and return its name. Returns an empty string after the first process is reached. This internal iterator can be used to browse the process names with the following properties: **FirstProcessName**, **LastProcessName**, **NextProcessName** and **PreviousProcessName**. Read only.

Syntax

control.**PreviousProcessName**

Type

String

Range

Not applicable.

Remark(s)

This property returns an empty string if it is placed at the beginning of the process list.

Example

```
Dim ProcName As String
ProcName = ProcessManager1.PreviousProcessName
```

Process

Description

Reference to a selected process. Use this property to access a process in the process list using its name or index. Read only.

Syntax

control.**Process** (*Process As Variant*)

Type

Object

Parameter(s)

Parameter	Description	Type	Range
<i>Process</i>	Process to access in the process list	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.

Range

Not applicable.

Example

```
Dim SomeProcess As Object
Set SomeProcess = ProcessManager1.Process ( 0 )
Dim Acquisition As HSAcquisitionDevice
```

```
Set Acquisition = ProcessManager1.Process("Acquisition")
```

ProcessAvailable

Description

Inquires if a specific process type is available in the current software license. If the process is not available, the property returns False.

Syntax

```
control.Process (ProcessTypeName As String) [= Value As Boolean]
```

Type

Object

Parameter(s)

Parameter	Description	Type	Range
<i>ProcessTypeName</i>	Type name of process to verify	String	Not applicable

Range

Not applicable.

Example

```
Dim Available As Boolean
Available = ProcessManager.ProcessAvailable("HSLocator")
```

ProcessCount

Description

Number of processes in the process list. Read only.

Syntax

```
control.ProcessCount
```

Type

Integer

Range

Minimum: 0

Maximum: unlimited

Example

```
Dim Number As Integer
Number = ProcessManager1.ProcessCount
```

ProcessIndex

Description

Index of the selected process in the process list. Read only.

Syntax

```
control.ProcessIndex ( ProcessName As String )
```

Type

Integer

Range

Minimum: 0

Maximum: **ProcessCount** - 1

Parameter(s)

Name	Description	Type	Range
<i>ProcessName</i>	Name of the selected process	String	Not applicable.

Example

```
Dim ProcessIndex As Integer
ProcessIndex = ProcessManager1.ProcessIndex("Acquisition")
```

ProcessListEditable

Description

Specifies if the list of the instantiated processes can be edited. When this property is set to False, the following buttons are disabled: **Add**, **Remove**, **Move Up**, **Move Down** and **Load Configuration**. Furthermore, in such a case, the list cannot be modified by programming. The properties of the instantiated processes can be edited and the configuration can be saved regardless of the status of this property.

If the **InteractiveConfigurationEnabled** property is set to False then the status of the **ProcessListEditable** property is not considered.

Syntax

control.**ProcessListEditable** [=Value As Boolean]

Type

Boolean

Range

True, False

Example

```
Dim Editable As Boolean
Editable = ProcessManager1.ProcessListEditable
ProcessManager1.ProcessListEditable = False
```

ProcessTypeName

Description

Type name of the selected process. Read only.

Syntax

control.**ProcessTypeName**(Process As Variant)

Type

String

Range

Not applicable.

Parameter(s)

Parameter	Description	Type	Range
<i>Process</i>	Selected process.	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.

Example

```
Dim ProcType As String
ProcType = ProcessManager1.ProcessTypeName
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = ProcessManager1.Version
```

Methods

AboutBox

Description

Displays the Process Manager control About box.

Syntax

control.**AboutBox**

Parameter(s)

None

Return

No return value.

Example

```
ProcessManager1.AboutBox
```

AddProcess

Description

Adds a new process to the list of instantiated processes.

Syntax

control.**AddProcess** (*ProcessTypeName* As String [, *ProcessName* As Variant]) As Object

Parameter(s)

Name	Description	Type	Range
<i>ProcessTypeName</i>	Type name of the process to add.	String	Not applicable.
<i>ProcessName</i>	Name of the process. Optional.	Variant	Since only the String type is supported the concept of range is not applicable.

Return

Object: reference to the newly added process.

Example

```
Dim NewProcess As HSPatternLocator
Set NewProcess = ProcessManager1.AddProcess ("HSPatternLocator",
"MyPatternLocator")
```

ControlPropertyExchange

Description

Reserved for internal use.

DuplicateProcess

Description

Duplicates the selected process and adds it as a new tool in the process list.

Syntax

control.**DuplicateProcess**(*Process* As Variant,[*ProcessName* As Variant]) As Object

Parameter(s)

Name	Description	Type	Range
<i>Process</i>	Selected process to be duplicated.	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.
<i>ProcessName</i>	Name of the of the newly duplicated process. Optional.	Variant	Since only String type is supported the concept of range is not applicable.

Return

Object: reference to the newly duplicated process.

Example

```
Dim DuplicateProcess As HSPatternLocator
```

```
Set DuplicateProcess = ProcessManager1.DuplicateProcess
("PatternLocator_1", "PatternLocator_2")
```

Execute

Description

Executes some or all the processes of the list of instantiated processes.

Syntax

```
control.Execute ( [ From As Variant ] [, To As Variant ] ) As Single
```

Parameter(s)

Name	Description	Type	Range
<i>From</i>	Name or index of the first process to execute. Optional.	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.
<i>To</i>	Last process to execute. Optional.	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.

Return

Single: time in milliseconds required to execute the processes.

Remark(s)

When no parameter is provided, all the processes in the process list are executed sequentially.

When only one parameter is provided, the process that corresponds to this parameter and all those following it are executed.

When two parameters are provided, the first parameter corresponds to the first process to be executed and the second one corresponds to the last process to be executed. All processes in the process list located between these two are executed sequentially as well.

Example

```
Dim TimeLength As Single
TimeLength = ProcessManager1.Execute
ProcessManager.Execute(0)
ProcessManager.Execute("AcquisitionDevice0", "Locator1")
```

ExportConfiguration

Description

Exports all processes or selection of processes, along with their configurations, from the current configuration to a specific configuration file (*.cfg).

Syntax

```
control.ExportConfiguration([FileName As Variant], [Selection As Variant]) As Boolean
```

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of configuration file to export. Optional. If no filename is specified, a standard File Save dialog box will appear.	Variant	Not applicable.
<i>Selection</i>	String representing the process to be exported. Optional.	Variant	String with numbers separated by ', ' - ' may be used to define a range. Example: "0-4,8,10" means that 0,1,2,3,4,8,10 are selected

Return

Boolean

Example

```
ApplicationControl.ProcessManager.ExportConfiguration , "0-4,8,10"
```

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideInterface

Description

Hides the ProcessManager interface.

Syntax

```
control.HideInterface ( ) [=Value As Boolean]
```

Parameter(s)

None

Return

Boolean

Range

Returns True if the interface was displayed when the method was called, otherwise False is returned.

Example

```
Dim WasDisplayed As Boolean
WasDisplayed = ProcessManager1.HideInterface
```

HideProperties

Description

Hides the properties interface of the control.

Syntax

```
control.HideProperties( )
```


Parameter(s)

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = ProcessManager1.HideProperties
```

ImportConfiguration

Description

Imports all processes or a selection of processes, along with their configuration, from a specific configuration file (*.cfg).

Syntax

```
control.ImportConfiguration(DestroyCurrent As Boolean, [FileName As Variant], [Mode As Variant], [Selection As Variant]) As Boolean
```

Parameter(s)

Name	Description	Type	Range
<i>DestroyCurrent</i>	If set to True, all existing processes will be removed before importing the configuration.	Boolean	True, False
<i>FileName</i>	Name of configuration file to export. Optional. If no filename is specified, a standard File Open dialog box will appear.	Variant	Not applicable.
<i>Mode</i>	Specifies the mode used to import the configuration. If no mode is specified, the hsImportModeAsk mode is used. Optional.	Variant	See table below.
<i>Selection</i>	String that defines selected processes that will be imported from the configuration file. Optional	Variant	String with numbers separated by ' , ' - ' may be used to define a range. Example: "0-4,8,10" means that processes 0,1,2,3,4,8,10 are selected.

hsImportMode	Description
hsImportModeSkip	Don't import the process if a process with the same name exists.
hsImportModeReplace	If a process with the same name already exists, replace the existing process with the selected process.
hsImportModeRename	If a process with the same name already exists, automatically rename and import the selected process.

hsImportMode	Description
hsImportModeAsk	Display a dialog box that inquires what Import Mode to use if a process with the same name exists: <i>Skip</i> , <i>Replace</i> or <i>Rename</i> .

Return

Boolean

Example

```
Dim Success As Boolean
Success = ProcessManager1.ImportConfiguration False
Success = _
    ProcessManager1.ImportConfiguration False, " ..\..\Hook.cfg ",
    hsImportModeReplace, " 0-4 "
```

InquireConfiguration

Description

Inquires information on the processes defined in a specific configuration file. This is useful for retrieving configuration file information before an **ImportConfiguration** call.

Syntax

```
control.InquireConfiguration([FileName As Variant]) As String
```

Parameter(s)

Name	Description	Type	Range
<i>FileName</i>	Name of configuration file to export. Optional. If no filename is specified, a standard File Open dialog box will appear.	Variant	Not applicable.

Return

String

Remark(s)

A typical String has the following format: Index\t Type\t Name\n. Example of a typical string:
"0\t HSAcquisitionDevice\t Acquisition\n 1\t HSLocator\t Locator\n"

Example

```
Dim lString As String
lString =
    ApplicationControl.ProcessManager.InquireConfiguration
```

LoadConfiguration

Description

Loads a new set of processes in the process list, along with their configuration, from a configuration file.

Syntax

```
control.LoadConfiguration(FileHandle As OLE_HANDLE, FileName As String) As Boolean
```

Parameter(s)

Parameter	Description	Type	Range
<i>FileHandle</i>	OLE_HANDLE to the file to be loaded.	OLE_HANDLE	Always set to zero.
<i>FileName</i>	Name of the file to be loaded.	String	Not applicable. If the FileName parameter is not provided, a file open dialog appears.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = ProcessManager1.LoadConfiguration(0,"")
Success = ProcessManager1.LoadConfiguration _
    (0,"..\..\Myconfig.cfg")
```

MoveProcess

Description

Moves a selected process to a new position in the process list.

Syntax

control.**MoveProcess** (*Process* As Variant, *NewIndex* As Integer) As Integer

Parameter(s)

Name	Description	Type	Range
<i>Process</i>	Selected process	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.
<i>NewIndex</i>	Position of the selected process in the process list after being moved.	Integer	Minimum: 0 Maximum: ProcessCount - 1

Return

Integer: New index of the process

Example

```
Dim NewPos As Integer
NewPos = ProcessManager1.MoveProcess(0,1)
ProcessManager1.MoveProcess("MyCaliper",4)
```

Refresh

Description

Refreshes the user interface of the ProcessManager.

Syntax*control.Refresh***Parameter(s)**

None

Return

No return value.

Example`ProcessManager1.Refresh`

RemoveProcess

Description

Removes a selected process from the process list.

Syntax*control.RemoveProcess (Process As Variant) As Boolean***Parameter(s)**

Name	Description	Type	Range
<i>Process</i>	Index or name of the selected process.	Variant	Integer: Minimum: 0 Maximum: ProcessCount - 1 String: Range not applicable. Other types not supported.

Return

Boolean

Value	Description
False	The process has not been removed.
True	The process has been removed.

Example

```
Dim BeenRemoved As Boolean
BeenRemoved = ProcessManager1.RemoveProcess ( 0 )
ProcessManager1.RemoveProcess("MyLocator")
```

Reset

Description

Empties the process list.

Syntax*control.Reset***Parameter(s)**

None

Return

Nothing

Example`ProcessManager1.Reset`

SaveConfiguration

Description

Saves the current set of processes, along with their configuration, in a configuration file.

Syntax

control.**SaveConfiguration**(*FileHandle* As OLE_HANDLE, *FileName* As String) [= *Value* As Boolean]

Parameter(s)

Parameter	Description	Type	Range
<i>FileHandle</i>	OLE_HANDLE to the file to be loaded.	OLE_HANDLE	Always zero.
<i>FileName</i>	Name of the file to be saved.	String	Not applicable. If the FileName parameter is not provided, a file open dialog appears.

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Success As Boolean
Success = ProcessManager1.SaveConfiguration(0, "")
Success = ProcessManager1.SaveConfiguration _
    (0, "..\..\Myconfig.cfg")
```

SetDebugParameter

Description

Reserved for internal use.

ShowInterface

Description

Displays the user interface of the ProcessManager control.

Syntax

control.**ShowInterface**(*Modal* As Boolean, [*DisplayTopMost* As Variant]) As Boolean

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	True, False
<i>DisplayTopMost</i>	When True, the Process manager interface window is displayed topmost. Optional	Variant	True, False

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeeded As Boolean
Succeeded = ProcessManager1.ShowInterface (False)
```

ShowProperties

Description

Displays the control properties interface.

Syntax

```
control.ShowProperties(Modal As Boolean) As Boolean
```

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	True, False

Return

Type	Description	Range
Boolean	Specifies if the operation was successful.	True, False

Example

```
Dim Succeeded As Boolean
Succeeded = ProcessManager1.ShowProperties(True)
```

Events

AskDatabase

Description

Reserved for internal use.

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

ModeChange

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

Sampling Tool

Description

The Sampling Tool outputs an image sampled from a region of interest defined on a larger image. Dimensional and positional information is computed from the sampled image.

Properties

BilinearInterpolationEnabled

Description

Specifies if bilinear interpolation is used to sample the image.

Syntax

control.**BilinearInterpolationEnabled** [= *Value* As Boolean]

Type

Boolean

Range

Value	Description
True	Bilinear interpolation is enabled.
False	Bilinear interpolation is disabled.

Remark(s)

Bilinear interpolation increases tool accuracy but slows the sampling process.

Example

Dim Result As Boolean

Result = Sampling1.**BilinearInterpolationEnabled**

Sampling2.**BilinearInterpolationEnabled** = True

CalibratedImageHeight

Description

Height of the sampled image expressed using the units specified by the **UnitsLength** property. Read only.

Syntax

control.**CalibratedImageHeight**

Type

Single

Range

Greater than zero.

Remark(s)

This property is equal to **ImageWidth** * **PixelHeight** and is therefore subject to the same validity conditions as the **PixelHeight** property.

Example

```
Dim Value As Single
Value = Sampling1.CalibratedImageHeight
```

CalibratedImageWidth

Description

Width of the sample image expressed in the unit specified by the **UnitsLength** property. Read only.

Syntax

```
control.CalibratedImageWidth
```

Type

Single

Range

Greater than zero.

Remark(s)

This property is equal to **ImageWidth** * **PixelWidth** and is therefore subject to the same validity conditions as the **PixelWidth** property.

Example

```
Dim Value As Single
Value = Sampling1.CalibratedImageWidth
```

CalibratedUnitsEnabled

Description

Specifies if the tool uses calibrated units to express dimensions.

Syntax

```
control.CalibratedUnitsEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	Dimensions are expressed in calibrated units.
False	Dimensions are expressed in pixel units.

Remark(s)

If the **ModelTransformEnabled** property is set to True (the tool is model-based), this parameter is overridden and all dimensions are expressed in calibrated units.

Example

```
Dim Result As Boolean
Result = Sampling1.CalibratedUnitsEnabled
Sampling2.CalibratedUnitsEnabled = False
```

CoordinateSystem

Description

Coordinate system used by the tool to return its results.

Syntax

control.**CoordinateSystem** [= Value As hsCoordinateSystem]

Type

hsCoordinateSystem

Range

The valid range for this property is as follows:

Constant	Value	Description
hsCoordinateSystemImage	0	Results are expressed in pixel units with respect to the Image coordinate system.
hsCoordinateSystemWorld	1	Results are expressed with respect to the coordinate system used to calibrate the input image if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, no results are output.
hsCoordinateSystemObject	2	Results are expressed in calibrated units with respect to the Object coordinate system if and only if the tool is model-based (the ModelTransformEnabled property is set to True). Otherwise, no results are output.
hsCoordinateSystemTool	3	Results are expressed in calibrated units with respect to the Tool coordinate system if the tool uses calibrated units to express dimensions (ModelTransformEnabled or CalibratedUnitsEnabled properties are set to True). Otherwise, results are expressed in pixels with respect to Tool coordinate system.

Example

```
Dim Result As hsCoordinateSystem
Result = Sampling1.CoordinateSystem
Sampling1.CoordinateSystem = hsCoordinateSystemObject
```

Database

Description

Reserved for internal use.

DatabasePassword

Description

Reserved for internal use.

Description

Description

String that describes the standard use of the tool. Read only.

Syntax

control.**Description**

Type

String

Range

Not applicable.

Example

```
Dim Description As String
Description = Sampling1.Description
```

ElapsedTime

Description

Elapsed time (in milliseconds) of the last execution of the tool. Read only.

Syntax

control.**ElapsedTime**

Type

Single

Range

Minimum: 0.0

Maximum: unlimited

Remark(s)

This property gives the real elapsed time, not the CPU time used.

Example

```
Dim TimeSpent As Single
TimeSpent = Sampling1.ElapsedTime
```

ErrorObject

Description

Returns a reference to the internal **HSErrorObject** object of the control. It is used for handling errors in an application. Read only.

Syntax

control.**ErrorObject**

Type

HSErrorObject

Range

Not applicable.

Example

```
On Error GoTo Handler
```

```
Handler:
```

```
MsgBox "Error: " & Locator.ErrorObject.Description
```

ImageBottomLeftX

Description

X coordinate of the bottom left corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax*control.ImageBottomLeftX***Type**

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single

Value = Sampling1.**ImageBottomLeftX**

ImageBottomLeftY

Description

Y coordinate of the bottom left corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax*control.ImageBottomLeftY***Type**

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single

Value = Sampling1.**ImageBottomLeftY**

ImageBottomRightX

Description

X coordinate of the bottom right corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax

control.**ImageBottomRightX**

Type

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single

Value = Sampling1.**ImageBottomRightX**

ImageBottomRightY

Description

Y coordinate of the bottom right corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax

control.**ImageBottomRightY**

Type

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single
Value = Sampling1.**ImagebottomRightY**

ImageHeight

Description

Height of the sampled image expressed in pixels. Read only.

Syntax

*control.***ImageHeight**

Type

Integer

Range

Any positive Integer value.

Example

Dim Value As Integer
Value = Sampling1.**ImageHeight**

ImageTopLeftX

Description

X coordinate of the top left corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax

*control.***ImageTopLeftX**

Type

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single
Value = Sampling1.**ImageTopLeftX**

ImageTopLeftY

Description

Y coordinate of the top left corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax*control.ImageTopLeftY***Type**

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single

Value = Sampling1.**ImageTopLeftY**

ImageTopRightX

Description

X coordinate of the top right corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax*control.ImageTopRightX***Type**

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single

Value = Sampling1.**ImageTopRightX**

ImageTopRightY

Description

Y coordinate of the top right corner of the sampled image expressed with respect to the coordinate system specified by the **CoordinateSystem** property. Read only.

Syntax

control.**ImageTopRightY**

Type

Single

Range

Unbounded

Remark(s)

The units used to express this property depends on the positioning mode of the tool as displayed in the following table:

Positioning mode	Units
Image-Based Uncalibrated	Pixel.
Image-Based Calibrated	As specified by the UnitsLength property.
Manual Model-Based	As specified by the UnitsLength property.
Automatic Model-Based	As specified by the UnitsLength property.

Example

Dim Value As Single

Value = Sampling1.**ImageTopRightY**

ImageWidth

Description

Width of the sampled image expressed in pixels. Read only.

Syntax

control.**ImageWidth**

Type

Integer

Range

Any positive Integer value.

Example

Dim Value As Integer

Value = Sampling1.**ImageWidth**

InputImage

InputGreyScaleImage

Description

Name of the input image.

Syntax

control.**InputImage** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Sampling1.InputImage
Sampling2.InputImage = "image1"
```

InputView

InputGreyScaleView

Description

Name of the View from which the image is retrieved. See **InputImage** for more details.

Syntax

control.**InputView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Sampling1.InputView
Sampling2.InputView = "view0"
```

InputInstanceScene

Description

Name of the Input Instance Scene.

Syntax

control.**InputInstanceScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim SceneName As String
SceneName = Sampling1.InputInstanceScene
Sampling2.InputInstanceScene = "scene0"
```

InputInstanceView

Description

Name of the View containing the Input Instance Scene.

Syntax

control.**InputInstanceView** [= Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Sampling1.InputInstanceView
Sampling2.InputInstanceView = "view0"
```

InputModelScene

Description

Name of the Input Model Scene.

Syntax

control.**InputModelScene** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelSceneName As String
ModelSceneName = Sampling1.InputModelScene
Sampling2.InputModelScene = "model_scene0"
```

InputModelView

Description

Name of the View containing the input Model Scene.

Syntax

control.**InputModelView** [=Name As String]

Type

String

Range

Not applicable.

Example

```
Dim ModelViewName As String
ModelViewName = Sampling1.InputModelView
Sampling2.InputModelView = "view0"
```

InstanceSelectionEnabled

Description

If **ModelTransformEnabled** is set to TRUE, this parameter indicates if the Instance Transform is to be specified by the user or if it must be extracted from an Instance Scene as output by the HSLocator.

Syntax

control.**InstanceSelectionEnabled** [= Value As Boolean]

Type

Boolean

Range

Value	Description
True	Instance transform parameters are automatically copied from the Instance Scene.
False	Instance transform parameters are user-controlled.

Example

```
Dim Enabled As Boolean
Enabled = Sampling1.InstanceSelectionEnabled
Sampling2.InstanceSelectionEnabled = True
```

InstanceSelectionIndex

Description

Index of the instance of the Instance Scene used to specify the model transform if **ModelTransformEnabled** and **InstanceSelectionEnabled** are TRUE.

Syntax

```
control.InstanceSelectionIndex [= Value As Integer]
```

Type

Integer

Range

Minimum: 0

Maximum: Number of instances found by the Locator minus 1

Example

```
Dim Ndx As Integer
Ndx = Sampling1.InstanceSelectionIndex
Sampling2.InstanceSelectionIndex = 0
```

ModelTransformEnabled

Description

Boolean property used to specify if the tool is Model-based or not.

Syntax

```
control.ModelTransformEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
True	The control is model-based.
False	The control is not model-based.

Example

```
Dim Enabled As Boolean
Enabled = Sampling1.ModelTransformEnabled
Sampling2.ModelTransformEnabled = True
```

Name

Description

Name used to access the tool through the Process Manager.

Syntax

control.**Name** [= Value As String]

Type

String

Range

Not applicable.

Example

```
Dim ProcName As String
ProcName = Sampling1.Name
Sampling2.Name = "EdgeLocA"
```

OutputImage

OutputGreyScaleImage

Description

Name of the Output image.

Syntax

control.**OutputImage**

Type

String

Range

Not applicable.

Example

```
Dim ImageName As String
ImageName = Sampling1.OutputImage
Sampling2.OutputImage = "Sampled Image"
```

OutputView

OutputGreyScaleView

Description

Name of the View containing the Output image.

Syntax

control.**OutputView**

Type

String

Range

Not applicable.

Example

```
Dim ViewName As String
ViewName = Sampling1.OutputView
Sampling2.OutputView = "Sampling View"
```

PixelHeight

Description

Height of a pixel of the sampled image. Read only.

Syntax

control.**PixelHeight**

Type

Single

Range

Any positive Single value.

Remark(s)

The status of the **PixelHeight** property depends on the tool's positioning mode and the **CoordinateSystem** property as displayed in the following tables:

Image Based Uncalibrated positioning mode

Coordinate System	Units	PixelHeight
Image	UnitsLength	1.0
World	N/A	N/A
Object	N/A	N/A
Tool	UnitsLength	1.0

Image Based Calibrated positioning mode

Coordinate System	Units	PixelHeight
Image	N/A	N/A
World	UnitsLength	SamplingStep
Object	N/A	N/A
Tool	UnitsLength	SamplingStep

Automatic or Manual Model Based positioning mode

Coordinate System	Units	PixelHeight
Image	N/A	N/A
World	UnitsLength	SamplingStep
Object	UnitsLength	SamplingStep
Tool	UnitsLength	SamplingStep

Under specific conditions, the property is not applicable (N/A in the tables) and should not be used when these conditions are met.

Example

Dim Value As Single

Value = Sampling1.**PixelHeight**

PixelWidth

Description

Width of a pixel of the sampled image. Read only.

Syntax*control.PixelWidth***Type**

Single

Range

Any positive Single value.

Remark(s)

The status of the **PixelWidth** property depends on the tool's positioning mode and the **CoordinateSystem** property as displayed in the following tables:

Image Based Uncalibrated positioning mode

CoordinateSystem	Units	PixelWidth
Image	UnitsLength	1.0
World	N/A	N/A
Object	N/A	N/A
Tool	UnitsLength	1.0

Image Based Calibrated positioning mode

Coordinate System	Units	PixelWidth
Image	N/A	N/A
World	UnitsLength	SamplingStep
Object	N/A	N/A
Tool	UnitsLength	SamplingStep

Automatic or Manual Model Based positioning mode

Coordinate System	Units	PixelWidth
Image	N/A	N/A
World	UnitsLength	SamplingStep
Object	UnitsLength	SamplingStep
Tool	UnitsLength	SamplingStep

Under specific conditions, the property is not applicable (N/A in the tables) and should not be used when these conditions are met.

Example

Dim Value As Single

Value = Sampling1.**PixelWidth**

ProcessingFormat

Description

Selects the image format used to process the input image.

Syntax*control.ProcessingFormat* [= Value As hsProcessingFormat]

Type

hsProcessingFormat

Range

The valid range for this property is as follows:

Constant	Value	Description
hsProcessingFormatNative	0	Uses native input image format for processing. Resulting sampled image will be of the same format as the selected input image.
hsProcessingFormatGreyScale	1	Force grey scale processing. Resulting sampled image will be a grey scale image.

Example

```
Dim Format As hsProcessingFormat
Format = Tool.ProcessingFormat
Tool.ProcessingFormat = hsProcessingFormatGreyScale
```

SamplingStep

Description

Sampling step used by the tool in the last execution.

Syntax*control*.**SamplingStep****Type**

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

The sampling step can either be default or custom, depending on the value of the **SamplingStepCustomEnabled** property.

Example

```
Dim Step As Single
Step = Sampling1.SamplingStep
```

SamplingStepCustom

Description

If **SamplingStepCustomEnabled** is TRUE, this sampling step is used to sample the image instead of **SamplingStepDefault**.

Syntax*control*.**SamplingStepCustom** [= Value As Single]**Type**

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the input image.

Remark(s)

A custom sampling step is usually not recommended.

Example

```
Dim Step As Single
Step = Sampling1.SamplingStepCustom
Sampling2.SamplingStepCustom = 2.1
```

SamplingStepCustomEnabled

Description

If enabled, the tool will use the user-defined sampling step (**SamplingStepCustom**) instead of using a best sampling-step estimation.

Syntax

```
control.SamplingStepCustomEnabled [= Value As Boolean]
```

Type

Boolean

Range

Value	Description
False	The tool uses the default sampling step.
True	The default sampling step is overridden by SamplingStepCustom.

Example

```
Dim Enabled As Boolean
Enabled = Sampling1.SamplingStepCustomEnabled
Sampling2.SamplingStepCustomEnabled = True
```

SamplingStepDefault

Description

Best sampling step computed by the tool to sample the image. Automatically used by the tool if **SamplingStepCustomEnabled** is TRUE.

Syntax

```
control.SamplingStepDefault
```

Type

Single

Range

Minimum: greater than zero.
Maximum: boundaries of the input image.

Example

```
Dim Sampling As Single
Sampling = Sampling1.SamplingStepDefault
```

ToolHeight

Description

Height of the tool.

Syntax

control.**ToolHeight** [= *Value* As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Example

Dim Height As Single

Height = Sampling1.**ToolHeight**

Sampling2.**ToolHeight** = 23.4

ToolPositionX

Description

X coordinate of the tool location on either the Input image or the Input Model Scene.

Syntax

control.**ToolPositionX** [= *Value* As Single]

Type

Single

Range

Unbounded

Example

Dim X_Pos As Single

X_Pos = Sampling1.**ToolPositionX**

Sampling2.**ToolPositionX** = 23.2

ToolPositionY

Description

Y coordinate of the tool location on either Input image or the Input Model Scene.

Syntax

control.**ToolPositionY** [= *Value* As Single]

Type

Single

Range

Unbounded

Example

Dim Y_Pos As Single

Y_Pos = Sampling1.**ToolPositionY**

Sampling2.**ToolPositionY** = 23.2

ToolRotation

Description

Rotation of the tool.

Syntax

control.**ToolRotation** [= *Value* As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property:

UnitsOrientation property value	ToolRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

The property is expressed with respect to a coordinate system specified in the following table:

Mode	Coordinate system
CalibratedUnitsEnabled = False ModelTransformEnabled = False	Image
CalibratedUnitsEnabled = True ModelTransformEnabled = False	World
ModelTransformEnabled = True	Object (of the model)

Example

```
Dim Angle As Single
Angle = Sampling1.ToolRotation
Sampling2.ToolRotation = -10.0
```

ToolWidth

Description

Width of the tool.

Syntax

control.**ToolWidth** [= Value As Single]

Type

Single

Range

Minimum: greater than zero.

Maximum: boundaries of the image.

Remark(s)

This property is expressed in the calibrated units set by the **UnitsLength** property if the tool uses calibrated units to express dimensions (**ModelTransformEnabled** or **CalibratedUnitsEnabled** properties set to True). Otherwise, it is expressed in pixels.

Example

```
Dim Width As Single
Width = Sampling1.ToolWidth
Sampling2.ToolWidth = 23.4
```

TransformRotation

Description

Y translation component of the transform applied to the model if **ModelTransformEnabled** is TRUE.

Syntax

control.TransformRotation [= Value As Single]

Type

Single

Range

The range of this property depends on the value of the **UnitsOrientation** property.

UnitsOrientation property value	TransformRotation property range
hsDegree	Minimum: -180 deg Maximum: +180 deg
hsRadian	Minimum: -Pi rad Maximum: +Pi rad

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Angle As Single
Angle = Sampling1.TransformRotation
Sampling2.TransformRotation = 22.0
```

TransformScale

Description

Scale component of the transform applied to the model if **ModelTransformEnabled** is TRUE.

Syntax

control.TransformScale [= Value As Single]

Type

Single

Range

Minimum: greater than zero.
Maximum: unlimited.

Remark(s)

This property is only applicable if the **ModelTransformEnabled** property is set to True.

Example

```
Dim Scale As Single
Scale = Sampling1.TransformScale
Sampling2.TransformScale = 1.10
```

TransformTranslationX

Description

X translation component of the transform applied to the model if **ModelTransformEnabled** is TRUE.

Syntax

control.TransformTranslationX [= Value As Single]

Type

Single

Range

Unbounded

Example

```
Dim X_Trans As Single
X_Trans = Sampling1.TransformTranslationX
Sampling2.TransformTranslationX = 22.0
```

TransformTranslationY

Description

Y translation component of the transform applied to the model if **ModelTransformEnabled** is TRUE.

Syntax

control.TransformTranslationY [= Value As Single]

Type

Single

Range

Unbounded

Remark(s)

This property is only used if the **ModelTransformEnabled** property is set to True. The **TransformTranslationY** property is expressed in length units, which are determined by the value of the **UnitsLength** property.

Example

```
Dim Y_Trans As Single
Y_Trans = Sampling1.TransformTranslationY
Sampling2.TransformTranslationY = 12.2
```

TypeName

Description

ActiveX type name for the tool. Read only.

Syntax

control.TypeName

Type

String

Range

Not applicable.

Remark(s)

This property is always "HSSamplingTool".

Example

```
Dim ControlType As String
ControlType = Sampling1.TypeName
```

UnitsLength

Description

Unit used to express lengths and coordinates for the interaction between the user of the tool (either by GUI or programming) and the tool if the tool is either calibrated or model-based.

Syntax

control.**UnitsLength** [= *Value* As hsUnitsLength]

Type

HSUnitsLength

Range

The valid range for this property is as follows:

Constant	Value	Description
hsMeter	0	meter
hsCentimeter	1	centimeter
hsMillimeter	2	millimeter
hsMicron	3	micron (1E-6 meter)
hsYard	4	yard
hsFoot	5	foot
hsInch	6	inch
hsMil	7	one thousandth of an inch
hsCalibrationUnit	8	one calibration unit

Example

```
Dim Unit As hsUnitsLength
Unit = Sampling1.UnitsLength
Sampling2.UnitsLength = hsInch
```

UnitsOrientation

Description

Unit used to express orientation for the interaction between the user of the tool (either by GUI or programming) and the tool.

Syntax

control.**UnitsOrientation** [= *Value* As hsUnitsOrientation]

Type

HSUnitsOrientation

Range

The valid range for this property is as follows:

Constant	Value	Description
hsRadian	0	radian
hsDegree	1	degree

Example

```
Dim Unit As hsUnitsOrientation
```

```
Unit = Sampling1.UnitsOrientation
Sampling2.UnitsOrientation = hsDegree
```

Version

Description

String containing the current HexSight version and build. Read only.

Syntax

control.**Version**

Type

String

Range

Not applicable.

Remarks

Example of a typical return string: "3.3.000 R 3371 (release)".

Example

```
Dim Version As String
Version = Sampling1.Version
```

Methods

AboutBox

Description

Displays the tool About box.

Syntax

control.**AboutBox**()

Parameter(s)

None

Range

No return value.

Example

```
Sampling1.AboutBox
```

ControlPropertyExchange

Description

Reserved for internal use.

Execute

Description

Executes the tool.

Syntax

control.**Execute**()

Parameter(s)

None

Return

Single: Processing time in milliseconds.

Example`Sampling1.Execute`

GetDebugParameter

Description

Reserved for internal use.

GetDispatchPointer

Description

Reserved for internal use.

HideProperties

Description

Hides the property pages of the tool if they are displayed.

Syntax`control.HideProperties( )`**Parameter(s)**

None

Return

Type	Description	Range
Boolean	Specifies if the interface was displayed when the method was called.	True, False

Example

```
Dim InterfaceWasDisplayed As Boolean
InterfaceWasDisplayed = Sampling1.HideProperties
```

LoadConfiguration

Description

Reserved for internal use.

SaveConfiguration

Description

Reserved for internal use.

SetDebugParameter

Description

Reserved for internal use.

ShowProperties

Description

Shows the property pages of the tool.

Syntax

control.**ShowProperties**(*Modal* As Boolean, [*Variant Option*])

Parameter(s)

Name	Description	Type	Range
<i>Modal</i>	A window is modal when it must be closed before the application can continue.	Boolean	False, True
<i>Option</i>	Optional bitfield parameter. It specifies which buttons are displayed in the interface and the behavior of the control when the OK button is clicked. When this parameter is not provided, all the bitfield elements are set except hsExecuteOnOK and hsExecuteRemaining .	Variant	The elements of the bitfield are displayed in the following table.

Option parameter: bitfield elements

Element	Value	Description
hsShowApply	1	Displays APPLY button.
hsShowReset	2	Displays RESET button.
hsShowDefault	4	Displays DEFAULT button.
hsShowExecute	8	Displays EXECUTE button.
hsShowOK	16	Displays OK button.
hsExecuteOnOK	32	The tool is executed when the OK button is clicked on the interface.
hsExecuteRemaining	64	The tool and all those following in the Process Manager sequence are executed when the Execute button is clicked in the interface. Clicking OK can generate the same effects provided that the hsExecuteOnOK and hsExecuteRemaining bitfields are both set.

Return

Boolean

Range

Returns True if operation succeeds, otherwise False is returned.

Example

```
Sampling1.ShowProperties(True, hsShowOK + hsShowExecute)
```

Events

DatabaseChange

Description

Reserved for internal use.

Done

Description

Reserved for internal use.

Error

Description

Reserved for internal use.

PropertiesChange

Description

Reserved for internal use.

SingleExecute

Description

Reserved for internal use.

Index

A

AboutBox 171, 249, 293, 323, 359, 403, 470, 496, 537, 553, 566, 583, 612, 635, 647, 688, 707, 739, 795, 862, 918, 930, 975, 1005, 1024, 1047, 1082, 1165, 1232, 1261, 1291, 1303, 1336

AboutBoxCustom 599

AdaptiveThresholdEnabled 749

AddArc 583, 707

AddAxesMarker 862

AddCircleShape 612

AddCircularArcMarker 863

AddColorFilter 739

AddImage 583, 707

AddLabelMarker 864

AddLine 584, 708

AddLineMarker 865

AddModel 1165

AddModelEdgePath 1202

AddPair 293, 538

AddPoint 585, 709

AddPointArrayMarker 866

AddPointMarker 867

AddProcess 1304

AddRectangleMarker 868

AddRectangleShape 613

AddReferencePoint 1203

AddScene 709

AddTargetMarker 869

AddView 585

AdvancedParametersVisible 1087

AnalyzeModel 1204

AnalyzeModelClick 1211

Angle 627, 647

AngleVariation 750

AngleWithinRange 553

Appearance 801

AppendImageToEmulationDatabase 171

Apply 1204, 1218

ApplyClick 1211, 1221

ApplyTransform 614

Arc 570, 694

ArcAverageContrast 327

ArcCenterPointPositionX 327

ArcCenterPointPositionY 328

ArcCount 694

ArcEndAngle 328

ArcEndPointPositionX 328

ArcEndPointPositionY 329

ArcExists 695

ArcFitQuality 329

ArcFound 330

ArcIndex 695

ArcMatchQuality 330

ArcMustBeTotallyEnclosed 330

ArcName 696

ArcRadius 331

ArcStartAngle 331

ArcStartPointPositionX 331

ArcStartPointPositionY 332

ArcTo 615

ArithmeticClippingMode 1009

ArithmeticConstant 1009

ArithmeticScale 1010

AskDatabase 198, 252, 297, 326, 362, 406, 473, 498, 541, 747, 921, 933, 979, 1007, 1026, 1050, 1085, 1184, 1235, 1266, 1294, 1312

AssignmentConstant 1010

AssignmentHeight 1010

AssignmentWidth 1011

AttachImagePointer 616

AutoCoarsenessSelectionEnabled 1237

AutomaticCandidateCountEnabled 1028

AutomaticLevels 1185

AutoPanEnabled 801

AverageElementSize 750

AverageImageError 199

AveragePixelHeight 199, 219, 232, 242, 557, 601, 651

AveragePixelSize 557

AveragePixelWidth 200, 219, 232, 557, 601, 651

AverageWorldError 200

AxesMarkerChange 886

AxesMarkerConstraints 802

AxesMarkerLabel1 802

AxesMarkerLabel2 803

AxesMarkerLength1 803

AxesMarkerLength2 804

AxesMarkerMidAngle 805

AxesMarkerRotation 805

AxisXPositionX 200

AxisXPositionY 201

AxisYPositionX 202

AxisYPositionY 202

B

BackgroundColor 602

BackgroundImage 602

BarcodeAngle 363

BarcodeCornerX 363

BarcodeCornerY 364

BarcodeDecodeTime 365

BarcodeFoundCount 366

BarcodeStatusCode 367
 BarcodeStatusCodeString 368
 BarcodeString 369
 BarcodeToDecode 370
 BarcodeType 370
 BarcodeTypeEnabled 372
 BarcodeTypeString 373
 BilinearInterpolationEnabled 254, 298, 373, 407, 500, 717,
 751, 895, 934, 981, 1028, 1237, 1314
 BlobArea 407
 BlobBoundingBoxBottom 408
 BlobBoundingBoxCenterX 409
 BlobBoundingBoxCenterY 409
 BlobBoundingBoxHeight 410
 BlobBoundingBoxLeft 411
 BlobBoundingBoxRight 411
 BlobBoundingBoxRotation 412
 BlobBoundingBoxTop 413
 BlobBoundingBoxWidth 413
 BlobChainCode 414
 BlobChainCodeDeltaX 415
 BlobChainCodeDeltaY 415
 BlobChainCodeLength 416
 BlobChainCodeStartX 416
 BlobChainCodeStartY 417
 BlobConvexPerimeter 418
 BlobCount 418
 BlobElongation 419
 BlobExtentBottom 419
 BlobExtentLeft 420
 BlobExtentRight 421
 BlobExtentTop 421
 BlobGreyLevelMaximum 422
 BlobGreyLevelMean 422
 BlobGreyLevelMinimum 423
 BlobGreyLevelRange 423
 BlobGreyLevelStdDev 424
 BlobHoleCount 424
 BlobInertiaMaximum 425
 BlobInertiaMinimum 425
 BlobInertiaXAxis 426
 BlobInertiaYAxis 427
 BlobIntrinsicBoundingBoxBottom 428
 BlobIntrinsicBoundingBoxCenterX 428
 BlobIntrinsicBoundingBoxCenterY 429
 BlobIntrinsicBoundingBoxHeight 430
 BlobIntrinsicBoundingBoxLeft 431
 BlobIntrinsicBoundingBoxRight 431
 BlobIntrinsicBoundingBoxRotation 432
 BlobIntrinsicBoundingBoxTop 433
 BlobIntrinsicBoundingBoxWidth 433
 BlobIntrinsicExtentBottom 434

BlobIntrinsicExtentLeft 435
 BlobIntrinsicExtentRight 436
 BlobIntrinsicExtentTop 436
 BlobPositionX 437
 BlobPositionY 438
 BlobPrincipalAxesRotation 438
 BlobRawPerimeter 439
 BlobRoundness 440
 BorderIntegrity 751
 BorderStyle 806
 BoundaryBottom 603, 652
 BoundaryLeft 603, 652
 BoundaryRight 603, 652
 BoundaryTop 604, 653
 BoundingAreaBottom 1185
 BoundingAreaChange 1211
 BoundingAreaLeft 1186
 BoundingAreaRight 1186
 BoundingAreaTop 1187
 Brightness 806
 BuildModel 1204
 BuildModelClick 1211

C

Calibrate 616
 Calibrated 544, 628
 CalibratedImageHeight 1314
 CalibratedImageWidth 1315
 CalibratedUnitsChange 886
 CalibratedUnitsEnabled 254, 298, 332, 374, 440, 500, 717,
 751, 807, 895, 934, 981, 1029, 1052, 1087, 1238, 1267,
 1315
 Calibration 59, 545, 604, 629, 640, 653
 CalibrationConfigurationName 59
 CalibrationDescription 60
 CalibrationDotCount 203
 CalibrationDotPositionXImage 203
 CalibrationDotPositionXWorld 204
 CalibrationDotPositionYImage 204
 CalibrationDotPositionYWorld 205
 CalibrationMethod 60
 CalibrationMethodName 61
 CalibrationParameter 62
 CalibrationParameterCount 64
 CalibrationParameterDescription 65
 CalibrationParameterEditable 66
 CalibrationParameterName 67
 CalibrationPointer 558
 CancelClick 1211
 CandidatePointsCount 1029
 CenterPoint 545
 ChainCodeResultsEnabled 441
 ChangeAxesMarker 870

- ChangeCircularArcMarker 871
- ChangeLabelMarker 872
- ChangeLineMarker 873
- ChangePointArrayMarker 874
- ChangePointMarker 875
- ChangeRectangleMarker 876
- ChangeTargetMarker 876
- CharacterConfidence 935
- CharacterCornerX 935
- CharacterCornerY 937
- CharacterHeight 938
- CharactersOnRow 939
- CharacterSpacing 939
- CharacterWidth 940
- ChecksumEnabled 374
- Circle 546
- CircularArcMarkerConstraints 807
- CircularArcMarkerOpening 808
- CircularArcMarkerRadius 809
- CircularArcMarkerRotation 809
- CircularArcMarkerThickness 810
- CircularMarkerChange 887
- Clear 599
- ClearOutputBlobImageEnabled 441, 718
- ClearSceneSelection 877
- Click 887
- ColorBackground 810
- ColorCalibConfigurationName 68
- ColorCalibDescription 68
- ColorCalibMethod 69
- ColorCalibMethodName 70
- ColorCalibParameter 70
- ColorCalibParameterCount 71
- ColorCalibParameterDescription 71
- ColorCalibParameterEditable 72
- ColorCalibParameterName 72
- ColorCalibration 67
- ColorFilterBestMatchIndex 718
- ColorFilterCount 719
- ColorFilterEnabled 719
- ColorFilterMatchPixelCount 720
- ColorFilterMatchQuality 720
- ColorFilterName 721
- ColorForeground 811
- ColorGrid 812
- CompactMemory 1166
- ComputeArea 616
- ComputeProperties 688
- ComputeShapes 616
- ConfidenceThreshold 941
- ConfigurationCount 73
- ConfigurationDefault 73
- ConfigurationEnabled 74
- ConfigurationIndex 75
- ConfigurationName 75
- ConformityTolerance 333, 1052, 1088
- ConformityToleranceRangeEnabled 1088
- Connectivity 1267
- ConnectivityEnabled 1268
- Constraints 299, 896
- ContextMenuEnabled 812
- ContextMenuMask 813
- ContourDetectionParametersChange 1211
- Contrast 813
- ContrastPolarity 1089
- ContrastThreshold 333, 1053, 1090, 1187, 1268
- ContrastThresholdMode 334, 1053, 1090, 1187, 1269
- ControlPropertyExchange 172, 249, 294, 323, 359, 403, 470, 496, 538, 740, 795, 878, 919, 931, 975, 1005, 1024, 1048, 1083, 1166, 1233, 1261, 1291, 1304, 1336
- CoordinateSystem 255, 299, 335, 375, 441, 475, 501, 721, 752, 896, 941, 1054, 1091, 1238, 1269, 1315
- CoordinateSystemChange 1212
- Copy 617
- CopyArc 586
- CopyFormatFrom 553, 635, 647, 689
- CopyFromSafeArray 617
- CopyImage 586
- CopyLine 587
- CopyModel 1166
- CopyPoint 587
- CopyScene 588
- CopyToSafeArray 618
- CopyView 588
- CornerApproximatePositionX 220
- CornerApproximatePositionY 220
- CornerDetectorHeight 221
- CornerDetectorPositionX 222
- CornerDetectorPositionY 223
- CornerDetectorWidth 223
- Create 566, 619
- CreateConfiguration 172
- CreateEx 619
- CreateMask 1205
- CreatePatternFromInput 1262
- CropModel 1206
- CrossHairVisible 814
- CustomFielding 942
- D**
- Database 76, 245, 256, 300, 335, 376, 442, 475, 501, 722, 753, 897, 923, 943, 982, 1011, 1030, 1055, 1092, 1188, 1213, 1222, 1239, 1270, 1296, 1316
- DatabaseChange 198, 252, 297, 326, 362, 406, 473, 499, 541, 556, 596, 626, 638, 650, 692, 716, 747, 800, 922, 933, 979, 1008, 1026, 1050, 1085, 1184, 1235, 1266, 1294, 1312,

1338
 DatabasePassword 76, 256, 300, 336, 376, 442, 475, 502,
 722, 753, 897, 923, 943, 982, 1030, 1055, 1092, 1222,
 1270, 1316
 DbClick 888
 DecodeLeftToRightEnabled 376
 DefaultConformityTolerance 336, 1055, 1092
 DeleteConfiguration 172
 Description 76, 245, 256, 300, 336, 377, 442, 475, 502, 547,
 558, 570, 597, 604, 629, 641, 653, 696, 722, 753, 814, 897,
 923, 944, 982, 1011, 1030, 1056, 1092, 1222, 1239, 1270,
 1296, 1316
 DetailLevel 1092, 1188
 DetailParametersChange 1221
 Detect 216, 229
 DetectionCount 206, 224, 233
 DetectionMode 224, 233
 DisplayCalibrationMode 1188, 1213
 DisplayContourSceneEnabled 206
 DisplayGreyScaleImageEnabled 206
 DisplayMessageOnError 245
 DisplayMode 207
 DisplayMouseDown 888
 DisplayMouseMove 888
 DisplayMouseUp 889
 Distance 648
 DoClick 878
 Done 198, 252, 297, 326, 362, 406, 473, 499, 542, 747, 800,
 922, 933, 979, 1008, 1027, 1050, 1085, 1184, 1235, 1266,
 1294, 1312, 1339
 DoneClick 1221
 DotConformity 207
 DotPitch 207
 Drag 878
 DuplicateProcess 1304
 DynamicUpdate 1093

E

Edge1Constraints 256, 502
 Edge1Magnitude 257, 503
 Edge1MagnitudeConstraint 257, 503
 Edge1MagnitudeScore 258, 504
 Edge1Polarity 259, 504
 Edge1PositionConstraint 259, 505
 Edge1PositionScore 260, 506
 Edge1PositionX 261, 507
 Edge1PositionY 261, 507
 Edge1Radius 262
 Edge1Rotation 262, 508
 Edge1Score 263, 509
 Edge1ScoreThreshold 264, 509
 Edge2Constraints 264, 510
 Edge2Magnitude 265, 510
 Edge2MagnitudeConstraint 265, 511
 Edge2MagnitudeScore 266, 511
 Edge2Polarity 267, 512
 Edge2PositionConstraint 267, 513
 Edge2PositionScore 268, 514
 Edge2PositionX 269, 514
 Edge2PositionY 269, 515
 Edge2Radius 270
 Edge2Rotation 270, 515
 Edge2Score 271, 516
 Edge2ScoreThreshold 272, 517
 EdgeCount 301, 898
 EdgeFilterHalfWidth 272, 517
 EdgeCount 653
 EdgellImageX 654
 EdgellImageXArray 655
 EdgellImageY 656
 EdgellImageYArray 658
 EdgelWorldX 659
 EdgelWorldXArray 660
 EdgelWorldY 662
 EdgelWorldYArray 663
 EdgeMagnitude 301, 898
 EdgeMagnitudeScore 301, 898
 EdgeMagnitudeThreshold 273, 518
 EdgePathArea 664
 EdgePathCentroidX 665
 EdgePathCentroidY 666
 EdgePathClosed 666
 EdgePathCount 667
 EdgePathFirstEdgelIndex 668
 EdgePathLastEdgelIndex 669
 EdgePathPerimeter 669
 EdgePositionScore 302, 899
 EdgePositionX 302, 899
 EdgePositionY 303, 900
 EdgeRadius 303
 EdgeRotation 304, 900
 EdgeScore 305, 901
 EdgeSensitivity 208
 ElapsedTime 76, 273, 305, 336, 378, 443, 476, 518, 722, 753,
 901, 923, 944, 982, 1012, 1030, 1056, 1093, 1222, 1239,
 1271, 1296, 1317
 EmulationDatabaseFileName 76
 EmulationDeleteImage 172
 EmulationDisableAll 173
 EmulationEnableAll 173
 EmulationExport 174
 EmulationImageCount 77
 EmulationImageEnabled 78
 EmulationImageName 78
 EmulationImport 175
 EmulationNextImageIndex 79
 EmulationNextImageName 80

EndAngle 547
 EndDelta 629
 EndDialog 229, 240, 242, 1206, 1218
 EndPoint 548, 630
 EntityColor 815
 EntityDatabase 816
 EntityDatabasePassword 816
 EntityKValues 816
 EntityName 816
 EntityPenWidth 817
 EntitySelectionChange 889
 EntitySelectionEnabled 817
 EntitySize 818
 EntityViewName 818
 Error 198, 253, 297, 326, 362, 406, 473, 499, 542, 556, 569,
 596, 626, 639, 650, 693, 716, 747, 800, 889, 922, 933, 979,
 1008, 1027, 1050, 1085, 1235, 1266, 1295, 1313, 1339
 ErrorCode 81
 ErrorDescription 81
 ErrorObject 81, 246, 273, 305, 337, 378, 443, 476, 518, 548,
 558, 571, 605, 630, 641, 670, 696, 723, 753, 819, 902, 924,
 944, 983, 1012, 1031, 1056, 1094, 1223, 1240, 1271, 1297,
 1317
 Events 1026
 Execute 175, 294, 323, 359, 403, 470, 496, 538, 740, 741, 742,
 795, 919, 931, 975, 1005, 1024, 1048, 1083, 1167, 1233,
 1262, 1292, 1305, 1336
 Explorer 246
 ExplorerSelectionChange 253
 ExplorerTabVisible 246
 ExportConfiguration 1305
 ExtrinsicBoxResultsEnabled 443
 ExtrinsicInertiaResultsEnabled 444

F

FeatureSelection 1189
 FeatureSelectionChange 1212
 FeatureSelectionSliderChange 1212
 Filename 571
 FilterHalfWidth 306, 902
 FilteringClippingMode 1012
 FilteringCustomKernelAnchorX 1013
 FilteringCustomKernelAnchorY 1013
 FilteringCustomKernelHeight 1014
 FilteringCustomKernelValue 1014
 FilteringCustomKernelWidth 1014
 FilteringKernelSize 1015
 FilteringScale 1015
 FindModelEdgePathFromEdgel 689
 FineTuneMethodEnabled 754, 945
 FirstImageName 571
 FirstModelName 1094
 FirstProcessName 1297
 FirstSceneName 572

FirstViewName 572
 FitMode 337
 FocusChange 890
 FocusX 819
 FocusY 820
 Font 820
 FontType 946

G

GetColorFilterHSLTolerances 740, 741, 742
 GetColorFilterHSLValues 741, 742
 GetColorFilterRGBValues 742
 GetDebugParameter 176, 250, 294, 324, 359, 403, 471, 496,
 539, 742, 796, 919, 931, 976, 1006, 1025, 1048, 1167,
 1233, 1262, 1292, 1306, 1337
 GetDispatchPointer 176, 217, 230, 240, 243, 250, 294, 324,
 359, 404, 471, 497, 539, 554, 566, 589, 619, 636, 648, 689,
 709, 742, 796, 878, 919, 931, 976, 1006, 1025, 1048, 1083,
 1167, 1207, 1219, 1233, 1262, 1292, 1306, 1337
 GetDisplayCoordinates 878
 GetFreeBackgroundSelection 879
 GetInstanceIntrinsicBoundingBox 1168
 GetProcessingFormatEnable 324
 GetProcessingFormatEnabled 294, 360, 471, 539, 919,
 1083, 1292
 GlobalThresholdEnabled 755
 GrabberDCAMModeSupport 92
 GrabberDCAMPParameter 81
 GrabberDCAMPParameterSupport 86
 GrabberDepict4x1GetBusyCount 176
 GrabberDepict4x1InputPort 93
 GrabberDepict4x1OutputPort 94
 GrabberDepict4x1Parameter 94
 GrabberDepict4x1ResetBusyCount 177
 GrabberDirectShowDialog 178
 GrabberDirectShowParameter 97
 GrabberDirectShowParameterRange 101
 GrabberDt3152OutputPort 103
 GrabberDt3152Parameter 103
 GrabberDt3155OutputPort 105
 GrabberDt3155Parameter 105
 GrabberDt3157InputPort 107
 GrabberDt3157OutputPort 107
 GrabberDt3157Parameter 108
 GrabberItIlcAsyncInputPort 114
 GrabberItIlcAsyncOutputPort 114
 GrabberItIlcAsyncParameter 115
 GrabberItIlcAsyncPortParameter 116
 GrabberItIlcPcVisionInputPort 122
 GrabberItIlcPcVisionOutputPort 123
 GrabberItIlcPcVisionPortParameter 124
 GrabberItIpcDigInputPort 118
 GrabberItIpcDigOutputPort 118
 GrabberItIpcDigParameter 119

GrabberItiPcDigPortParameter 121
 GrabberItiPcVisionParameter 123
 GrabberMake 126
 GrabberMatroxMeteor2dioUserBit 127
 GrabberMatroxMeteor2dParameter 128
 GrabberMatroxMeteor2dioUserBit 130
 GrabberMatroxMeteor2mcloUserBit 131
 GrabberMatroxMeteor2mcParameter 134
 GrabberMatroxMeteor2Parameter 132
 GrabberMatroxOriondioUserBit 137
 GrabberMatroxOrionParameter 138
 GrabberModel 140
 GrabberPcx200FireStrobe 179
 GrabberPcx200Parameter 150
 GrabberPx610aFireStrobe 178
 GrabberPx610aParameter 145
 GrabberPxc200InputPort 149
 GrabberPxc200OutputPort 149
 GreylevelRange 983
 GreyLevelResultsEnabled 444
 GridVisible 820

H

Handle 573
 Height 605, 670
 HeightImage 234
 HeightWorld 234
 HelpContext 597
 HelpFile 597
 Hide 217, 230, 240, 243, 1207, 1219
 HideCalibrationDialog 180
 HideColorCalibrationDialog 180
 HideInterface 250, 1306
 HideLearnInterface 796, 976
 HideModelEditor 1169
 HideModelOptimizer 1169
 HideProperties 181, 250, 295, 324, 360, 404, 471, 497, 539,
 742, 796, 879, 920, 931, 976, 1006, 1025, 1048, 1084,
 1169, 1233, 1262, 1292, 1306, 1337
 HideRealTimeDisplay 181
 HideResultsViewer 1170
 Histogram 983
 HistogramPixelCount 984
 HistogramThreshold 1016
 HoleFillingEnabled 445
 HorizontalCaliperHeight 234
 HorizontalCaliperPositionX 235
 HorizontalCaliperPositionY 235
 HorizontalCaliperWidth 235
 HorizontalProbeSpacing 378

I

IconVisible 1190, 1213
 Image 573, 697

ImageBottomLeftX 1317
 ImageBottomLeftY 1318
 ImageBottomRightX 1319
 ImageBottomRightY 1319
 ImageCoordinateSystemType 163, 605, 671
 ImageCount 697
 ImageDatabase 821
 ImageDatabasePassword 821
 ImageExists 697
 ImageHeight 559, 723, 984, 1320
 ImageIndex 698
 ImageName 698, 821
 ImageOriginY 1190
 ImagePalette 822
 ImagePaletteColor 823
 ImagePaletteCycle 824
 ImagePaletteMaskMode 825
 ImagePaletteTransparencyIndex 825
 ImagePaletteType 826
 ImagePixelCount 724, 985
 ImageSize 827
 ImageSubsampling 724, 985
 ImageTopLeftX 1320
 ImageTopLeftY 1320
 ImageTopRightX 1321
 ImageTopRightY 1322
 ImageToWorld 182, 567, 619, 690
 ImageViewName 828
 ImageWidth 559, 724, 986, 1322
 ImageX 642
 ImageY 642
 ImportColorFilter 743
 ImportConfiguration 1307
 ImportModel 1170
 ImportPatternFromDatabase 1263
 Information 548, 559, 573, 606, 631, 643, 671, 699
 InitialDirectory 1095
 InputDevice 163
 InputDeviceAvailable 164
 InputDeviceCount 164
 InputDeviceDescription 165
 InputDeviceName 165
 InputDeviceType 165
 InputEntity 477, 924, 1223
 InputEntityCount 477, 924, 928
 InputEntityDatabase 477, 925, 928, 1224
 InputEntityType 478, 925, 929, 1224
 InputEntityView 478, 925, 929
 InputGreyScaleImage 274, 306, 338, 379, 445, 479, 519, 725,
 756, 903, 926, 927, 929, 946, 947, 986, 1031, 1057, 1095,
 1223, 1225, 1229, 1240, 1241, 1271, 1322
 InputGreyScaleImageView 1095
 InputGreyScaleView 274, 307, 338, 339, 380, 446, 480, 519,

520, 725, 726, 756, 757, 903, 947, 987, 1032, 1057, 1058,
 1224, 1241, 1272, 1323
 InputImage 1016
 InputInstanceScene 274, 307, 339, 380, 446, 520, 726, 757,
 903, 948, 987, 1032, 1058, 1096, 1242, 1272, 1323
 InputInstanceView 275, 307, 339, 381, 446, 520, 727, 757,
 904, 948, 988, 1033, 1058, 1096, 1242, 1273, 1323
 InputModelScene 275, 308, 340, 381, 447, 521, 727, 758, 904,
 948, 988, 1033, 1059, 1242, 1273, 1324
 InputModelView 275, 308, 340, 381, 447, 521, 727, 758, 904,
 949, 988, 1033, 1059, 1243, 1273, 1324
 InputView 1016
 InquireConfiguration 1308
 InstanceClearQuality 1097
 InstanceCount 672, 1098
 InstanceFirstEdgePathIndex 674
 InstanceFitQuality 1098
 InstanceLastEdgePathIndex 676
 InstanceMatchQuality 1098
 InstanceModel 1099
 InstanceModelName 1099
 InstanceOrdering 1100
 InstanceOrderingReferenceX 1101
 InstanceOrderingReferenceY 1102
 InstanceReferencePointCount 1102
 InstanceReferencePointPositionX 1102
 InstanceReferencePointPositionY 1103
 InstanceRotation 1104
 InstancesAnalyzed 1214
 InstanceScaleFactor 1104
 InstanceSelectionEnabled 276, 308, 340, 382, 447, 521, 728,
 758, 905, 949, 989, 1034, 1059, 1105, 1243, 1274, 1324
 InstanceSelectionIndex 276, 309, 341, 382, 448, 522, 728,
 759, 905, 950, 989, 1034, 1060, 1243, 1274, 1325
 InstanceSymmetry 1106
 InstanceTime 1106
 InstanceTranslationX 1107
 InstanceTranslationY 1107
 InstanceVisible 1108
 InteractionMode 828
 InteractionModeChange 890
 InteractiveConfigurationEnabled 1297
 InteractiveSelectionChange 890
 InteractiveSelectionEnabled 829
 InterfaceVisible 246, 1298
 InterpolatePositionMode 1274
 InterpolatePositionModeEnabled 1275
 Intersected 481
 IntrinsicBoxResultsEnabled 448
 IntrinsicInertiaResultsEnabled 449
 IsMarkerDefined 829
 ItiInportInterrupt 198, 253

K

KernelSize 1035
 KeyDown 890
 KeyPress 891
 KeyUp 891

L

LabelMarkerText 829
 LabelMarkerTextAlign 830
 LastImageName 574
 LastModelName 1108
 LastProcessName 1298
 LastSceneName 574
 LastViewName 575
 Learn 404, 797, 977, 1171
 LearnElapsedTime 950
 LearnFontType 950
 LearnOcrMethod 951
 LearnPrimaryCharacter 951
 LearnPrimaryCharacterBoxBottom 952
 LearnPrimaryCharacterBoxLeft 952
 LearnPrimaryCharacterBoxRight 953
 LearnPrimaryCharacterBoxTop 953
 LearnStatusCode 953
 LearnStatusCodeString 954
 LearnTextOrientation 956
 LearnTextPolarity 956
 LearnTime 1109
 LearnTrainingBoxBottom 957
 LearnTrainingBoxLeft 957
 LearnTrainingBoxRight 958
 LearnTrainingBoxTop 958
 LearnTrainingString 958
 Line 575, 699
 LineAngle 1060
 LineAverageContrast 1061
 LineCount 699
 LineEndPointPositionX 1061
 LineEndPointPositionY 1061
 LineExists 700
 LineFitQuality 1062
 LineFound 1062
 LineIndex 700
 LineMarkerChange 891
 LineMarkerConstraints 830
 LineMarkerX2 831
 LineMarkerY2 832
 LineMatchQuality 1063
 LineName 701
 LineStartPointPositionX 1063
 LineStartPointPositionY 1064
 LineTo 620
 LineVectorPointPositionX 1064

LineVectorPointPositionY 1065
 LinkOutlineWithDetail 1214
 Load 554, 567, 589, 620, 636, 649, 690, 710
 LoadCalibration 182
 LoadColorCalibration 183
 LoadConfiguration 183, 250, 295, 324, 360, 405, 472, 497,
 539, 589, 743, 798, 879, 920, 932, 977, 1006, 1025, 1049,
 1084, 1171, 1234, 1263, 1293, 1308, 1337
 LoadEmulationDatabase 183
 LoadGrabberParameters 184
 LoadModelDatabase 1172
 LoadPatternDatabase 1263
 LocalThresholdEnabled 759
 LockOnCornersEnabled 225
 LogicalConstant 1017

M

MagnitudeConstraint 309, 906
 MagnitudeThreshold 310, 906
 MarkerAnchorColor 832
 MarkerAnchorStyle 833
 MarkerCalibrated 834
 MarkerColor 834
 MarkerDefined 835
 MarkerDisplayName 836
 MarkerGroup 836
 MarkerLineStyle 836
 MarkerPenWidth 837
 MarkerX 838
 MarkerY 838
 MatchCount 1244
 MatchPositionX 1244
 MatchPositionY 1245
 MatchRotation 1245
 MatchStrength 1246
 MatchThreshold 1246
 MatrixAngle 760
 MatrixBorderMatchPercentage 761
 MatrixColumns 761
 MatrixColumnsEnabled 762
 MatrixContrast 762
 MatrixCornerX 763
 MatrixCornerY 764
 MatrixDamagePercentage 765
 MatrixDecodeTime 766
 MatrixECCLevel 766
 MatrixECCLevelEnabled 767
 MatrixFoundCount 767
 MatrixHeight 768
 MatrixHeightEnabled 769
 MatrixImageStyle 769
 MatrixImageStyleEnabled 770
 MatrixOrientation 770
 MatrixOrientationEnabled 771
 MatrixPolarity 772
 MatrixPolarityEnabled 772
 MatrixRows 773
 MatrixRowsEnabled 773
 MatrixStatusCode 774
 MatrixStatusCodeString 775
 MatrixString 776
 MatrixThreshold 777
 MatrixToDecode 777
 MatrixTotalErrorBits 778
 MatrixWidth 779
 MatrixWidthEnabled 779
 MaximalBoundaryHeight 839
 MaximalBoundaryWidth 839
 MaximumAngleDeviation 341, 1065
 MaximumBarWidth 382
 MaximumBlobArea 449
 MaximumCodeHeight 383
 MaximumCodeLength 384
 MaximumConformityTolerance 342, 1065, 1110
 MaximumDotRadius 208
 MaximumGreylevelValue 990
 MaximumImageError 209
 MaximumInstanceCount 1110, 1247
 MaximumInstanceCountEnabled 1110, 1247
 MaximumNumberOfBars 384
 MaximumRotation 1111
 MaximumScaleFactor 1111
 MaximumWorldError 209
 Mean 990
 MeasurementPointsCount 1035
 Median 990
 MessageCount 342, 1066, 1112, 1190, 1275
 MessageDescription 342, 1066, 1112, 1190, 1276
 MessageNumber 343, 1067, 1113, 1191, 1276
 Methods 1024
 MinimumArcPercentage 343
 MinimumBarWidth 385
 MinimumBlobArea 450
 MinimumClearPercentage 1113
 MinimumClearPercentageEnabled 1114
 MinimumCodeHeight 385
 MinimumCodeLength 386
 MinimumConformityTolerance 344, 1067, 1114
 MinimumDotRadius 209
 MinimumEdgeStrength 386, 780
 MinimumGreylevelValue 991
 MinimumLinePercentage 1068
 MinimumModelPercentage 1114
 MinimumNumberOfBars 387
 MinimumQuietZone 387

- MinimumRecurrenceDetail 1215
- MinimumRecurrenceOutline 1215
- MinimumRequiredFeatures 1115
- MinimumRotation 1115
- MinimumScaleFactor 1116
- Mode 481, 926, 991, 1226, 1298
- ModeChange 1313
- ModelAutomaticLevels 1116
- ModelBasedMaximumRotation 1117
- ModelBasedMaximumScaleFactor 1117
- ModelBasedMinimumRotation 1118
- ModelBasedMinimumScaleFactor 1118
- ModelBoundingAreaBottom 1120
- ModelBoundingAreaLeft 1121
- ModelBoundingAreaRight 1121
- ModelBoundingAreaTop 1122
- ModelContrastThreshold 1123
- ModelContrastThresholdMode 1123
- ModelCount 1124
- ModelDatabase 1124
- ModelDatabaseFileName 1109, 1125
- ModelDatabaseModified 1125
- ModelDetailLevel 1125
- ModelDisambiguationEnabled 1126
- ModelEdgePathArea 678
- ModelEdgePathCentroidX 678
- ModelEdgePathCentroidY 679
- ModelEdgePathCount 680, 1192
- ModelEdgePathFirstEdgeIndex 680
- ModelEdgePathLastEdgeIndex 682
- ModelEdgePathLocation 1192
- ModelEdgePathNextEdgeIndex 682
- ModelEdgePathPerimeter 683
- ModelEdgePathPreviousEdgeIndex 683
- ModelEdgePathRequired 1193
- ModelEditorActive 1126
- ModelEditorEnabled 1127
- ModelEnabled 1127
- ModelFeatureSelection 1128
- ModelImage 1193
- ModelMask 1194
- ModelName 1128
- ModelOptimizerActive 1129
- ModelOptimizerEnabled 1129
- ModelOptimizerReset 1173
- ModelOriginPositionX 1130
- ModelOriginPositionY 1130
- ModelOriginRotation 1131
- ModelOutlineLevel 1132
- ModelReferencePointCount 1132
- ModelReferencePointPositionX 1133
- ModelReferencePointPositionY 1133
- ModelSceneDetail 1195
- ModelSceneOutline 1195
- ModelShadingAreaBottom 1134
- ModelShadingAreaLeft 1135
- ModelShadingAreaRight 1135
- ModelShadingAreaTop 1136
- ModelTrackingInertia 1136
- ModelTransformEnabled 277, 310, 344, 388, 450, 522, 728, 780, 906, 959, 991, 1036, 1068, 1247, 1277, 1325
- ModePixelCount 992
- Modify 589, 621, 691, 710
- ModifyCount 549, 575, 606, 631, 643, 684, 701
- MorphingDetail 1216
- MorphingOutline 1216
- MorphologicalNeighborhoodSize 1017
- Move 879
- MoveProcess 1309
- MultiSelectionEnabled 840
- N**
- Name 166, 247, 277, 311, 345, 388, 451, 484, 523, 549, 576, 607, 631, 643, 684, 702, 729, 781, 907, 927, 959, 992, 1017, 1036, 1069, 1137, 1195, 1216, 1227, 1248, 1277, 1299, 1326
- NearestPoint 554, 636
- NewEmulationDatabase 184
- NewModelDatabase 1165, 1173
- NewPatternDatabase 1264
- NextImageName 576
- NextModelName 1138
- NextProcessName 1299
- NextSceneName 577
- NextViewName 577
- NominalRotation 1138
- NominalRotationEnabled 1138
- NominalScaleFactor 1139
- NominalScaleFactorEnabled 1139
- Number 598
- O**
- OcrMethod 960
- OKClick 1212
- OperandImage 1018
- OperandView 1018
- Operation 1018
- OriginPositionX 210, 1196
- OriginPositionXWorld 210
- OriginPositionY 211, 1196
- OriginPositionYWorld 211
- OriginRotation 1197
- OriginX 607, 685
- OriginY 607, 685
- OutlineLevel 1140, 1197
- OutlineParametersChange 1221

OutputArc 485, 927, 1228
 OutputBlobImage 451
 OutputBlobImageEnabled 451
 OutputBlobView 452
 OutputDetailScene 1140
 OutputDetailSceneEnabled 1141
 OutputDetailSceneView 1141
 OutputEntity 345, 485, 1069, 1228, 1229, 1278
 OutputEntityCount 485
 OutputEntityDatabase 486
 OutputEntityEnabled 345, 1069, 1278
 OutputEntityExtrapolated 486
 OutputEntityType 487
 OutputEntityView 346, 487, 1070, 1279
 OutputFilterImage 729
 OutputFilterImageEnabled 730
 OutputFilterView 730
 OutputGreyScaleImage 166, 1326
 OutputGreyScaleView 1326
 OutputImage 1020
 OutputInstanceScene 1141
 OutputInstanceSceneEnabled 1142
 OutputInstanceSceneView 1142
 OutputLine 488
 OutputMode 1143
 OutputModelEnabled 1143
 OutputModelView 1144
 OutputOutlineScene 1144
 OutputOutlineSceneEnabled 1145
 OutputOutlineSceneView 1145
 OutputPoint 488
 OutputSymmetricInstances 1146
 OutputView 167, 1020
 OverrideModelImageEnabled 1248
 OverrideType 1021
 OverrideTypeEnabled 1021

P

PairCount 277, 523
 PairIndex 278, 523
 PairName 278, 524
 PairPositionX 279, 524
 PairPositionY 280, 525
 PairRadius 280
 PairRotation 281, 526
 PairScore 282, 527
 PairSize 282, 527
 Parameter 489, 559
 ParameterCount 491, 561, 1229
 ParameterDescription 562
 ParameterName 491, 562
 ParametersBasedOn 1146
 ParameterType 492
 PassThruRunning 168
 Password 577
 PatternDatabaseFileName 1249
 PatternHeight 1249
 PatternPositionX 1250
 PatternPositionY 1250
 PatternRotation 1251
 PatternWidth 1252
 PercentageOfPointsToAnalyze 1147
 PerimeterResultsEnabled 452
 Perpendicular 637
 Pixel 608
 PixelHeight 1327
 PixelWidth 1327
 Point 578, 702
 PointArrayMarkerSize 840
 PointArrayMarkerX 840
 PointArrayMarkerY 841
 PointAverageContrast 1279
 PointCount 702
 PointExists 703
 PointFound 1279
 PointIndex 703
 PointMarkerChange 892
 PointMarkerConstraints 842
 PointName 704
 PointPositionX 1280
 PointPositionY 1280
 Polarity 311, 389, 907
 PolarityEnabled 390
 PolarityMode 346, 1070, 1280
 PositionConstraint 312, 908
 PositioningCoarseness 1252
 PositioningLevel 347, 1071, 1147, 1281
 PositionXImage 225
 PositionXWorld 226
 PositionYImage 227
 PositionYWorld 227
 PreviousImageName 578
 PreviousModelName 1148
 PreviousProcessName 1300
 PreviousSceneName 578
 PreviousViewName 579
 PriorityMask 842
 Process 1300
 ProcessAvailable 1301
 ProcessCount 1301
 ProcessIndex 1301
 ProcessingAlgorithm 391, 781, 960
 ProcessingFormat 283, 312, 347, 453, 528, 908, 1022, 1071, 1148, 1281, 1328
 ProcessingSize 392, 782, 961

ProcessListEditable 1302
 ProcessManager 247
 ProcessManagerModeChange 253
 ProcessManagerTabVisible 247
 ProcessTypeName 1302
 ProjectionMode 284, 313
 PropertiesChange 198, 253, 297, 326, 362, 406, 474, 499,
 542, 747, 800, 892, 922, 933, 980, 1008, 1027, 1051, 1086,
 1184, 1236, 1266, 1295, 1313, 1339

R

Radius 549
 RawImagePointer 608
 RecognitionLevel 1149
 RectangleMarkerChange 892
 RectangleMarkerConstraints 843
 RectangleMarkerHeight 844
 RectangleMarkerRotation 844
 RectangleMarkerSkew 845
 RectangleMarkerWidth 846
 ReferencePointCount 1198
 ReferencePointPositionX 1198
 ReferencePointPositionY 1198
 ReferencePointsChange 1212
 Refresh 251, 879, 1309
 RefreshChangedMarker 880
 RefreshDisplay 880, 1207
 RefreshMarker 881
 RefreshMarkers 881
 RemoveAllMarker 881
 RemoveArc 590, 710
 RemoveColorFilter 743
 RemoveImage 590, 711
 RemoveLine 591, 711
 RemoveMarker 882
 RemoveMarkerGroup 882
 RemoveMask 1209
 RemoveModel 1173
 RemoveModelEdgePath 1208
 RemovePair 295, 540
 RemovePoint 591, 712
 RemoveProcess 1310
 RemoveReferencePoint 1208
 RemoveScene 592, 712
 RemoveView 592
 RenameArc 592, 713
 RenameImage 593, 713
 RenameLine 593, 713
 RenameModel 1174
 RenamePoint 594, 714
 RenameScene 594, 714
 RenameView 595
 Render 622

Reset 555, 595, 638, 649, 691, 715, 1219, 1310
 ResetClick 1221
 ResetHistory 1049
 ResetShapes 622
 RespectContrastPolarity 1149
 ResultsDisplayMode 1022
 ResultSelectionChange 1221
 Revert 1209
 RowsToRead 962
 RulersVisible 846

S

SamplesPerCell 783
 SamplingStep 284, 313, 392, 453, 529, 730, 783, 909, 962,
 993, 1036, 1252, 1329
 SamplingStepCustom 285, 314, 393, 454, 529, 731, 784, 909,
 963, 993, 1037, 1253, 1329
 SamplingStepCustomEnabled 285, 314, 393, 454, 530, 731,
 784, 910, 963, 993, 1037, 1253, 1330
 SamplingStepDefault 285, 315, 394, 455, 530, 732, 785, 910,
 964, 994, 1038, 1254, 1330
 Save 555, 568, 596, 623, 638, 649, 691, 715
 SaveCalibration 185
 SaveColorCalibration 186
 SaveConfiguration 186, 251, 296, 325, 360, 405, 472, 497,
 540, 596, 744, 745, 798, 883, 920, 932, 977, 1006, 1025,
 1049, 1084, 1174, 1234, 1264, 1293, 1311, 1337
 SaveEmulationDatabase 186
 SaveGrabberParameters 187
 SaveModelDatabase 883, 1174
 SavePatternDatabase 1264
 ScaleToFitEnabled 847
 Scene 579, 704
 SceneColor 847
 SceneCount 704
 SceneDatabase 848
 SceneDatabasePassword 848
 SceneExists 705
 SceneIndex 705
 SceneName 706, 848
 ScenePenWidth 849
 SceneSelection 850
 SceneSelectionColor 851
 SceneSelectionIndex 852
 SceneSelectionPenWidth 852
 SceneSelectionType 853
 SceneSize 854
 SceneViewName 855
 ScoreThreshold 315, 911
 ScrollBarsVisible 855
 SearchAreaBottom 1150
 SearchAreaLeft 1150
 SearchAreaRight 1150
 SearchAreaTop 1150

- SearchAreaUseEntireImage 1150
- SearchBasedOnOutlineLevelOnly 1150
- SearchCoarseness 1254
- SearchDirection 394, 785
- SearchMode 348, 1072, 1282
- SearchSpeed 785
- SearchTime 1151
- SegmentationDark 455
- SegmentationDynamicDark 456
- SegmentationDynamicInside 456
- SegmentationDynamicLight 457
- SegmentationDynamicOutside 457
- SegmentationInside 458, 459
- SegmentationLight 459
- SegmentationMode 460
- SegmentationOutside 460
- SelectionGreyScaleImage 168
- SelectionView 168
- SetColor 623
- SetColorFilterHSLTolerances 744
- SetColorFilterHSLValues 745
- SetColorFilterRGBValues 745
- SetDebugParameter 187, 251, 296, 325, 361, 405, 472, 497, 540, 746, 798, 920, 932, 977, 1006, 1025, 1049, 1084, 1175, 1234, 1264, 1293, 1311, 1337
- SetFirst 624
- SetFocus 883
- SetFocusXY 883
- SetForcePositiveSign 624
- SetMarkerPosition 884
- SetPointArrayMarkerPosition 884
- ShadingAreaBottom 1199
- ShadingAreaLeft 1199
- ShadingAreaRight 1200
- ShadingAreaTop 1200
- ShapesCalibrated 609
- Sharpness 1038
- SharpnessHistory 1039
- SharpnessHistoryScale 1039
- SharpnessHistorySize 1040
- SharpnessPeak 1040
- Show 217, 231, 241, 243, 1210, 1219
- ShowBars 1200, 1217
- ShowCalibDistortionLUTDialog 187
- ShowCalibDistortionModelDialog 188
- ShowCalibPerspectiveDialog 189
- ShowCalibXYScaleDialog 189
- ShowColorCalibrationDialog 190
- ShowContourSceneDetail 1151
- ShowContourSceneOutline 1152
- ShowCoordinateSystem 1152
- ShowEdges 348, 1073, 1283
- ShowEntityFound 349, 1073, 1283
- ShowEntityFoundAs 349, 1073
- ShowExtrapolations 492, 1231
- ShowGreyScaleImage 350, 492, 493, 1074, 1152, 1201, 1217, 1283
- ShowImageMode 1201, 1217
- ShowInputEntities 493
- ShowInstanceScene 1153
- ShowInstanceSceneAs 1153
- ShowInterface 251, 1311
- ShowLearnInterface 798, 978
- ShowModal 1210, 1220
- ShowModelEditor 1175
- ShowModelEditorModal 1176
- ShowModelOptimizer 1176
- ShowModelOptimizerModal 1177
- ShowOutputEntities 493
- ShowProperties 191, 252, 296, 325, 361, 405, 472, 497, 540, 746, 799, 885, 920, 932, 978, 1006, 1025, 1049, 1084, 1168, 1178, 1234, 1265, 1293, 1312, 1338
- ShowRealTimeDisplay 192
- ShowReferencePoints 1154
- ShowResultsViewer 1179
- ShowSearchArea 350, 1074, 1154, 1284
- SingleExecute 198, 297, 326, 362, 406, 474, 499, 542, 748, 800, 922, 933, 980, 1008, 1027, 1051, 1086, 1184, 1236, 1266, 1295, 1339
- SizeVariation 786
- Solve 231, 241, 244, 568
- SortBlobsBy 461
- SortResultsEnabled 315, 463, 911
- Source 598
- StandardDeviation 994
- StandardDeviationThreshold 1040
- Start 192, 1180
- StartAngle 550
- StartDelta 632
- StartPassThru 193
- StartPoint 551, 632
- StartShape 624
- StartStopEnabled 395
- State 563
- StatusBarVisible 856
- StatusCode 964
- StatusCodeString 965
- Stop 1181
- StopPassThru 194
- String 966
- StringConfidence 966
- StringLengthVariation 967
- StrokeWidth 940
- SubSampleFrom 625
- SubSamplingFactor 609
- SubsamplingLevel 351, 1075, 1284

T

TailBlack 995
 TailBlackGreylevelValue 995
 TailWhite 996
 TailWhiteGreylevelValue 996
 TargetMarkerChange 893
 TargetMarkerConstraints 856
 TargetMarkerRadius 857
 TargetMarkerRotation 857
 TextOrientation 967
 TextPolarity 968
 ThresholdBlack 997
 ThresholdWhite 997
 Timeout 396, 787, 1155
 TimeoutEnabled 1155
 TimeStampHigh 169
 TimeStampLow 169
 Title 858
 TitleVisible 858
 ToolBarVisible 859
 ToolGuidelineOffset 351, 1075, 1284
 ToolHeight 396, 463, 530, 732, 787, 911, 969, 998, 1041, 1075, 1156, 1255, 1285, 1330
 ToolOpening 286, 316, 351
 ToolPositionX 286, 316, 352, 396, 464, 531, 733, 787, 912, 969, 998, 1041, 1076, 1156, 1255, 1285, 1331
 ToolPositionY 287, 317, 352, 397, 464, 531, 733, 788, 912, 970, 999, 1042, 1076, 1157, 1256, 1286, 1331
 ToolRadius 287, 317, 353
 ToolRotation 288, 318, 353, 397, 465, 532, 734, 788, 913, 970, 999, 1042, 1077, 1157, 1256, 1286, 1331
 ToolSkew 533, 914
 ToolThickness 289, 319, 354
 ToolUseEntireImage 1158
 ToolWidth 398, 465, 533, 734, 789, 914, 971, 1000, 1043, 1078, 1158, 1257, 1287, 1332
 TopologicalResultsEnabled 466
 TrackingInertia 1159, 1202
 TransformFlags 1023
 TransformRotation 289, 319, 355, 399, 466, 533, 735, 790, 914, 971, 1001, 1044, 1078, 1159, 1257, 1287, 1333
 TransformScale 290, 320, 355, 399, 467, 534, 735, 790, 915, 972, 1001, 1044, 1079, 1160, 1258, 1288, 1333
 TransformTranslationX 290, 320, 356, 400, 467, 535, 736, 791, 916, 972, 1002, 1045, 1079, 1160, 1258, 1288, 1333
 TransformTranslationY 291, 321, 356, 400, 468, 535, 736, 791, 916, 973, 1002, 1045, 1080, 1161, 1259, 1289, 1334
 Type 563, 609, 686
 TypeName 170, 291, 321, 357, 401, 468, 494, 536, 551, 564, 580, 598, 610, 633, 644, 686, 706, 737, 792, 917, 930, 973, 1003, 1023, 1046, 1080, 1161, 1232, 1259, 1289, 1334

U

UnitsLength 212, 228, 236, 291, 321, 357, 401, 469, 494, 536, 551, 564, 633, 644, 687, 737, 792, 859, 917, 930, 974, 1003, 1046, 1080, 1162, 1229, 1260, 1290, 1335
 UnitsOrientation 292, 322, 357, 401, 469, 495, 537, 552, 611, 634, 645, 738, 792, 859, 918, 974, 1004, 1047, 1081, 1162, 1260, 1290, 1335
 Unlearn 1182
 Update 1220
 UpdateDisplayOnDone 248
 UpdateExplorerOnDatabaseChange 248
 UseCustomShadingArea 1202
 UseDefaultConformityTolerance 358, 1081, 1163

UseExternalModelDatabase 1163
 UseMinimumClearPercentage 1164

V

ValidateCalibration 195
 ValidateColorCalibration 196
 ValidationDotCount 212
 ValidationDotPositionXImage 213
 ValidationDotPositionXWorld 213
 ValidationDotPositionYImage 214
 ValidationDotPositionYWorld 215
 Variance 1004
 VectorPoint 634
 Version 170, 249, 292, 322, 358, 402, 470, 495, 537, 552, 565, 580, 599, 611, 635, 645, 687, 706, 738, 793, 860, 918, 930, 975, 1004, 1024, 1047, 1082, 1164, 1232, 1261, 1291, 1303, 1336
 VerticalCaliperHeight 236
 VerticalCaliperPositionX 237
 VerticalCaliperPositionY 237
 VerticalCaliperWidth 237
 VerticalProbeSpacing 402
 View 580
 ViewCount 581
 ViewExists 581
 ViewIndex 582
 ViewName 582

W

Wait 196, 1182
 WarpMethod 793
 WheelOperationMode 860
 Width 612, 687
 WidthHeightRatio 794
 WidthHeightRatioEnabled 794
 WidthImage 238
 WidthWorld 238
 WorldCoordinateSystemType 215, 229, 239, 565
 WorldToImage 197, 569, 625, 692
 WorldX 645
 WorldY 646

Z

Zoom 861
 ZoomChange 894
 ZoomToPage 861



www.LMItechnologies.com

©2009 LMI Technologies Inc.
Conditions and specifications subject to change without notice.
HexSight is a registered trademark of LMI Technologies Inc.
All other names are property of their respective owners.

