

# Porting Guide

*From ImageViz to ImageDev*

This document provides important information and instructions for how to replace the ImageViz extension for Open Inventor by the ImageDev library.

ImageDev 2023.1 and later
<b>Revision Date: March 10, 2023</b>

## Table of Contents

Overview.....	3
Installation .....	4
Unavailable features .....	4
Algorithm .....	4
Naming rules.....	4
Programming modes.....	5
Example.....	5
Image Model.....	6
Image Model to Volume Data .....	6
Image Properties.....	6
Quantification results .....	7
Global measurements .....	7
Individual Measurements .....	8
Custom Measurements .....	9
Geometric objects.....	10
Transformation matrix .....	10
Line Set.....	10
Behavior changes.....	11
SolImageRegistrationTransform to AffineRegistration .....	11
SoLabelingProcessing to Labeling2d and Labeling3d .....	11
Correspondence table.....	12

## Overview

ImageDev is a modern image processing library that is a substitute for the legacy ImageViz extension of Open Inventor. Unlike ImageViz, the ImageDev library is independent from Open Inventor and delivered as separate packages.

The algorithmic content of ImageDev is mostly the same as ImageViz: both libraries rely on the same computing kernel, but ImageDev is based on a new major version of this kernel. The main differences are:

- The names of algorithms and their parameters
- The type of objects used for managing specific parameters such as images, geometries, arrays or analysis results
- The supported languages

Feature	ImageViz	ImageDev
Supported language	C++, C#, Java	C++, C#, Python
Supported OS	Windows 10, RHEL 7 & 8, Ubuntu 18.04 & 20.04	Windows 10, Ubuntu 18.04 & 20.04
C++ Compiler	VC 15, GCC 7 & 9	VC 15 & 16, GCC 7 & 9
Supported .NET Implementation	.NET Framework 4.6.1 & 4.7.1	.NET Framework ≥ 4.6.1 .NET Core ≥ 2.0 .NET 5 & 6
Dependency with Open Inventor	Requires OIV Core	None
Data Model	Image Adapters	IOLink ImageView
Programming Mode	Class only (OIV engines)	Class and functions
Reentrancy	No	Yes (for recent algorithms)
In place usage	No	Yes (for most of algorithms)
Disposable images (.NET)	No	Yes
Segmentation by machine learning	No model management. Training necessarily done with prediction.	Model can be trained and reused separately.
Out of core processing	Only for input (read mode)	Yes (read & write)
Superpixel segmentation	No	Yes

## Installation

The `OpenInventorImageDevExamplesC++` and `OpenInventorImageDevExamplesNet` packages contain some examples and utilities that can be used to visualize ImageDev results with Open Inventor.

This section provides instructions for how to configure and build the projects containing these examples.

More details about the installation of the ImageDev library are available in the "Getting Started" pages of the ImageDev reference manual.

To use ImageDev with Open Inventor you need to:

1. Download and unzip the `OpenInventorImageDevExamples` package from the [ImageDev download pages](#).
2. Follow the steps of the `README.md` file to build and run examples of interoperability between ImageDev and Open Inventor.

## Unavailable features

- ImageDev is not available in Java.
- The mechanism for connecting engines is not available in ImageDev.
- The following ImageViz features are currently unavailable in ImageDev:
  - The [SoObjectToSegmentApproximation2d](#) engine
  - The [SoQuadrilateralMeshSlicerApproximation3d](#) engine
  - The [SoObjectToRegionProcessing2d](#) and [SoRegionToObjectProcessing2d](#) engines
  - The [SoExtremalImpositionProcessing](#)
- The [DentalPanoramicExtractor](#) class is not directly available in ImageDev. From ImageDev 2023.1, the [DentalWallSegmentation](#), [DentalMeshExtraction](#) and [DentalUnfolding3d](#) algorithms perform the main steps of this feature.

## Algorithm

### Naming rules

Most of image processing engines used in ImageViz can be easily found in ImageDev by applying the following rules:

- Remove the *'So'* prefix from its name.
- Remove the *'Processing'*, *'Generation'*, *'Approximation'* or *'Quantification'* suffix from its name.

Parameter names are usually unchanged except for the input and output data, for which the *'in'* and *'out'* prefixes are replaced by *'input'* and *'output'*.

Be aware that:

- Some algorithms or parameters have a different name to improve the API consistency.

- Some algorithms may have been split into several algorithms in ImageDev, generally to separate the 2D and 3D cases. For example, [SoLabelingProcessing](#) is replaced by [Labeling2d](#) and [Labeling3d](#).

A complete table is available at the end of the document. It shows the correspondence between ImageViz engines and ImageDev algorithms.

## Programming modes

ImageDev provides two modes for invoking an image processing algorithm: a class and a function programming mode.

When using an algorithm that has few parameters, the function is more convenient. When using an algorithm that has many parameters and using the default value for most of them, the class mode produces more readable code.

## Example

This example shows the differences between ImageViz and ImageDev for executing a denoising algorithm.

### ImageViz code

You can consult the 01.2.DenoisingImage3D example of the Open Inventor ImageVizExamples solution for the complete code of the following snippet.

```
// Apply a bilateral denoising filter in ImageViz
SoRef<SoBilateralFilterProcessing> bilateralFilter = new SoBilateralFilterProcessing();
bilateralFilter->inImage = inputImage.ptr();
bilateralFilter->computeMode = SoBilateralFilterProcessing::MODE_3D;
bilateralFilter->kernelSize.setValue(SbVec3d(5, 5, 5));
SoImageDataAdapter* outputImage = bilateralFilter->outImage.getValue();
```

### ImageDev code (class mode)

You can consult the DenoisingImage3D example of the OpenInventorImageDevExample solution for the complete code of the following snippet.

```
// Apply a bilateral denoising filter in ImageDev with class syntax
BilateralFilter3d bilateralFilter;
bilateralFilter.inputImage = inputImage;
bilateralFilter.setKernelSizeX(5);
bilateralFilter.setKernelSizeY(5);
bilateralFilter.setKernelSizeZ(5);
bilateralFilter.execute();
auto outputImage = bilateralFilter.outputImage();
```

### ImageDev code (function mode)

```
// Apply a bilateral denoising filter in ImageDev with function syntax
auto outputImage = BilateralFilter3d(inputImage, 5, 5, 5, 20.0f, BilateralFilter3d::BILATERAL);
```

## Image Model

The image model used by ImageDev is the **ImageView** object of the IOLink library. IOLink is an internal library developed and maintained by ThermoFisher which is embedded within ImageDev.

## Image Model to Volume Data

These examples compare how to convert an image processing image model into an **SoVolumeData** used by Open Inventor for visualization.

### ImageViz code

You can consult the 01.2.DenoisingImage3D example of the Open Inventor ImageVizExamples solution for the complete code of the following snippet.

```
// Create a Volume Data from an ImageViz output image which is an SoImageDataAdapter object
SoRef<SoVRImageDataReader> imageVizReader = new SoVRImageDataReader;
imageVizReader->imageData = outputImage;
SoVolumeData* volumeFilteredImage = new SoVolumeData;
volumeFilteredImage->setReader(*imageVizReader);
```

### ImageDev code

Converting a **MemoryView** object into an Open Inventor **SoVolumeData** object requires integration of the **ImageViewReader** class, available in the IOLinkInterop project of the OpenInventorImageDevExamples packages.

You can consult the DenoisingImage3D example of the OpenInventorImageDevExamples solution for the complete code of the following snippet.

```
// Create a Volume Data from an ImageDev output image which is an ImageView object
ImageViewReader* reader = new ImageViewReader(outputImage);
SoVolumeData* pVolData = new SoVolumeData();
pVolData->setReader(*reader);
```

## Image Properties

In ImageDev, the **ImageView** object replaces the **SoImageDataAdapter**. Some ImageView properties, like calibration, are not straightforward to access. ImageDev provides some helpers to simplify their access.

ImageViz	ImageDev
SoImageDataAdapter:: getInterpretation	imagedev::getImageInterpretation
SoImageDataAdapter:: setInterpretation	imagedev::setImageInterpretation
SoImageDataAdapter:: getImageDataType	ImageView::datatype
SoImageDataAdapter:: getSize	ImageView::shape

ImageViz	ImageDev
SoImageDataAdapter:: getVoxelSize	ImageView::properties::calibration::spacing imageDev:: getCalibrationSpacing
SoImageDataAdapter:: setVoxelSize	ImageView::properties::calibration::setSpacing imageDev:: setCalibrationSpacing
SoImageDataAdapter:: getExtent	Not directly available, straightforward to deduce from shape and spacing.
SoImageDataAdapter:: getOrigin	ImageView::properties::calibration::origin imageDev:: getCalibrationOrigin
SoImageDataAdapter:: setOrigin	ImageView::properties::calibration::setOrigin imageDev:: setCalibrationOrigin
SoImageDataAdapter:: setDirection	ImageView::properties::calibration::setDirection

Any ImageViz 2D algorithm can be applied on a 3D volume, considering it as a stack of 2D images. ImageDev algorithms dedicated to 2D processing do not accept 3D volumes as inputs. To get this behavior back, the `dimensionalInterpretation` property of the input `ImageView` must be changed to `IMAGE_SEQUENCE`.

### ImageViz code

```
SoRef<SoFillHolesProcessing2d> holeFilling = new SoFillHolesProcessing2d();
holeFilling->inObjectImage = inputAdapter.ptr();
SoImageDataAdapter* outputAdapter = holeFilling->outObjectImage.getValue();
```

### ImageDev code

```
imageDev::setDimensionalInterpretation( inputImage, iolink::ImageTypeId::IMAGE_SEQUENCE );
auto outputImage = imageDev::fillHoles2d( inputImage, imageDev::FillHoles2d::CONNECTIVITY_4 );
imageDev::setDimensionalInterpretation( outputImage, iolink::ImageTypeId::VOLUME );
```

## Quantification results

### Global measurements

Global analysis algorithms extract a series of features for the whole image. In the following section, *'MeasureName'* replaces the name of a global measurement available in the libraries, such as `IntensityExtrema`, `ObjectCount`, `Area` or any other.

### ImageViz code

In ImageViz, global measurements computed by any `SoMeasureNameQuantification` engine are returned in an `SoImageAnalysisResult` field containing an `SbMeasureNameDetail` object, allowing access to all analysis results through `getFieldName` methods.

```
// Extract minimum and maximum intensities of an ImageDataAdapter
SoRef<SoIntensityExtremaQuantification> extrema = new SoIntensityExtremaQuantification();
extrema->inImage = inputImage.ptr();
float minIntensity = extrema->outResult.getDetail(0).getMinimum();
float maxIntensity = extrema->outResult.getDetail(0).getMaximum();
```

### ImageDev code

In ImageDev, each global measurement computation is implemented in a **MeasureName** class. The computation outputs an instance of the **MeasureNameMsr** class with dedicated **fieldName** accessors for all fields. These accessors have no 'get' prefix.

```
// Extract minimum and maximum intensities of an ImageView (function mode)
auto extrema = intensityExtrema(inputImage, 0);
float minIntensity = extrema->minimum(0);
float maxIntensity = extrema->maximum(0);
```

## Individual Measurements

Individual analysis algorithms extract a series of features for each object of a label image.

### ImageViz code

In ImageViz, the selected individual measurements are defined by setting the **measureList** field of analysis engines and results are returned in an **SoLabelAnalysisResult** object, which allows access to analysis results through **getMeasureAsType** methods.

You can consult the 03.1.LabelAnalysis example of the Open Inventor ImageVizExamples solution for the complete code of the following snippet.

```
// Initialize a label analysis engine to compute diameter of particles
SoRef<SoLabelAnalysisQuantification> labelAnalysis = new SoLabelAnalysisQuantification;
labelAnalysis->measureList.set1Value(0,
    new SoDataMeasurePredefined(SoDataMeasurePredefined::EQUIVALENT_DIAMETER));
// Plug its inputs on a label image previously computed
labelAnalysis->inLabelImage = labelImage;
labelAnalysis->inIntensityImage = labelImage;
// Get the first result of this analysis
SoLabelAnalysisResult* analysis = labelAnalysis->outAnalysis.getValue();
double firstDiameter = analysis->getValueAsDouble(0, 0, 0);
```

### ImageDev code

In ImageDev, individual measurements are selected by calling the **select** method, which returns an instance of the **MeasurementName** class inherited from the **Measurement** class. After the computation, results can be accessed by calling the **value** method on the previously returned instance of the **MeasurementName** class. Note that the signature of the **value** method is specific to the selected measurement. For instance, if the measurement is a simple “shape-based” measurement, like **Area**, its signature is **value(int label)** where label is the desired label. In the case of an intensity-based measurement, like **IntensityMinimum**, its signature is **value(int label, int channel)**, allowing the user to access the minimum by label and by image channel.

Note that there also exist generic accessors defined in the Measurement class.



```
// Initialize an analysis object to compute diameter of particles
auto analysis = std::make_shared< AnalysisMsr >();
auto diameter = analysis->select(NativeMeasurements::equivalentDiameter);
// Launch this analysis on a label image previously computed
labelAnalysis(labelImage, labelImage, analysis);
// Get the first result of this analysis
double firstDiameter = diameter-> value(0);
```

## Custom Measurements

Custom measurements are individual measurements created from user-defined formula.

### ImageViz code

In ImageViz, the custom measurements are defined by using the [SoDataMeasureCustom](#) node.

```
// Initialize a new ImageViz analysis object
SoRef<SoLabelAnalysisQuantification> labelAnalysis = new SoLabelAnalysisQuantification;
// Create a circularity factor, theoretically between 0 and 1
SoDataMeasureCustom* myCircularity = new SoDataMeasureCustom();
myCircularity->measureName = "Circularity";
myCircularity->formula = "1.0/Sqrt(" +
SoDataMeasurePredefined::getName(SoDataMeasurePredefined::SHAPE_FACTOR_AP_2D) + ")";
// Select the new measurement in the analysis
labelAnalysis->measureList.set1Value(0, myCircularity);
```

### ImageDev code

In ImageDev, the custom measurements are defined by using the [CustomMeasurement](#) class.

```
// Initialize a new ImageDev analysis object
AnalysisMsr* analysis = new AnalysisMsr();
// Create a circularity factor formula, theoretically between 0 and 1
std::string circularityFormula = "1.0/InverseCircularity";
// Select the new measurement in the analysis after having checked its syntax
if (checkMeasurementFormula(circularityFormula))
{
    auto customMeasurementInfo =
        AnalysisMsr::registerCustomMeasurement("Circularity", circularityFormula, "My
circularity factor.");
    auto circularity = analysis->select(customMeasurementInfo);
}
```

## Geometric objects

### Transformation matrix

Transformation matrices are used to visualize the result of a registration between two images.

#### ImageViz code

In ImageViz, the result of an image registration is directly stored in an **SbMatrix** object that can be used to update a geometric transformation in an Open Inventor scene graph.

You can consult the Registration example of the Open Inventor ImageVizExamples solution for the complete code of the following snippet.

```
SoRef<SoImageRegistrationTransform> registration;
// set the moving image transformation to the registration output transform
updateTransformation( registration->getOutputTransformation() );
```

#### ImageDev code

In ImageDev, the result of an image registration is stored in an **Matrix4d** IOLink object that can be used to update the geometric transformation. It can be converted in an Open Inventor **SbMatrix** with the Converter::toSbMatrix method, available in the IOLinkInterop project of the OpenInventorImageDevExamples packages.

You can consult the AffineRegistration example of the OpenInventorImageDevExamples solution for the complete code of the following snippet.

```
std::shared_ptr< imagedev::AffineRegistration > registration;
// set the moving image transformation to the registration output transform
SbMatrix oivMatrix = Converter::toSbMatrix( registration->outputTransform() );
updateTransformation( oivMatrix );
```

## Line Set

#### ImageViz code

In ImageViz, the result of a centerline extraction is directly stored in an **SoIndexedLineSet** object that can be displayed with Open Inventor.

You can consult the Centerline example of the Open Inventor ImageVizExamples solution for the complete code of the following snippet.

```
SoRef<SoCenterLineApproximation3d> centerlineExtractor = new SoCenterLineApproximation3d();
// get the line set output
SoRef<SoIndexedLineSet> cLineSet = centerlineExtractor->outLineSet.getValue();
```

### ImageDev code

In ImageDev, the result of a centerline extraction is stored in two [ArrayX](#) IOLink objects. They can be converted in an Open Inventor [SoIndexedLineSet](#) with the `Converter::toSoIndexedLineSet` method, available in the `IOLinkInterop` project of the `OpenInventorImageDevExamples` packages.

You can consult the Centerline example of the `OpenInventorImageDevExample` solution for the complete code of the following snippet.

```
imagedev::Centerline3d centerline;
// get the line set output
SoRef< SoIndexedLineSet > cllineSet = Converter::toSoIndexedLineSet(
    centerline.outputIndices(), centerline.outputVertices() );
```

## Behavior changes

Most of the ImageDev algorithms are identical to their ImageViz engine counterparts. However, some of them have been rewritten and their behavior may have slightly changed.

### SoImageRegistrationTransform to AffineRegistration

The [AffineRegistration](#) algorithm no longer supports input images with the BINARY interpretation. To apply an affine registration on binary inputs, their interpretation must be changed to GRAYSCALE beforehand with the `setImageInterpretation` method.

The `outputThreshold` and `optimizerType` new parameters are now available for performing an affine registration.

### SoLabelingProcessing to Labeling2d and Labeling3d

The [SoLabelingProcessing](#) algorithm has been replaced by two algorithms, [Labeling2d](#) and [Labeling3d](#). This new implementation brings the following benefits:

- Management of 8-bit label data type, in addition to 16 and 32-bit only supported with the former version.
- The minimum output type is now settable. Previously, the output was initialized with a 16-bit output and labeling restarted with a 32-bit output when the number of objects exceeded 65535. This new option allows saving a significant computation time in this case.
- The 2D implementation now allows the selection of a 4-neighbors connectivity.
- The 3D computing is faster than before.

## Correspondence table

The following table shows which ImageDev algorithm can be used to replace an ImageViz engine. The engines are sorted by category of the ImageViz module reference manual.

ImageViz Category	ImageViz Engine	ImageDev Algorithm
ArithmeticAndLogic/ArithmeticOperations	SoAbsoluteValueProcessing	AbsoluteValue
ArithmeticAndLogic/ArithmeticOperations	SoArithmeticImageProcessing	ArithmeticOperationWithImage
ArithmeticAndLogic/ArithmeticOperations	SoArithmeticValueProcessing	ArithmeticOperationWithValue
ArithmeticAndLogic/ArithmeticOperations	SoBlendWithImageProcessing	BlendWithImage
ArithmeticAndLogic/ArithmeticOperations	SoBlendWithValueProcessing	BlendWithValue
ArithmeticAndLogic/ArithmeticOperations	SoCombineByMaskProcessing	CombineByMask
ArithmeticAndLogic/ArithmeticOperations	SoDirectionalBlendProcessing2d	DirectionalBlend2d
ArithmeticAndLogic/ArithmeticOperations	SoEigenDecompositionProcessing2d	EigenDecomposition2d
ArithmeticAndLogic/ArithmeticOperations	SoEigenDecompositionProcessing3d	EigenDecomposition3d
ArithmeticAndLogic/ArithmeticOperations	SoImageFormulaProcessing	ImageFormula
ArithmeticAndLogic/ArithmeticOperations	SoMaskImageProcessing	MaskImage
ArithmeticAndLogic/ArithmeticOperations	SoMaxAbsoluteValueProcessing2d	AbsoluteMaximumWithImage
ArithmeticAndLogic/LogicalOperations	SoBitShiftProcessing	BitShift
ArithmeticAndLogic/LogicalOperations	SoInvertImageProcessing	InvertImage
ArithmeticAndLogic/LogicalOperations	SoLogicalImageProcessing	LogicalOperationWithImage
ArithmeticAndLogic/LogicalOperations	SoLogicalNotProcessing	LogicalNot
ArithmeticAndLogic/LogicalOperations	SoLogicalValueProcessing	LogicalOperationWithValue
EdgeDetection/EdgeMarking	SoCannyEdgeDetectionProcessing	CannyEdgeDetector2d CannyEdgeDetector2d
EdgeDetection/EdgeMarking	SoDijkstraShortestPathProcessing2d	DijkstraShortestPath2d
EdgeDetection/EdgeMarking	SoEigenvaluesToStructurenessProcessing2d	EigenvaluesToStructureness2d
EdgeDetection/EdgeMarking	SoEigenvaluesToStructurenessProcessing3d	EigenvaluesToStructureness3d
EdgeDetection/EdgeMarking	SoGaussianGradientTensorProcessing2d	GaussianGradientTensor2d
EdgeDetection/EdgeMarking	SoGaussianGradientTensorProcessing3d	GaussianGradientTensor3d
EdgeDetection/EdgeMarking	SoGradientLocalMaximaProcessing2d	ImageLocalMaxima2d
EdgeDetection/EdgeMarking	SoGradientLocalMaximaProcessing3d	ImageLocalMaxima3d
EdgeDetection/EdgeMarking	SoRidgeDetectionProcessing	RidgeDetection
EdgeDetection/EdgeMarking	SoTensorVotingProcessing2d	TensorVoting2d
EdgeDetection/EdgeMarking	SoZeroCrossingsProcessing2d	ZeroCrossings2d
EdgeDetection/Gradient	SoGradientMagnitudeProcessing2d	VectorToMagnitude2d
EdgeDetection/Gradient	SoGradientMagnitudeProcessing3d	VectorToMagnitude3d
EdgeDetection/Gradient	SoGradientOperatorProcessing2d	GradientOperator2d
EdgeDetection/Gradient	SoGradientOperatorProcessing3d	GradientOperator3d
EdgeDetection/Gradient	SoMorphologicalGradientProcessing	MorphologicalGradient
EdgeDetection/Gradient	SoRadialGradientCenteredProcessing2d	RadialGradientCentered2d

EdgeDetection/Gradient	SoRadialGradientCenteredProcessing3d	RadialGradientCentered3d
EdgeDetection/Gradient	SoRadialGradientContourProcessing2d	RadialGradientContour2d
EdgeDetection/Gradient	SoRadialGradientContourProcessing3d	RadialGradientContour3d
EdgeDetection/Gradient	SoRadialGradientLabelProcessing2d	RadialGradientLabel2d
EdgeDetection/Gradient	SoRadialGradientLabelProcessing3d	RadialGradientLabel3d
EdgeDetection/Laplacian	SoDoBFilterProcessing	DobFilter2d DobFilter3d
EdgeDetection/Laplacian	SoMorphologicalLaplacianProcessing	MorphologicalLaplacian
EdgeDetection/Laplacian	SoRecursiveLaplacianProcessing	RecursiveLaplacian2d RecursiveLaplacian3d
EdgeDetection/OtherDerivatives	SoGaussianDerivativeProcessing	GaussianDerivative2d GaussianDerivative3d
EdgeDetection/OtherDerivatives	SoHessianMatrixProcessing2d	GaussianHessianMatrix2d
EdgeDetection/OtherDerivatives	SoHessianMatrixProcessing3d	GaussianHessianMatrix3d
GeometryAndMatching/GeometricTransforms	SoFlipAxisProcessing2d	FlipImage2d
GeometryAndMatching/GeometricTransforms	SoFlipAxisProcessing3d	FlipImage3d
GeometryAndMatching/GeometricTransforms	SoResampleElasticProcessing2d	ResampleElastic2d
GeometryAndMatching/GeometricTransforms	SoResampleImageProcessing3d	ResampleAffine3d
GeometryAndMatching/GeometricTransforms	SoRotateAroundZProcessing2d	RotateCenterImage2d
GeometryAndMatching/GeometricTransforms	SoRotateProcessing2d	RotateImage2d
GeometryAndMatching/GeometricTransforms	SoRotateProcessing3d	RotateImage3d
GeometryAndMatching/GeometricTransforms	SoScaleByFactorProcessing	RescaleImageByFactor
GeometryAndMatching/GeometricTransforms	SoScaleImageToSizeProcessing	RescaleImage2d RescaleImage3d
GeometryAndMatching/GeometricTransforms	SoSwapAxisProcessing2d	TransposeImage2d
GeometryAndMatching/GeometricTransforms	SoSwapAxisProcessing3d	TransposeImage3d
GeometryAndMatching/GeometricTransforms	SoTranslateProcessing	TranslateImage2d TranslateImage3d
GeometryAndMatching/PatternRecognition	SoBinaryCorrelationProcessing2d	BinaryCorrelation2d
GeometryAndMatching/PatternRecognition	SoGrayscaleCorrelationProcessing2d	CrossCorrelation2d CorrelationByDifference2d CorrelationBySignChange2d
GeometryAndMatching/Registration	SoElasticRegistrationProcessing2d	ElasticRegistration2d
GeometryAndMatching/Registration	SoImagePreAlignmentTransform3d	ImagePrealignment3d
GeometryAndMatching/Registration	SoImageRegistrationTransform	AffineRegistration
ImageAnalysis/GlobalMeasures	SoAreaQuantification2d	Area2d
ImageAnalysis/GlobalMeasures	SoCentroidPathTortuosity Quantification3d	CentroidPathTortuosity3d
ImageAnalysis/GlobalMeasures	SoCurvatureIntegralsQuantification3d	CurvatureIntegrals3d
ImageAnalysis/GlobalMeasures	SoDistanceMeanQuantification2d	DistanceMean2d
ImageAnalysis/GlobalMeasures	SoGlobalAnalysisQuantification	BinaryAnalysis
ImageAnalysis/GlobalMeasures	SoIntensityBinHistogramQuantification	IntensityBinHistogram
ImageAnalysis/GlobalMeasures	SoIntensityHistogramQuantification	IntensityHistogram
ImageAnalysis/GlobalMeasures	SoIntensityIntegralQuantification2d	IntensityIntegral2d
ImageAnalysis/GlobalMeasures	SoIntensityIntegralQuantification3d	IntensityIntegral3d

ImageAnalysis/GlobalMeasures	SoObjectCountQuantification	ObjectCount
ImageAnalysis/IndividualMeasures	SoFilterAnalysisQuantification	FilterAnalysis
ImageAnalysis/IndividualMeasures	SoFilterByMeasureProcessing	FilterByMeasurement
ImageAnalysis/IndividualMeasures	SoLabelAnalysisQuantification	LabelAnalysis
ImageAnalysis/IndividualMeasures	SoLabelFilteringAnalysisQuantification	LabelFilteringAnalysis
ImageAnalysis/IndividualMeasures	SoLabelFilteringProcessing	LabelFiltering
ImageAnalysis/IndividualMeasures	SoMeasureImageProcessing	MeasurementToImage
ImageAnalysis/IndividualMeasures	SoSieveLabelingProcessing	SieveLabeling
ImageAnalysis/Morphometry	SoAverageObjectAreaPerSliceQuantification3d	AverageObjectAreaPerSlice3d
ImageAnalysis/Morphometry	SoAverageObjectNumberPerSliceQuantification3d	AverageObjectCountPerSlice3d
ImageAnalysis/Morphometry	SoAverageObjectThicknessQuantification3d	AverageObjectThickness3d
ImageAnalysis/Morphometry	SoAverageSpaceThicknessQuantification3d	AverageSpaceThickness3d
ImageAnalysis/Morphometry	SoDegreeOfAnisotropyQuantification	DegreeOfAnisotropy
ImageAnalysis/Morphometry	SoFractalDimensionQuantification	FractalDimension
ImageAnalysis/Morphometry	SoFragmentationIndexQuantification	FragmentationIndex
ImageAnalysis/Morphometry	SoImageVolumeQuantification	TotalImageVolume
ImageAnalysis/Morphometry	SoInertiaMomentProcessing2d	InertiaMoment2d
ImageAnalysis/Morphometry	SoObjectBoundaryAreaQuantification	ObjectBoundaryArea
ImageAnalysis/Morphometry	SoObjectImageRatioQuantification	ObjectImageRatio
ImageAnalysis/Morphometry	SoObjectLinearDensityQuantification3d	ObjectLinearDensity3d
ImageAnalysis/Morphometry	SoObjectSpecificSurfaceQuantification	ObjectSpecificSurface
ImageAnalysis/Morphometry	SoObjectSurfaceDensityQuantification	ObjectSurfaceDensity
ImageAnalysis/Morphometry	SoObjectVolumeQuantification	ObjectVolume
ImageAnalysis/Morphometry	SoPorosityPercentageQuantification3d	PorosityPercentage3d
ImageAnalysis/Morphometry	SoStructureModelIndexQuantification3d	StructureModelIndex3d
ImageAnalysis/Morphometry	SoTotalImageBoundaryAreaQuantification	TotalImageBoundaryArea
ImageAnalysis/Statistics	SoAutoThresholdingQuantification	AutoThresholdingValue
ImageAnalysis/Statistics	SoCompareImageQuantification	CompareImage
ImageAnalysis/Statistics	SoCompareValueQuantification	CompareValue
ImageAnalysis/Statistics	SoCooccurrenceQuantification2d	Cooccurrence2d
ImageAnalysis/Statistics	SoIntensityExtremaQuantification	IntensityExtrema
ImageAnalysis/Statistics	SoIntensityStatisticsQuantification	IntensityStatistics
ImageAnalysis/Statistics	SoLocalMaximaQuantification	LocalMaxima2d LocalMaxima3d
ImageAnalysis/Statistics	SoMaskedStatisticsQuantification	IntensityMaskedStatistics
ImageAnalysis/Statistics	SoMeasureGaussianNoiseQuantification	MeasureGaussianNoise
ImageFiltering/ColorTransforms	SoColorAntialiasingProcessing2d	ColorAntialiasing2d
ImageFiltering/ColorTransforms	SoColorGetPlaneProcessing2d	GetColorChannel
ImageFiltering/ColorTransforms	SoColorSpaceConversionProcessing	ColorSpaceConversion
ImageFiltering/ColorTransforms	SoColorToGrayscaleProcessing	ColorToGrayscale

ImageFiltering/ColorTransforms	SoColorToLightnessProcessing	ColorToLightness
ImageFiltering/ColorTransforms	SoDecorrelationStretchProcessing2d	DecorrelationStretch2d
ImageFiltering/ColorTransforms	SoGrayscaleToColorProcessing	GrayscaleToColor
ImageFiltering/FrequencyDomain	SoCartesianToPolarProcessing2d	CartesianToPolar2d
ImageFiltering/FrequencyDomain	SoComplexCenteredFFTIverseProcessing	ComplexCenteredFftInverse
ImageFiltering/FrequencyDomain	SoComplexCenteredFFTProcessing	ComplexCenteredFft
ImageFiltering/FrequencyDomain	SoComplexFFTIverseProcessing	ComplexFftInverse
ImageFiltering/FrequencyDomain	SoComplexFFTProcessing	ComplexFft
ImageFiltering/FrequencyDomain	SoConvolutionWithImageProcessing2d	ConvolutionWithImage2d
ImageFiltering/FrequencyDomain	SoGaborFilteringProcessing2d	GaborFiltering2d
ImageFiltering/FrequencyDomain	SoPolarToCartesianProcessing2d	PolarToCartesian2d
ImageFiltering/FrequencyDomain	SoSwapQuadrantsProcessing	SwapQuadrants
ImageFiltering/GrayscaleTransforms	SoAdaptiveHistogramEqualizationProcessing	AdaptiveHistogramEqualization
ImageFiltering/GrayscaleTransforms	SoApplyGrayscaleLutProcessing	ApplyGrayscaleLut
ImageFiltering/GrayscaleTransforms	SoBackgroundImageProcessing2d	BackgroundImage2d
ImageFiltering/GrayscaleTransforms	SoHistogramEqualizationProcessing	HistogramEqualization
ImageFiltering/GrayscaleTransforms	SoMatchContrastProcessing	MatchContrast
ImageFiltering/GrayscaleTransforms	SoMathematicalFunctionProcessing	Exponential Logarithm Square SquareRoot
ImageFiltering/GrayscaleTransforms	SoRescaleIntensityProcessing	RescaleIntensity
ImageFiltering/GrayscaleTransforms	SoShadingCorrectionProcessing	ShadingCorrection
ImageFiltering/Sharpening	SoDeblurProcessing2d	Deblurring2d
ImageFiltering/Sharpening	SoDelineateProcessing	Delineate2d Delineate3d
ImageFiltering/Sharpening	SoShadeProcessing2d	Shade2d
ImageFiltering/Sharpening	SoUnsharpMaskingProcessing	UnsharpMasking2d UnsharpMasking3d
ImageFiltering/SmoothingAndDenoising	SoBilateralFilterProcessing	BilateralFilter2d BilateralFilter3d
ImageFiltering/SmoothingAndDenoising	SoBoxFilterProcessing	BoxFilter2d BoxFilter3d
ImageFiltering/SmoothingAndDenoising	SoCurvatureDrivenDiffusionProcessing	CurvatureDrivenDiffusion
ImageFiltering/SmoothingAndDenoising	SoDespeckleProcessing	Despeckle2d Despeckle3d
ImageFiltering/SmoothingAndDenoising	SoFlowInpaintingProcessing	FlowInpainting
ImageFiltering/SmoothingAndDenoising	SoGaussianBlurFilterProcessing	GaussianFilter2d GaussianFilter3d
ImageFiltering/SmoothingAndDenoising	SoMajorityFilterProcessing	MajorityFilter2d MajorityFilter3d
ImageFiltering/SmoothingAndDenoising	SoMedianFilterProcessing	MedianFilter2d MedianFilter3d
ImageFiltering/SmoothingAndDenoising	SoNagaoFilterProcessing2d	NagaoFilter2d
ImageFiltering/SmoothingAndDenoising	SoNagaoFilterProcessing3d	NagaoFilter3d
ImageFiltering/SmoothingAndDenoising	SoNonLocalMeansFilterProcessing	NonLocalMeansFilter2d NonLocalMeansFilter3d

ImageFiltering/SmoothingAndDenoising	SoRecursiveExponentialFilterProcessing	RecursiveExponentialFilter2d RecursiveExponentialFilter3d
ImageFiltering/SmoothingAndDenoising	SoSNNFilterProcessing	SnnFilter2d SnnFilter3d
ImageFiltering/SmoothingAndDenoising	SoSigmaFilterProcessing	SigmaFilter2d SigmaFilter3d
ImageFiltering/TextureFilters	SoAutoCorrelationProcessing2d	LocalAutoCorrelation2d
ImageFiltering/TextureFilters	SoImageCurvatureProcessing2d	ImageCurvature2d
ImageFiltering/TextureFilters	SoImageCurvatureProcessing3d	ImageCurvature3d
ImageFiltering/TextureFilters	SoLocalStatisticsProcessing	MeanFilter2d MeanFilter3d VarianceFilter2d VarianceFilter3d SkewnessFilter2d SkewnessFilter3d KurtosisFilter2d KurtosisFilter3d ContrastFilter2d ContrastFilter3d VariationFilter2d VariationFilter3d EnergyFilter2d EnergyFilter3d EntropyFilter2d EntropyFilter3d
ImageFiltering/TextureFilters	SoMultiscaleStructureEnhancementProcessing2d	StructureEnhancementFilter2d
ImageFiltering/TextureFilters	SoMultiscaleStructureEnhancementProcessing3d	StructureEnhancementFilter3d
ImageManipulation/ImageEditing	SoConvertImageProcessing	ConvertImage
ImageManipulation/ImageEditing	SoCopyImageProcessing	CopyImage
ImageManipulation/ImageEditing	SoCropImageProcessing	CropImage2d CropImage3d
ImageManipulation/ImageEditing	SoDeinterlaceFramesProcessing2d	DeinterlaceFrames2d
ImageManipulation/ImageEditing	SoGetObliquePlaneFromVolumeProcessing3d	GetObliqueSliceFromVolume3d
ImageManipulation/ImageEditing	SoGetPlaneFromVolumeProcessing3d	GetSliceFromVolume3d
ImageManipulation/ImageEditing	SoInterlaceFramesProcessing2d	InterlaceFrames2d
ImageManipulation/ImageEditing	SoResetImageProcessing	ResetImage
ImageManipulation/ImageEditing	SoSampleImageProcessing	SampleImage2d SampleImage3d
ImageManipulation/ImageEditing	SoSetPlaneToVolumeProcessing3d	SetSliceToVolume3d
ImageManipulation/StackOperations	SoImageStackProjectionProcessing3d	ImageStackProjection3d
ImageManipulation/SyntheticImages	SoRandomImageGeneration	RandomImage2d RandomImage3d
ImageManipulation/SyntheticImages	SoRandomSphereGeneration3d	RandomSphereImage3d
ImageSegmentation/Binarization	SoAdaptiveThresholdingProcessing	FeatureAdaptiveThresholding
ImageSegmentation/Binarization	SoAutoThresholdingProcessing	AutoThresholdingBright AutoThresholdingDark
ImageSegmentation/Binarization	SoColorThresholdingProcessing	ColorThresholding
ImageSegmentation/Binarization	SoHysteresisThresholdingProcessing	HysteresisThresholding
ImageSegmentation/Binarization	SoLocalAdaptiveThresholdProcessing	AdaptiveThreshold
ImageSegmentation/Binarization	SoThresholdingByCriterionProcessing	ThresholdingByCriterion
ImageSegmentation/Binarization	SoThresholdingProcessing	Thresholding
ImageSegmentation/Binarization	SoTopHatProcessing	TopHat
ImageSegmentation/Classification	SoAutoIntensityClassificationProcessing	AutoIntensityClassification
ImageSegmentation/Classification	SoSupervisedTextureClassificationProcessing2d	SupervisedTextureClassification2d
ImageSegmentation/Classification	SoSupervisedTextureClassificationProcessing3d	SupervisedTextureClassification3d



ImageSegmentation/ComputationalGeometry	SoCenterLineApproximation3d	Centerline3d
ImageSegmentation/ComputationalGeometry	SoObjectToSegmentApproximation2d	Not Available
ImageSegmentation/ComputationalGeometry	SoPolylineExtrusionApproximation3d	PolylineExtrusion3d
ImageSegmentation/ComputationalGeometry	SoPolylineResamplerApproximation2d	PolylineResampler2d
ImageSegmentation/ComputationalGeometry	SoPolylineResamplerApproximation3d	PolylineResampler3d
ImageSegmentation/ComputationalGeometry	SoQuadrilateralMeshSlicerApproximation3d	Not Available
ImageSegmentation/ComputationalGeometry	SoSurfaceFittingApproximation3d	SurfaceFitting3d
ImageSegmentation/ComputationalGeometry	SoSurfaceUnfoldingProcessing3d	SurfaceUnfolding3d
ImageSegmentation/FeatureSelection	SoBorderKillProcessing	BorderKill
ImageSegmentation/FeatureSelection	SoLabelInterfacesProcessing	LabelInterfaces
ImageSegmentation/FeatureSelection	SoObjectBoundariesProcessing	ObjectBoundaries
ImageSegmentation/FeatureSelection	SoRemoveSmallHolesProcessing	RemoveSmallHoles
ImageSegmentation/FeatureSelection	SoRemoveSmallSpotsProcessing	RemoveSmallSpots
ImageSegmentation/Labeling	SoAddObjectToLabelProcessing	AddObjectToLabel
ImageSegmentation/Labeling	SoAssignLabelProcessing	AssignLabel
ImageSegmentation/Labeling	SoAutoSegmentation3PhasesProcessing	AutoSegmentation3Phases
ImageSegmentation/Labeling	SoGroupCloseLabelsIProcessing2d	GroupCloseLabelsI2d
ImageSegmentation/Labeling	SoLabelingProcessing	Labeling2d Labeling3d
ImageSegmentation/Labeling	SoObjectToRegionProcessing2d	Not Available
ImageSegmentation/Labeling	SoRegionToObjectProcessing2d	Not Available
ImageSegmentation/Labeling	SoReorderLabelsProcessing	ReorderLabels
ImageSegmentation/RegionGrowing	SoFloodFillThresholdProcessing	FloodFillThreshold2d FloodFillThreshold3d
ImageSegmentation/RegionGrowing	SoMarkerBasedWatershedProcessing	MarkerBasedWatershed2d MarkerBasedWatershed3d
ImageSegmentation/SeparatingAndFilling	SoConvexHullProcessing2d	ConvexHull2d
ImageSegmentation/SeparatingAndFilling	SoExpandLabelsProcessing	ExpandLabelBoundaries
ImageSegmentation/SeparatingAndFilling	SoFillHolesProcessing2d	FillHoles2d
ImageSegmentation/SeparatingAndFilling	SoFillHolesProcessing3d	FillHoles3d
ImageSegmentation/SeparatingAndFilling	SoFillImageBorderProcessing3d	FillImageBorder3d
ImageSegmentation/SeparatingAndFilling	SoHExtremaWatershedProcessing	HExtremaWatershed
ImageSegmentation/SeparatingAndFilling	SoSeparateObjectsProcessing	SeparateObjects
MathematicalMorphology/DistanceMaps	SoChamferDistanceMapProcessing2d	DistanceMap2d
MathematicalMorphology/DistanceMaps	SoChamferDistanceMapProcessing3d	DistanceMap3d
MathematicalMorphology/DistanceMaps	SoChessboardDistanceMapProcessing	DistanceMap2d DistanceMap3d
MathematicalMorphology/DistanceMaps	SoClosestBoundaryPointsProcessing2d	ClosestBoundaryPoints2d
MathematicalMorphology/DistanceMaps	SoClosestBoundaryPointsProcessing3d	ClosestBoundaryPoints3d
MathematicalMorphology/DistanceMaps	SoEuclideanDistanceMapProcessing3d	EuclideanDistanceMap3d
MathematicalMorphology/DistanceMaps	SoGeodesicDistanceMapProcessing	GeodesicDistanceMap
MathematicalMorphology/DistanceMaps	SoGeodesicPropagationProcessing2d	GeodesicPropagation2d

MathematicalMorphology/DistanceMaps	SoLocalThicknessMapProcessing3d	LocalThicknessMap3d
MathematicalMorphology/DistanceMaps	SoTimeMapProcessing	TimeMap
MathematicalMorphology/ErosionAndDilation	SoDilationBallProcessing3d	DilationBall3d
MathematicalMorphology/ErosionAndDilation	SoDilationCubeProcessing	Dilation2d Dilation3d
MathematicalMorphology/ErosionAndDilation	SoDilationDiskProcessing2d	DilationDisk2d
MathematicalMorphology/ErosionAndDilation	SoDilationDiskProcessing3d	DilationDisk3d
MathematicalMorphology/ErosionAndDilation	SoDilationLineProcessing2d	DilationLine2d
MathematicalMorphology/ErosionAndDilation	SoDilationLineProcessing3d	DilationLine3d
MathematicalMorphology/ErosionAndDilation	SoDilationSquareColorProcessing2d	DilationColor2d
MathematicalMorphology/ErosionAndDilation	SoErosionBallProcessing3d	ErosionBall3d
MathematicalMorphology/ErosionAndDilation	SoErosionCubeProcessing	Erosion2d Erosion3d
MathematicalMorphology/ErosionAndDilation	SoErosionDiskProcessing2d	ErosionDisk2d
MathematicalMorphology/ErosionAndDilation	SoErosionDiskProcessing3d	ErosionDisk3d
MathematicalMorphology/ErosionAndDilation	SoErosionLineProcessing2d	ErosionLine2d
MathematicalMorphology/ErosionAndDilation	SoErosionLineProcessing3d	ErosionLine3d
MathematicalMorphology/ErosionAndDilation	SoErosionSquareColorProcessing2d	ErosionColor2d
MathematicalMorphology/GeodesicTransformations	SoExtremalImpositionProcessing	Not Available
MathematicalMorphology/GeodesicTransformations	SoGrayscaleResconstructionProcessing	GrayscaleResconstruction2d GrayscaleResconstruction3d
MathematicalMorphology/GeodesicTransformations	SoHExtremaProcessing	Hminima HMaxima
MathematicalMorphology/GeodesicTransformations	SoReconstructionFromMarkersProcessing	ReconstructionFromMarkers2d ReconstructionFromMarkers3d
MathematicalMorphology/GeodesicTransformations	SoRegionalExtremaProcessing	RegionalMinima RegionalMaxima
MathematicalMorphology/GeodesicTransformations	SoUltimateErosionProcessing	UltimateErosion
MathematicalMorphology/HitOrMissAndSkeleton	SoApplyMorphologicalLutProcessing2d	ApplyMorphologicalLut2d
MathematicalMorphology/HitOrMissAndSkeleton	SoApplyMorphologicalLutProcessing3d	ApplyMorphologicalLut3d
MathematicalMorphology/HitOrMissAndSkeleton	SoCentroidProcessing2d	Centroid2d
MathematicalMorphology/HitOrMissAndSkeleton	SoEndPointsProcessing2d	EndPoints2d
MathematicalMorphology/HitOrMissAndSkeleton	SoEndPointsProcessing3d	EndPoints3d
MathematicalMorphology/HitOrMissAndSkeleton	SoInfluenceZonesProcessing	InfluenceZones
MathematicalMorphology/HitOrMissAndSkeleton	SoInteriorPointsProcessing2d	InteriorPoints2d
MathematicalMorphology/HitOrMissAndSkeleton	SoIsolatedPointsProcessing2d	IsolatedPoints2d
MathematicalMorphology/HitOrMissAndSkeleton	SoIsolatedPointsProcessing3d	IsolatedPoints3d
MathematicalMorphology/HitOrMissAndSkeleton	SoPruneProcessing2d	Pruning2d
MathematicalMorphology/HitOrMissAndSkeleton	SoPruningProcessing2d	Pruning2d
MathematicalMorphology/HitOrMissAndSkeleton	SoPruningProcessing3d	Pruning3d
MathematicalMorphology/HitOrMissAndSkeleton	SoSkeletonProcessing2d	Skeleton
MathematicalMorphology/HitOrMissAndSkeleton	SoTriplePointsProcessing2d	TriplePoints2d
MathematicalMorphology/OpeningAndClosing	SoClosingBallByReconstructionProcessing3d	ClosingBallByReconstruction3d
MathematicalMorphology/OpeningAndClosing	SoClosingBallProcessing3d	ClosingBall3d

MathematicalMorphology/OpeningAndClosing	SoClosingByReconstructionProcessing	ClosingByReconstruction
MathematicalMorphology/OpeningAndClosing	SoClosingCubeProcessing	ClosingCube
MathematicalMorphology/OpeningAndClosing	SoClosingDiskByReconstructionProcessing2d	ClosingDiskByReconstruction2d
MathematicalMorphology/OpeningAndClosing	SoClosingDiskProcessing2d	ClosingDisk2d
MathematicalMorphology/OpeningAndClosing	SoClosingDiskProcessing3d	ClosingDisk3d
MathematicalMorphology/OpeningAndClosing	SoClosingLineMinimumProcessing	ClosingLineMinimum
MathematicalMorphology/OpeningAndClosing	SoClosingLineProcessing2d	ClosingLine2d
MathematicalMorphology/OpeningAndClosing	SoClosingLineProcessing3d	ClosingLine3d
MathematicalMorphology/OpeningAndClosing	SoClosingSquareColorProcessing2d	ClosingColor2d
MathematicalMorphology/OpeningAndClosing	SoOpeningBallByReconstructionProcessing3d	OpeningBallByReconstruction3d
MathematicalMorphology/OpeningAndClosing	SoOpeningBallProcessing3d	OpeningBall3d
MathematicalMorphology/OpeningAndClosing	SoOpeningByReconstructionProcessing	OpeningByReconstruction
MathematicalMorphology/OpeningAndClosing	SoOpeningCubeProcessing	OpeningCube
MathematicalMorphology/OpeningAndClosing	SoOpeningDiskByReconstructionProcessing2d	OpeningDiskByReconstruction2d
MathematicalMorphology/OpeningAndClosing	SoOpeningDiskProcessing2d	OpeningDisk2d
MathematicalMorphology/OpeningAndClosing	SoOpeningDiskProcessing3d	OpeningDisk3d
MathematicalMorphology/OpeningAndClosing	SoOpeningLineMaximumProcessing	OpeningLineMaximum
MathematicalMorphology/OpeningAndClosing	SoOpeningLineProcessing2d	OpeningLine2d
MathematicalMorphology/OpeningAndClosing	SoOpeningLineProcessing3d	OpeningLine3d
MathematicalMorphology/OpeningAndClosing	SoOpeningSquareColorProcessing2d	OpeningColor2d