Release guide

ERDAS IMAGINE 2023 Update 2

Version 16.8.2

15 November 2024

Contents

About this release

This document describes enhancements in ERDAS IMAGINE 2023 Update 2 (v16.8.2), including IMAGINE Photogrammetry (formerly LPS Core).

ERDAS IMAGINE 2023 Update 2 is a full installer, meaning you can install it either on top of ERDAS IMAGINE 2023 or on its own (i.e., there is no requirement to install, or uninstall, ERDAS IMAGINE 2023 first).

Since ERDAS IMAGINE 2023 Update 2 is a full installer this document also describes all the new features introduced with the original release of ERDAS IMAGINE 2023 and Update 1, in the second half of the document.

ERDAS IMAGINE 2023 Update 2 can be installed alongside prior major versions of ERDAS IMAGINE if desired. I.e. there is no requirement to uninstall any prior version such as ERDAS IMAGINE 2022 Update 2 – both versions can be installed and used).

ERDAS IMAGINE 2023 Update 2 includes both enhancements and fixes. For information on fixes to customer reported issues that were made in ERDAS IMAGINE 2023 Update 2, see Issues resolved.

This document is an overview and does not provide all details about the product's capabilities. See the product description, online help and other documents provided with ERDAS IMAGINE for more information.

New features for Update 2 include:

- **Cloud Optimized GeoTIFF (COG) Exporter:** Create data ready for cloud sharing
- **Improved Stripped TIFF handling:** Zoom imagery much more rapidly without the need to build pyramids
- **Improved RPF exporter and direct read capabilities:** Better map display quality
- **Normalized Difference Water Index for Water Bodies:** Detect surface waters in wetland environments
- **Set Geometry Z operator:** Convert 2D geometries to 3D
- **Updated Iterator operator:** Define which input ports will be iterated upon
- **Apply Image Styling operator:** Burn raster data through a set of Image Chain display styling parameters
- **Land / Water Delineation from RCM:** Map flood extents during cloudy conditions
- **Miscellaneous solutions to over 20 customer-reported issues**

ERDAS IMAGINE product tiers

ERDAS IMAGINE performs advanced remote sensing analysis and spatial modeling to create new information. In addition, with ERDAS IMAGINE, you can visualize your results in 2D, 3D, movies and on cartographic-quality map compositions. The core of the ERDAS IMAGINE product suite is engineered to scale with your geospatial data production needs. Optional modules (add-ons) providing specialized functionalities are also available to enhance productivity and capabilities.

IMAGINE Essentials is the entry-level image processing product for map creation and simple feature collection tools. IMAGINE Essentials enables serial batch processing.

IMAGINE Advantage enables advanced spectral processing, image registration, mosaicking and image analysis and change detection capabilities. IMAGINE Advantage allows you to process parallel batches for faster output.

IMAGINE Professional includes a production toolset for spatial modeling, image classification, feature extraction and advanced spectral, hyperspectral and radar processing.

IMAGINE Photogrammetry maximizes productivity with state-of-the-art photogrammetric satellite and aerial image processing algorithms.

New platforms

Geospatial Licensing 2023

Geospatial Licensing 2023 is required to provide concurrent licenses to 2023 products, including ERDAS IMAGINE 2023 Update 2. Versions of Geospatial Licensing prior to 2023 cannot successfully license ERDAS IMAGINE 2023 Update 2.

ERDAS IMAGINE installers no longer attempt to automatically install Geospatial Licensing tools as part of the installer. To use geospatial licensing tools — for example, to set up a floating/concurrent license server — you must download and install Geospatial Licensing 2023 separately.

Version 16.8 license files are required to run this release. Customers who have current Maintenance contracts and valid v16.8 licenses already do not need additional licensing to use ERDAS IMAGINE 2023 Update 2.

You can find the appropriate download in the licensing portal.

Security vulnerabilities

A section is provided on the Hexagon community site to track security vulnerabilities and their potential impacts on Hexagon products, including ERDAS IMAGINE. Please refer to this link for further information:

Technical alerts (hexagon.com)

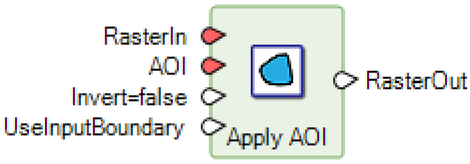
New technology for ERDAS IMAGINE 2023 Update 2

Spatial Modeler

Update 2 extends the capabilities offered by Spatial Modeler with new and improved operators, including an Apply Image Styling operator.

Updated Spatial Modeler operators in ERDAS IMAGINE 2023 Update 2

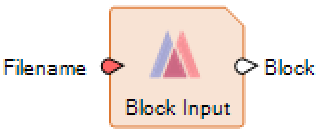
Apply AOI



Previously customers would attempt to use rectangular area geometries to clip out subsets from imagery. The rectangular region's dimensions were expected to encompass a very specific set of pixels, such as a rectangle measuring 1,000 m by 1,000 m was expected to extract 100 rows by 100 columns from an image with 10m pixels. However, due to the use of pixel centers as the anchoring coordinate, the subset image ended up being 101 rows by 101 columns instead.

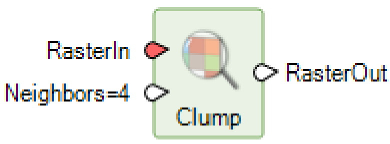
Consequently, Apply AOI has been updated to internally use the same improvements added to Convert To Raster and treat the vertex coordinates of the input geometries as being aligned to the edges of pixel cells, not the centers. In the above example this should generally result in an output image of 100 by 100 pixels.

Block Input



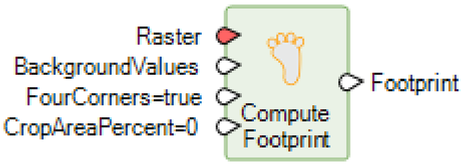
An IMAGINE Photogrammetry project (Block File *.blk) can now be added to the Spatial Model Editor by simply dragging the file from a Windows Explorer, or the Contents panel, to the editor canvas. A Block Input operator will be created and the name of the dragged file defined on its input **Filename** port.

Clump



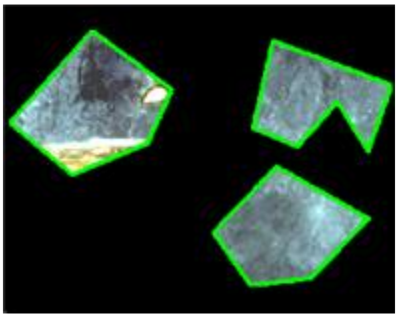
The Clump operator has been modified to re-block input data if necessary. In some instances (for example with stripped data) this was found to speed up processing by as much as five to 10 times.

Compute Footprint

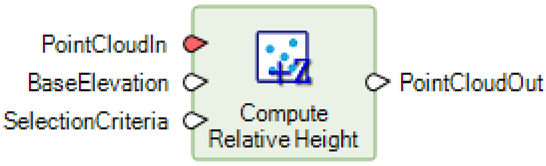


This operator generates the not-necessarily-rectangular footprint for raster data. The output Footprint is the outline polygon that defines the extent/boundary of the input Raster's valid (non-NoData, as determined by mask or supplied background values) area.

Previously, if the **FourCorner** port was set to False, the edge tracing would only encompass the first non-regular clump of imagery. This has been corrected so that all non-contiguous clumps of imagery will be traced to form multiple polygon geometries representing the footprint(s).



Compute Relative Height



This operator computes the difference between point cloud Z values and a base elevation as specified on port **BaseElevation** and updates the RelativeHeight attribute (automatically adding the attribute if it does not yet exist) of the selected points.

This is useful, for example, if you wish to select points based on height above ground. To achieve this, use this operator with a bare earth DEM or point cloud as the input **BaseElevation**. Then downstream of this operator, you can use the operator Point Cloud Build Selection Criteria, specifying MinimumHeight and/or MaximumHeight, and setting its HeightSource port value to "RelativeHeight" in order to constrain downstream operations on **PointCloudOut** to your height above ground selection.

If a raster is used for the **BaseElevation**, any points falling into NoData areas will have Not a Number (NaN) as the relative height value.

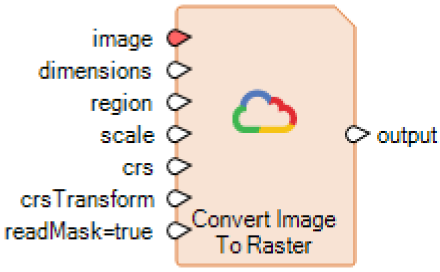
If the input is a point cloud, a TIN surface will be created from the input points. Any points falling outside of the TIN's convex hull will have NaN as the relative height value.

If **SelectionCriteria** has a value, any points that do not meet the selection criteria will have NaN as the relative height value.

This operator differs from Normalize Height in that the point Z values are not altered, and no points are removed if they overlay NoData areas if a raster is used as the **BaseElevation**.

Note that the RelativeHeight values are only persisted by Point Cloud Output if the output file format supports custom attributes.

Convert Image To Raster



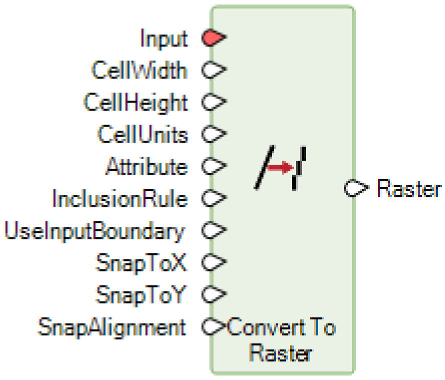
Note: This operator requires an ERDAS IMAGINE LiveLink for Google Earth Engine license.

This operator uses the Google Earth Engine Image expression, which is connected to the **image** port to create an IMAGINE.Raster on the **output** port. This enables raster tiles to be read by any of the Spatial Modeler operators that take an IMAGINE.Raster as an input. All of the computation defined by the input Image expression is performed on the Google cloud infrastructure. Computation is only performed for the requested tiles at the requested resolution.

New for Update 2, the extent, resolution and coordinate system of the output is determined by the configuration of the ports dimensions, region, scale, crs and crsTransform, which are used to determine the Pixel Grid for the output image. See the section Understanding the Pixel Grid. in the user guide for details. If none of these ports has data, the entire image is output at its native resolution.

This new capability permits characteristics such as the pixel resolution (scale) of the output Raster to be controlled.

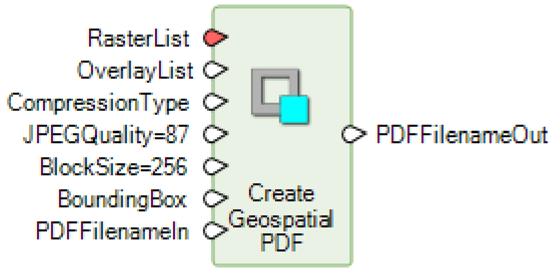
Convert To Raster



Convert To Raster rasterizes its input data at the specified resolution. In the case of Features input, a specific attribute field may be used to fill the rasterized geometries.

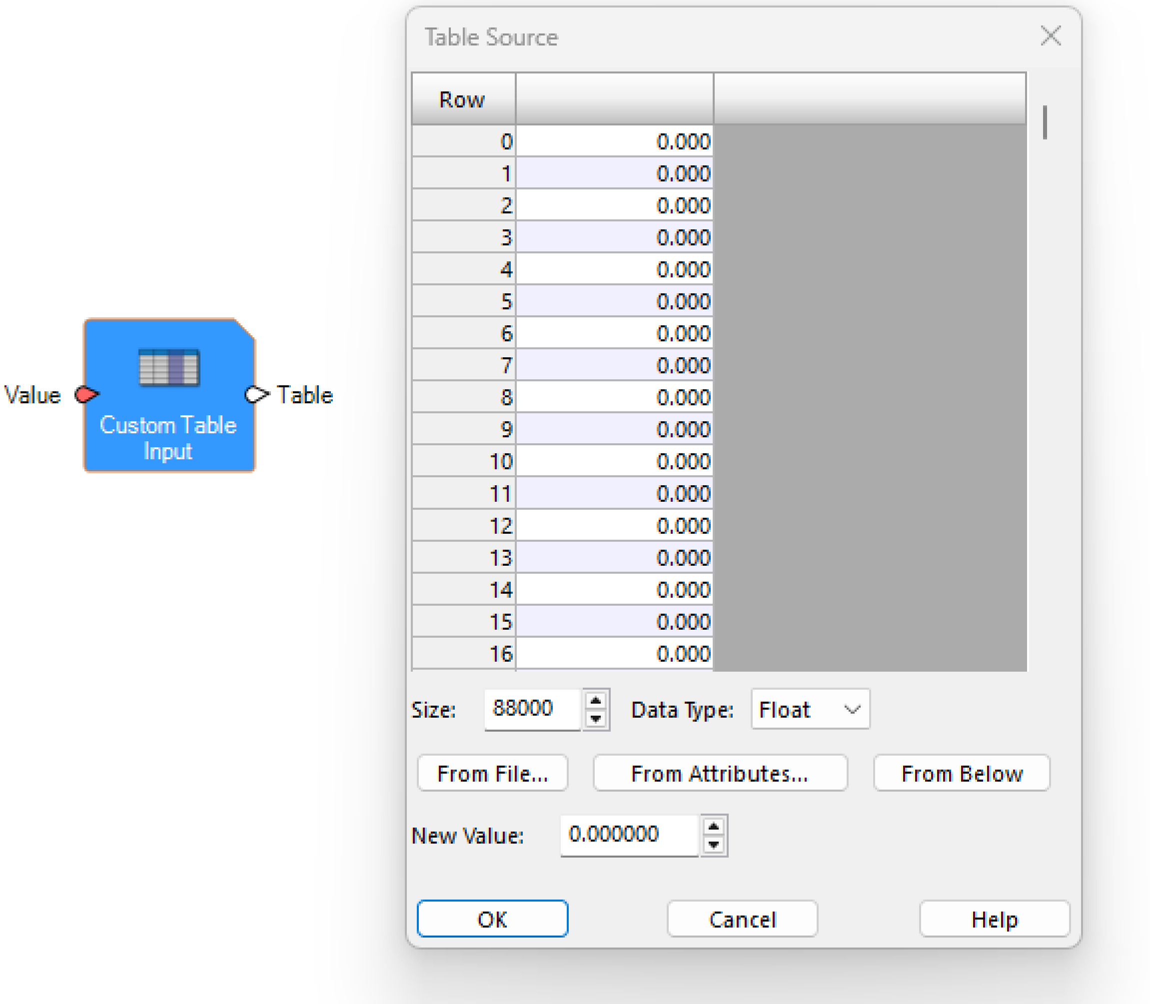
The operator has been updated to support raster alignment/snapping for greater control over how the newly created raster grid (e.g., pixel edges and corners) align with other datasets or specific map coordinates. As mentioned above this new capability is now used in the Apply AOI operator.

Create Geospatial PDF



The dialog used to define the input raster and vector datatypes did not previously filter on all the ERDAS IMAGINE supported data types. It has been updated to do so.

Custom Table Input



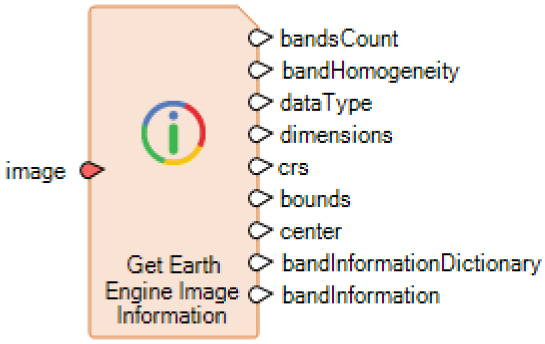
The Custom Table Input operator now supports definition of a Table size with more than 65,536 rows.

Fill DEM Holes

The operator identifies NoData pixels or areas, otherwise known as holes, in the input raster. It uses interpolation between pixels that border the NoData areas, to determine the DN values used to update NoData pixels. Only NoData pixels within the convex hull of the input pixels are modified. All other pixel DN values are passed through unchanged.

For Update 2 we have added the ability to control the interpolation method. Either the original TIN-based method can be used, or a new Focal Mean option. Focal Mean is best suited to flat terrain, yielding a smoother surface than TIN.

Get Earth Engine Image Information



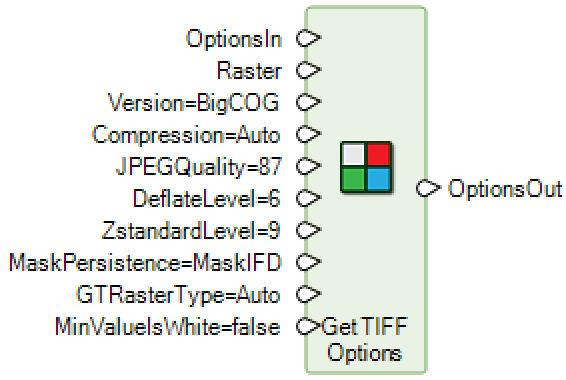
Note: This operator requires an ERDAS IMAGINE LiveLink for Google Earth Engine license.

This operator evaluates the Google Earth Engine Image expression given on the image port and returns a number of pieces of useful information about the resulting image.

The new **bandInformationDictionary** output port provides a Dictionary of the complete set of information for all bands (which can be of homogenous types in Google Earth Engine Images) which can be accessed using the Dicitonary Item operator.

The information provided in the Dictionary output on the **bandInformationDictionary** port can also be used to provide Pixel Grid definitions which might be necessary as inputs to the Convert Image to Raster operator and the various Earth Engine Export operators.

Get TIFF Options



The Get TIFF Options operator has been updated in several ways to provide additional control over the formatting of downstream Raster Output operators which are set to produce TIFF formatted files.

Firstly is the ability to specify that data should adhere to the Cloud Optimized GeoTIFF (COG) specification.

According to the USGS a "Cloud Optimized GeoTIFF (COG) is a GeoTIFF file with an internal organization that enables more efficient workflows in the cloud environment. It does this by leveraging the ability of clients issuing HTTP GET range requests to ask for just the parts of a file they need."

However, a COG image can generally also be used as a regular GeoTIFF file in most geospatial applications (such as ERDAS IMAGINE and GeoMedia) and provides efficiencies of reading due to its organization into tiles (rather than strips), compression, and the incorporation of embedded pyramid levels.

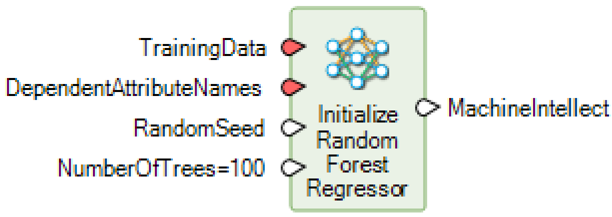
Secondly, the ability to compress the raster data using Zstandard compression has been added.

Zstandard was designed to give a compression ratio comparable to that of the Deflate algorithm (developed in 1991 and used in the original ZIP and gzip programs), but faster, especially for decompression. It is tuneable with compression levels ranging from 1 (fastest) to 22 (slowest in compression speed, but best compression ratio).

Compression speed can vary by a factor of 20 or more between the fastest and slowest levels, while decompression is uniformly fast, varying by less than 20% between the fastest and slowest levels.

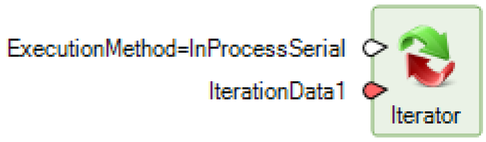
Lastly, an Auto option has been added for the compression type. This option automatically determines the compression type to be used based on the Version port and Preference values.

Initialize Random Forest Regressor



This operator defines and trains a random forest regressor which is used as an input for estimating data using the Predict Using Machine Learning operator. Similar to the improvement to Initialize Random Forest in Update 1, Initialize Random Forest Regressor has been extended to specify the number of trees, enabling tuning of model robustness.

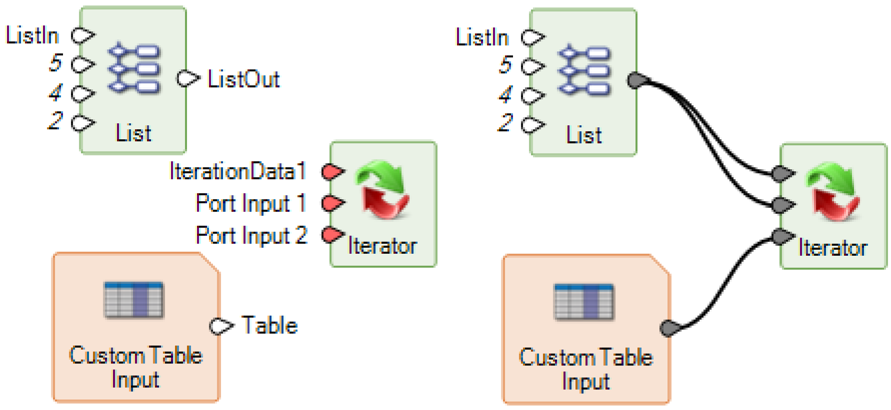
Iterator



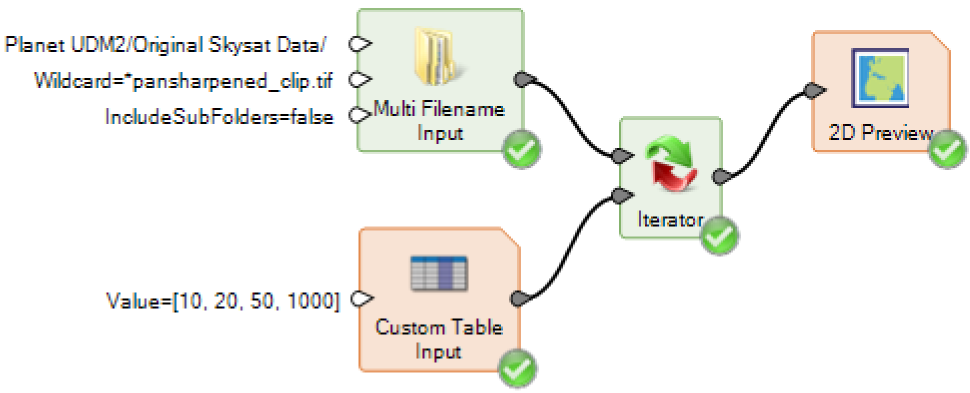
The Iterator operator takes the items off of a List (or Table) and puts each one through a sub-model the user builds inside the Iterator operator. This is done for each item in the List once.

If more than one list is added to this operator, then the number of iterations performed is, by default, based on the number of items in the smallest list. The larger lists that are an input to that same Iterator operator have only that number of items put through the sub-model in the Iterator operator, starting from the first item. The rest are ignored. This default behavior can be altered through the use of the new IterationData<N> ports

The data on which to iterate can now be controlled through the IterationData<N> ports, which are added by selecting "Add Port" and removed by selecting "Remove Port". This allows a List/Table to be passed in and used as a List/Table in each iteration. If an IterationData<N> is used, only the data on those ports will be iterated. In the following example, the List will be iterated, but the Table won't, meaning each iteration will work on the full Table.



Here is a more specific example:



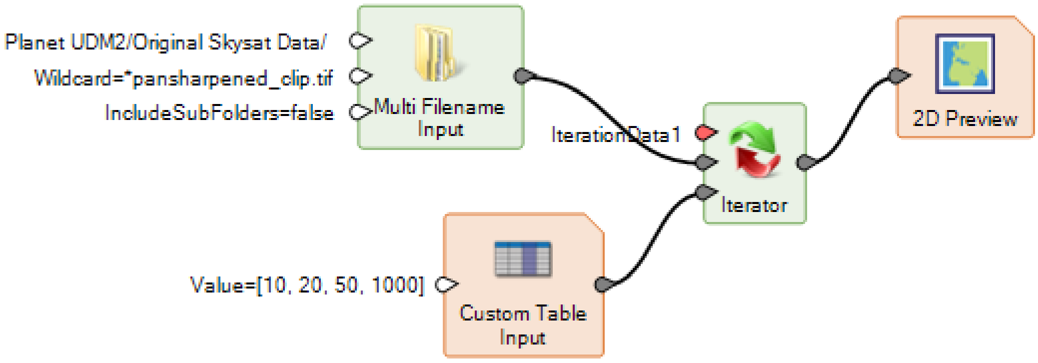
In the example shown in the screenshot above, the Multi Filename Input operator searches a specific directory for files which match a wildcard. This generates a list of six TIFF filenames, which are fed into the first port input of the Iterator. The TIFF images are 4-band RGBN images.

The Custom Table Input has a pre-defined set of four values which are intended to be used inside the Iterator to offset the corresponding band of each input image. So, band 1 should have 10 added to the DN values, band 2 should have 20 added, etc.

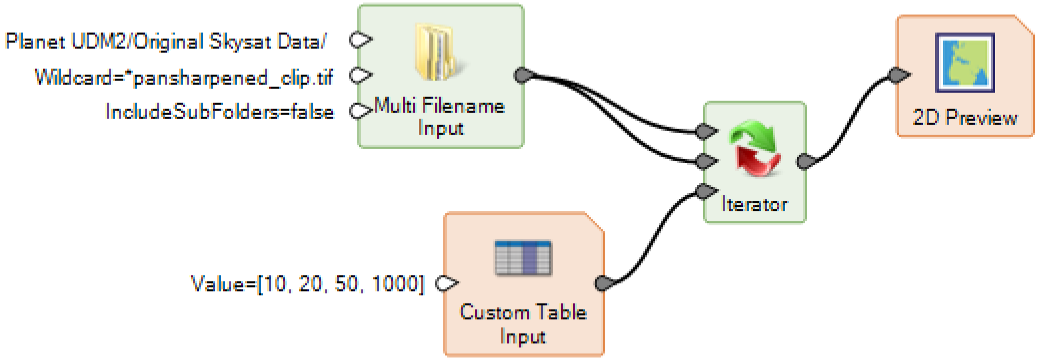
However, by default the model as shown above will not produce the expected / desired outputs. The Custom Table Input feeds the shorter list to the Iterator (four items, versus six for the Multi Filename Input) and so this list controls the iterations. Consequently, only four of the six input images are processed, and since each iteration only uses one value from the Custom Table Input's list, the first image has 10 added to all four of its bands, the second image has 20 added to all four of its bands, etc.

Instead, we can make use of the new IterationData<N> capability to control which of the input lists controls the iterations.

Right-click on the Iterator operator and select Add Port to add an IterationData1 port.

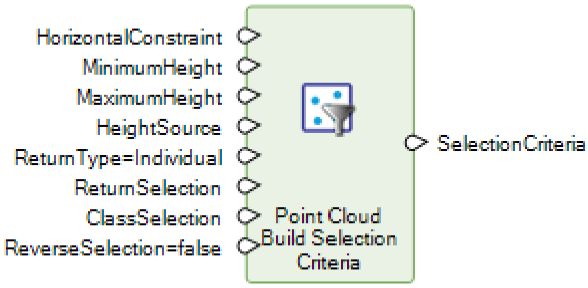


Since we want to iterate on the list of six TIFF files generated by the Multi Filename Input operator we make another connection from its output port to the IterationData1 port.



Now the model has been told to iterate using the list of six TIFF files. Consequently, for each of those raster images the sub-model inside the Iterator will add 10 to the DN values of band 1, 20 to band 2, 50 to band 3 and 1000 to band 4, as originally desired.

Point Cloud Build Selection Criteria

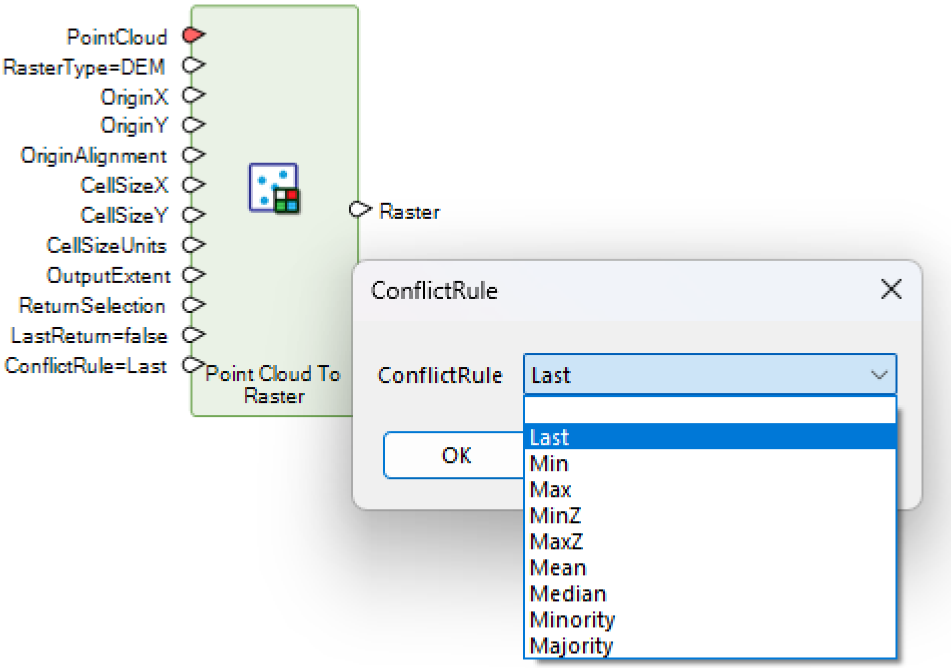


This operator enables you to define a set of selection criteria that can be used by other Point Cloud operators to define the points that they will operate upon. Several changes have bene made for Update 2:

You may define a height constraint for each feature on port **HorizontalConstraint** by providing an attribute name for **MaximumHeight** and/or **MinimumHeight**, in which case the attribute values are combined with each corresponding feature geometry to define the 3D spatial constraint.

The **HeightSource** port allows you to determine how point height values are determined. If **HeightSource** is an IMAGINE.Double, the port's value is subtracted from each point Z value. If **HeightSource** is IMAGINE.Raster or IMAGINE.PointCloud a surface is derived from the input and the height is determined by subtracting the surface elevation value from the point's Z value at the point's X, Y location. If the point lies outside the surface horizontal extent, height is set to (NaN). If **HeightSource** is IMAGINE.String the height value is taken from the named point attribute value, e.g., RelativeHeight. You may use operator Compute Relative Height to set the **RelativeHeight** attribute or it may be read from a PointCloud stored in a file whose format supports persisted custom attributes, e.g., LAS/LAZ/HSPC.

Point Cloud To Raster



A **ConflictRule** port has been added to avoid “random” results when converting a point cloud to a raster. Previously for each output pixel, the last input point encountered which falls inside that pixel extent would be used to determine the pixels DN value. Since order of processing can vary the "last" point could also vary, potentially resulting in differing results each time the model was run with the same inputs.

The new ConflictRule port provides various options beside Last, which produce consistent outputs.

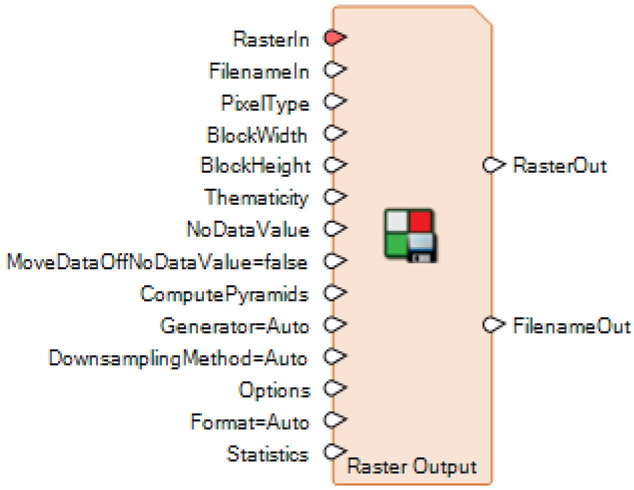
The Point Cloud To Raster operator produces one of three types of raster from point cloud input:

- DEM: A digital elevation model with one layer for each return specified at port **ReturnSelection**
- Multispectral: A multispectral image with layers for NIR, Red, Green, Blue if supported by the input point cloud, together with one layer of Intensity values for each return specified at port **ReturnSelection**
- Thematic: A thematic image with pixel values obtained from the classification attribute for each point

The ConflictRule port allows finer control over how the DN value is determined when multiple input points fall on an output pixel location. The following table describes how the rules apply for the three raster types:

Conflict Rule	**DEM (Z value)**	**Multispectral (Intensity, RGB, NIR)**	**Thematic (Classification)**
LastPoint	Obtained from the last processed point falling in the same raster cell. This can lead to different results every time the operator is executed	Obtained from the last processed point falling in the same raster cell. This can lead to different results every time the operator is executed.	Obtained from the last processed point falling in the same raster cell. This can lead to different results every time the operator is executed.
Min	The minimum Z value of all points falling in the same raster cell	Throws an error	The minimum value taken from all points falling in the same raster cell
Max	The maximum Z value of all points falling in the same raster cell	Throws an error.	The maximum value taken from all points falling in the same raster cell
MinZ	Same as Min.	Obtained from the point having the minimum Z value of all processed points falling in the same raster cell.	Obtained from the point having the minimum Z value of all processed points falling in the same raster cell.
MaxZ	Same as Max.	Obtained from the point having the maximum Z value of all processed points falling in the same raster cell.	Obtained from the point having the maximum Z value of all processed points falling in the same raster cell.
Mean	The mean Z value of all points falling in the same raster cell.	The mean values of each band value from all processed points falling in the same raster cell	Throws an error.
Median	The median Z value of all points falling in the same raster cell	Throws an error	The median value from all processed points falling in the same raster cell.
Minority	Throws an error.	Throws an error.	The classification value occurring least often in all of the points falling on the same cell. The median value is used If there is no clear minority.
Majority	Throws an error.	Throws an error.	The classification value occurring most often in all of the points falling in the same cell. The median value is used if there is no clear majority

Raster Output



A **MoveDataOffNoDataValue** port has been added.

NoData is a concept used to describe the areas of a raster image where there are no real (known) data values. For all raster images, this includes the area outside the extent of the raster dataset. Raster images are always organized as a rectangular array of pixels. A pixel is either a member of the original datasets or it is NoData. When pixels are stored in a file (or other mechanism such as a database) there are generally two common mechanisms for determining which pixel locations represent NoData locations.

Firstly, a DN value can be used. This is generally referred to as value-based NoData (depending on the data format it may also be referred to as NullValues, etc.) because one of the available Digital Numbers is used. For example, for Unsigned 8-bit imagery, the DN 0 may commonly be used to represent NoData locations. Obviously, this means that DN 0 cannot be used at data locations because it is reserved to denote NoData locations;

Consequently, the second, more modern, approach to defining NoData locations in an image is to use an additional mask layer, independent of the regular bands of the image. By using mask-based NoData none of the DN values of the image are "wasted". In the above example of Unsigned 8-bit imagery the full range of 0 to 255 DN values can be used at data locations. The mask-based NoData may take up slightly more disk space to store the locations of NoData, but due to the use of lossless compression techniques the overhead is generally minimal. If a raster format supports mask-based NoData it is recommended to use it rather than value-based NoData.

When an image is read into a spatial model (usually via Raster Input operators) any NoData locations identified by the parent file and its format are converted to mask-based NoData for use within the spatial model's processing (even if the source were value-based). Consequently, the data on the **RasterIn** port of the Raster Output operator will always have mask-based NoData (even if no locations are identified as being NoData) and is potentially using the full range of available DN values at data locations. By default, it is then down to the selected output file format (and associated Preferences and other settings) to determine if that mask-based NoData gets saved as a mask in the output file or is converted to another form such as value-based (or potentially lost entirely).

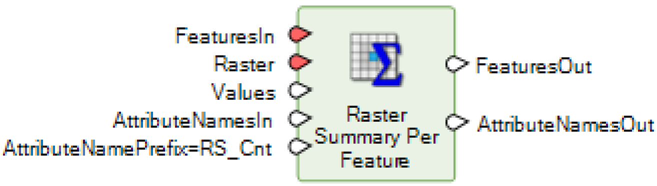
The **NoDataValue** port can be used to specify one of the available output DN values to be denoted as NoData locations. If the output format does not support mask-based NoData the pixels falling under the locations identified as NoData by the mask will be set to the selected **NoDataValue** DN and the header of the output file will denote that DN value as representing NoData locations. NOTE that this means that, by default, any data locations (i.e., pixel locations not marked as NoData by the NoData mask) with the specified **NoDataValue** will also be considered NoData locations in the output file. If manually defining a **NoDataValue** care should therefore be taken to ensure that a value is not used which could cause the accidental conversion of data locations to NoData locations in the output file.

One way of ensuring this is to use the **MoveDataOffNoDataValue** port. If this port is set to True then any data locations having the selected **NoDataValue** will be offset by 1 DN value. Any locations falling under the NoData mask's NoData locations will not be altered, even if they are the **NoDataValue**. For example, if the output is Unsigned 8-bit and 0 is selected as the **NoDataValue** and **MoveDataOffNoDataValue** is True, data locations of RasterIn with DN 0 will be changed to DN 1. Whereas NoData locations which had DN value 0 (or any other value) will be the **NoDataValue** in the output file. This way data locations are not unwittingly added to the NoData. But you may now have a lot more pixels with DN value 1.

Note that using the **MoveDataOffNoDataValue** port does require an additional pass through the data and will therefore add to the spatial model processing time.

A semi-related topic of concern to note is that not all software applications are adept at handling NoData settings and may simply assume that a value such as DN value 0 represent NoData, or transparent, locations, even if they are not specifically identified as NoData. In these instances, customers do not wish additional pixels to become zeros due to the processing that has been applied. For example, a customer might be processing an Unsigned 16-bit satellite image which initially has no pixels with DN 0. The customer is going to resample the image to a higher resolution using Cubic Convolution resampling. Due to the mathematics involved in Cubic Convolution resampling (as well as the rounding involved to produce an unsigned 16-bit output image file), it is entirely possible that some output pixels might be processed to an output value of 0. But the customer does not wish pixels of DN value 0 to be introduced. It would be possible to have the spatial model make an adjustment to DN values to achieve this, or the **MoveDataOffNoDataValue** port could be used (along with **NoDataValue)**. If **NoDataValue** were set to 0 and **MoveDataOffNoDataValue** were set to True, no output data pixels would have DN value 0.

Raster Summary Per Feature



In previous versions this operator had issues when writing out attribute fields to the Shapefile format (via a Features Output operator). Limitations in the field name length for Shapefiles caused names to be potentially re-used and therefore overwritten.

This has been corrected (or minimized) by defaulting to shorter attribute name prefixes.

New Spatial Modeler operators in ERDAS IMAGINE 2023 Update 2

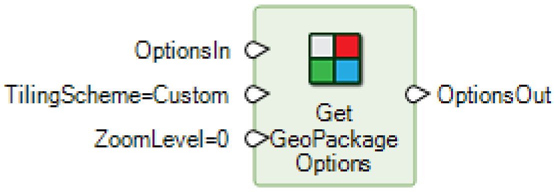
The following operators are new for the ERDAS IMAGINE 2023 Update 2 release.

Apply Image Styling

Apply the specified styling parameters to a Raster. The operator takes in an IMAGINE.Raster and an Image Chain Parameters (ICP) file. The latter is used to define the styling effects to be applied to the Raster, which produces a three-band unsigned 8-bit RGB Raster (or a single band if the ICP was for a Panchromatic Image Chain). Since the output utilizes the whole 8-bit range, it should be displayed using the no stretch option, so that it looks the same as the styled original image would have.

ICP files are generally saved automatically when displaying and styling raster data using Image Chains. The file may be stored in the user's $PERSONAL\ImageChainParams directory or, if the Per User option was turned off, in the same directory as the parent image file. You may also save an ICP file explicitly using the Save As option under the Settings | AutoSave Styling button on an Image Chain tab.

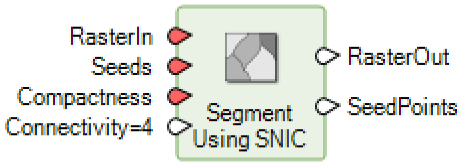
Get GeoPackage Options



This operator creates the format specific output option dictionary for use if the Raster Output operator it is connected to produces raster data in GeoPackage format.

Currently only TilingScheme and ZoomLevel are parameters which can be controlled.

Segment Using SNIC

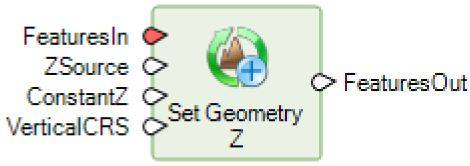


This operator performs segmentation on the input raster using the Simple Non-Iterative Clustering (SNIC) algorithm. Segmentation is a way of partitioning images into segments based on pixel values and locations. Pixels that are spatially connected and have similar values are grouped in a single segment.

The result is a thematic image where pixel values represent segment IDs. The input to the Seeds port can either be a positive number or a point feature stream. A number input specifies the desired number of seed points which will be generated programmatically such that they are evenly distributed across the input raster area. That is, the X-spacing and Y-spacing between adjacent seed points are the same or very close. For point feature stream input, the points which lie within the area of the input raster will be used as the seed points.

For the details of SNIC algorithm, refer to Achanta, Radhakrishna and Susstrunk, Sabine, “Superpixels and Polygons using Simple Non-Iterative Clustering,” CVPR, 2017.

Set Geometry Z



This operator sets the Z coordinate for all vertices in the primary geometry of FeaturesIn either from an attribute of the corresponding feature or from a supplied raster. An optional constant Z value can also be specified to provide a value for locations for which ZSource does not provide data.

Format Support in ERDAS IMAGINE 2023 Update 2:

CADRG

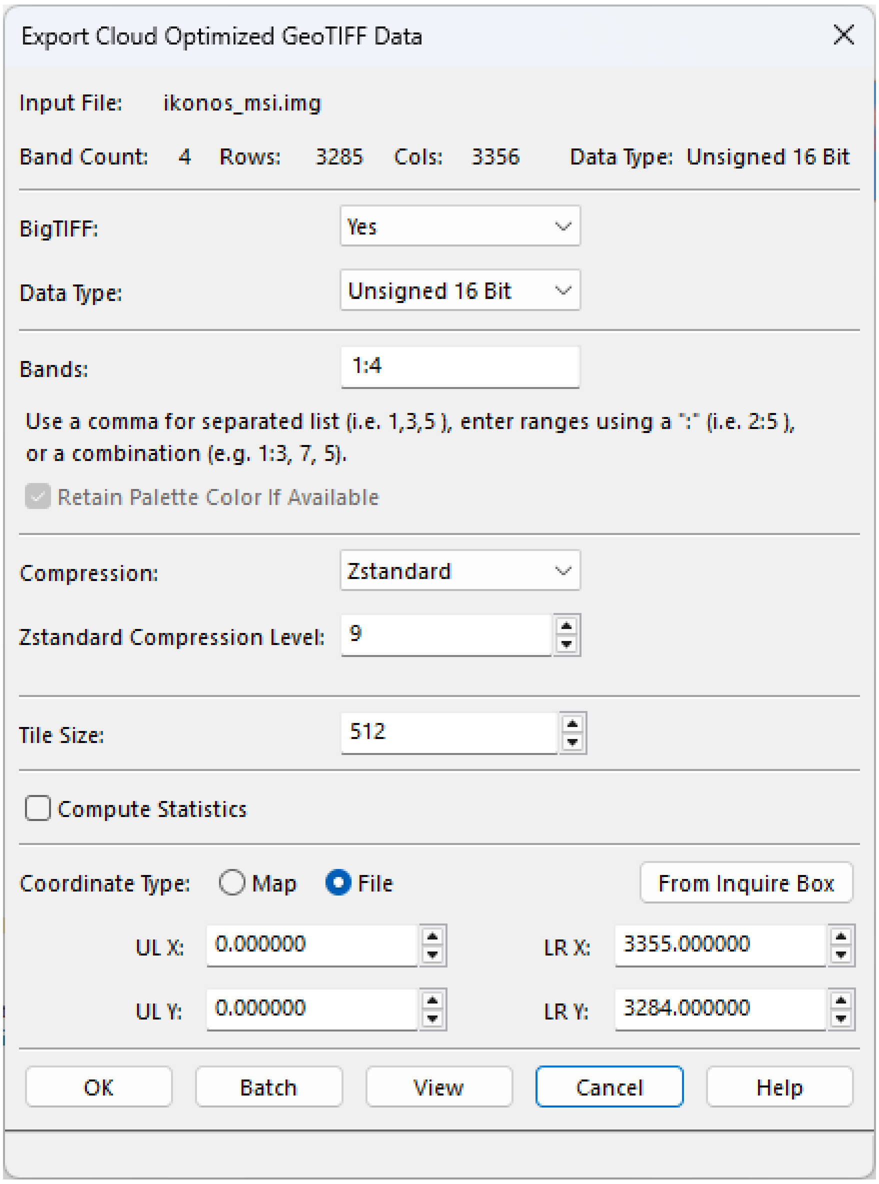
The display of CADRG has been improved, especially in the handling of “background” values, improving the rendition of the map colors.

An issue with the CADRG exporter has also been addressed where random input colors were sometimes incorrectly being turned to background.

Capella

Capella changed their SAR data formatting in September 2024 rendering some modes (such as Spotlight) inaccessible in ERDAS IMAGINE. The direct read capability has been updated to deal with the new formatting.

Cloud Optimized GeoTIFF Exporter



Based on the new options added to the Get TIFF Options operator (see above) a dedicated Cloud Optimized GeoTIFF (COG) Export dialog has been added.

EOS-04 CEOS

The EOS-04 SAR satellite (formerly known as RISAT-1A) is an Indian Space Research Organisation Radar Imaging Satellite designed to provide high-quality images under all weather conditions for applications such as agriculture, forestry and plantations, soil moisture and hydrology and flood mapping. It is a follow-on to the RISAT-1 satellite with similar configuration.

The data, in CEOS format, can now be directly read in ERDAS IMAGINE.

Hexagon Smart Point Cloud (HSPC) Archive format (*.hspc.pack)

HSPC is a point cloud format, processing and visualization framework from Hexagon.

It enables flexible workflows while reducing processing time. HSPC is based on a specialized coordinate encoding particularly suited to point collection by scanners. The resulting storage format has been optimized for data transfer, cloud-based storage, rapid spatial queries and fast visualization.

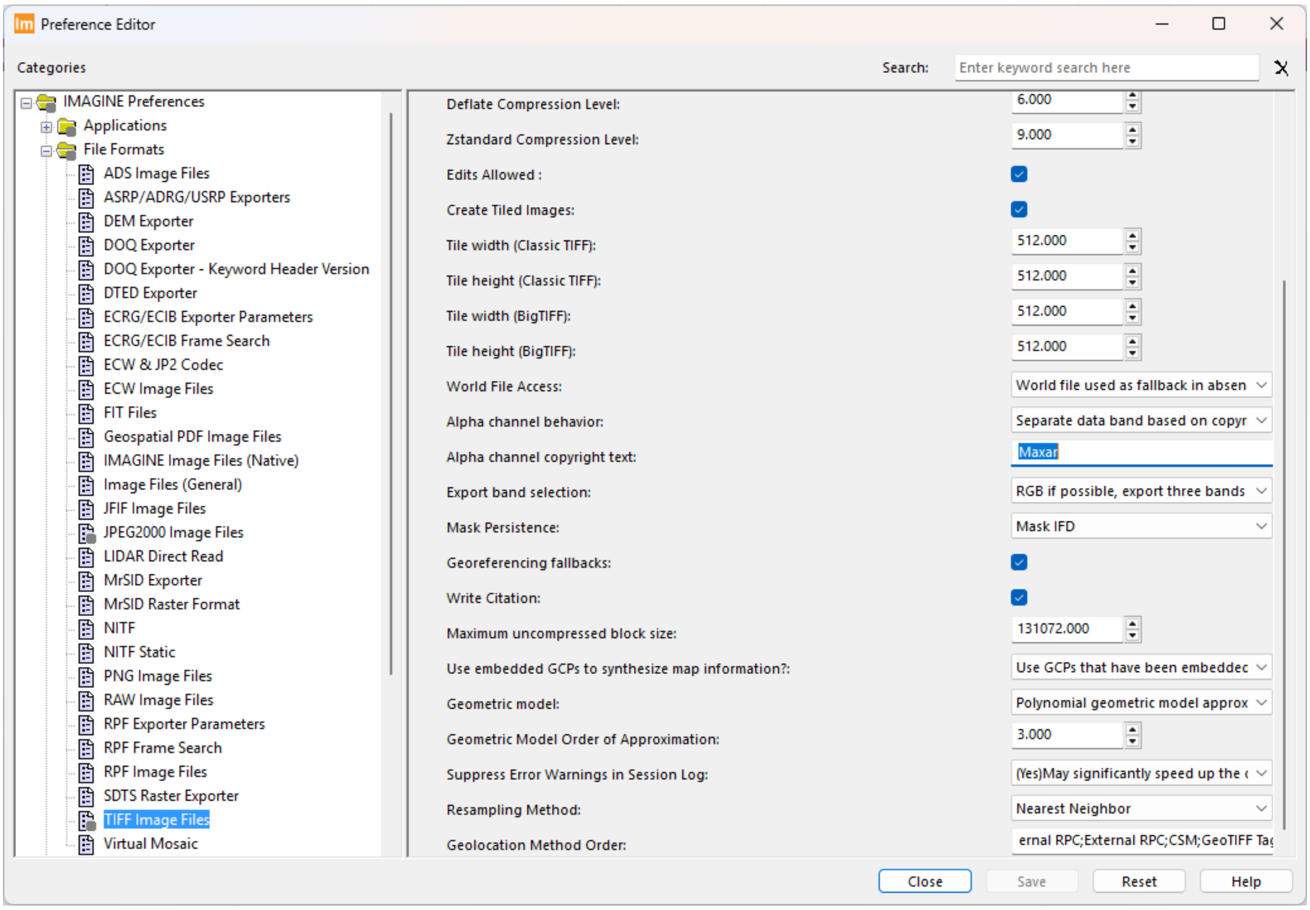
Regular HSPC datasets are currently stored as exploded directory structures, meaning many files. This made them impractical to move around.

Consequently, the HSPC Archive format was introduced which packs all the files into a single archive file that can easily be transferred to different computers and locations. This format can now be read natively as a point cloud in ERDAS IMAGINE, as well as directly written.

Maxar Legion

Support for the new Maxar Legion satellite constellation has been added, including rigorous and RPC sensor model support and the provision of Sensor Attribute Files (.saf) to help in the selection of appropriate band combinations when displaying the data or deriving Indices.

Note that some Maxar TIFF files (and other producers of TIFF) can occasionally have poor TIFF tag assignments in their headers. This is usually associated with four or more band TIFFs with a Photometric Interpretation defined, but the additional samples are identified as Alpha channel(s), not additional spectral bands. By default, ERDAS IMAGINE would read those additional samples as the specified alpha channels, resulting in unexpected display. A set of Preferences have been added to assist customers in bypassing the "correct" interpretation of the additional samples tag, if they have such data.



Stripped-TIFF Virtual Pyramids

TIFF imagery, which is physically organized into strips, rather than tiles, of raster information has previously been slow to display in ERDAS IMAGINE, especially when zooming out (e.g., performing a Fit to Frame operation to see the whole image extent at once) since it necessitates downsampling the pixel values and therefore reading of the source data. For data organised into strips this reading was inefficient. This could be alleviated by building pyramids for the image, but again this would take time to pre-generate.

Instead ERDAS IMAGINE now reads stripped TIFF data by treating it as if it has “virtual” pyramids, resulting in far faster downsampling of the data. A Fit to Frame can thus now be achieved in a couple of seconds without the need to generate real pyramids.

Note that there is a Preference controlling the use of Virtual Pyramids for stripped TIFF files if the customer wishes the software to behave the way it did previously.

THEOS-2 (Thailand Earth Observation System-2)

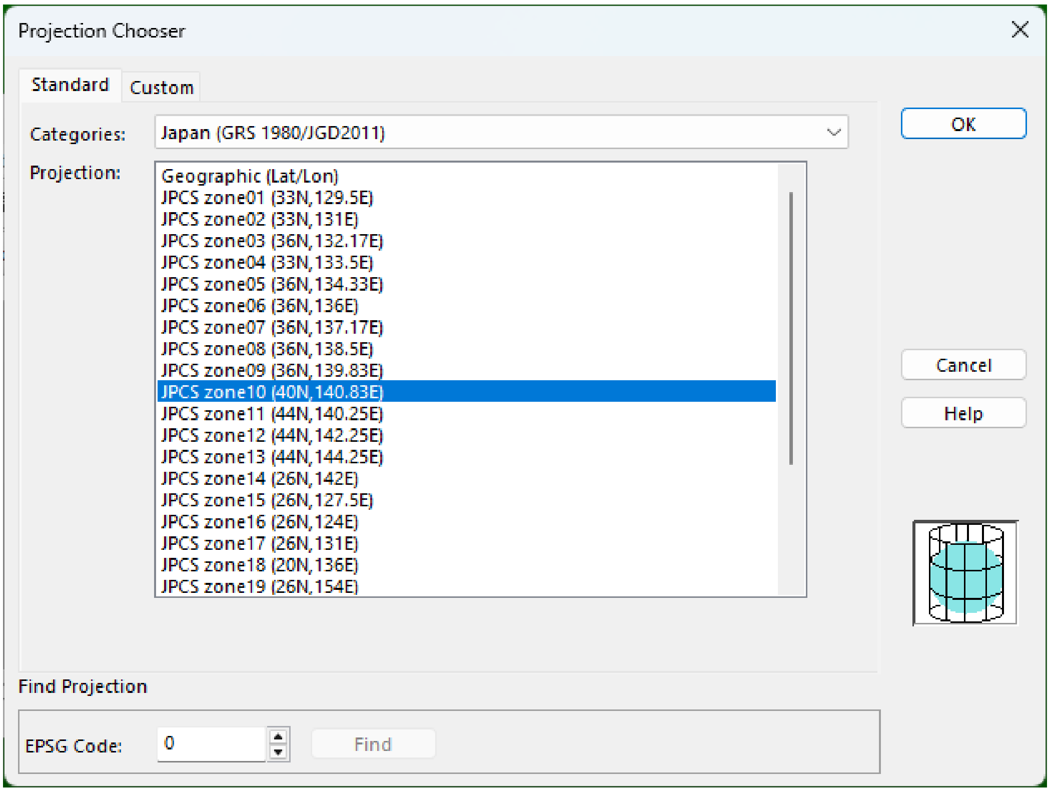
Launched on 9 October 2023 THEOS-2 is an optical satellite system delivering 0.5 m panchromatic and 2m multispectral (RGBN) ground resolution imagery, in DiMAP v2 format. This data can now be directly read in ERDAS IMAGINE, including support for RPCs on L2 products.

Projected Coordinate Systems (PCS)

The latest EPSG codes database (EPSG Version 11.015) has been imported.

Moroccan PCS definitions (including EPSG codes) have been added to the Standard tab of the Projection Chooser.

JGD2011-based projected coordinate systems for Japan have been added to a new category in the Projection Chooser’s Standard tab for ease of access:



General ERDAS IMAGINE 2023 Update 2

Water/Land Delineation using RADARSAT Constellation Mission (RCM)

When studying flood extents, especially while conditions in the region may still be affected by cloud and rain, SAR-based satellite imaging systems are invaluable. The RADARSAT Constellation Mission (RCM) satellites are one such example. The evolution of the RADARSAT program, the RCM includes a trio of Earth observation satellites, capable of scanning Earth, day or night, and in any weather conditions. The three-satellite configuration allows for daily revisits of Canada's vast territory and maritime approaches, as well as daily access to 90% of the world's surface and the Arctic up to four times a day.

To make use of this invaluable resource a dedicated tool has been added to use RCM imagery to quickly and easily identify and map the extent of water bodies in the scene. It does this through use of an AI model trained to understand the different characteristics between RCM data in water areas versus land areas.

As input it is recommended to use RCM Stripmap GRD data with HH & HV polarization in the spatial resolution range of 8 - 25m.

We recommend Stripmap-mode imagery for two reasons: Spatial and Radiometric resolution.

- ScanSAR imagery is spatially too coarse to produce a good product when mapping coastlines.
- And, secondly, the radiometric intensity of the HV band of ScanSAR imagery is so low that system noise is a major component. This greatly minimizes the value of using the HV band in ScanSAR data. The HV band is phenomenologically preferred for the Land / Water discrimination application and in Stripmap mode the HV polarization has a much better signal to noise ratio.

GRD (Ground Range geometry) format is recommended because in SLC, which is Slant Range geometry, features are distorted due to side-looking geometry. We want feature shape/structure to be independent of Look Angle.

An RCM image showing Vancouver and the eastern end of Vancouver Island:

The same RCM image processed through the RCM Land / Water Delineation tool to identify water bodies:

StereoSAR DEM replaced with StereoSAR DSM (IMAGINE Expansion Pack)

Sometimes it is not possible to obtain an interferometric pair of SAR images for highly accurate DEM extraction (or deformation monitoring). Whereas overlapping stereo pairs of SAR images are easier to obtain. So, for many years ERDAS IMAGINE has maintained a StereoSAR DEM application for extracting elevation information from such data even though the technique is less efficacious than using interferometry. However, the advent of Semi-Global Matching (SGM) techniques for stereo elevation extraction has provided an advancement to this approach, enabling switching to a new, streamlined StereoSAR DSM user interface.

Note that the legacy StereoSAR DEM interface is still provided as part of ERDAS IMAGINE 2023 Update 2 but will be removed from the next major release. This is to provide a transition period for customers to move from one to the other.

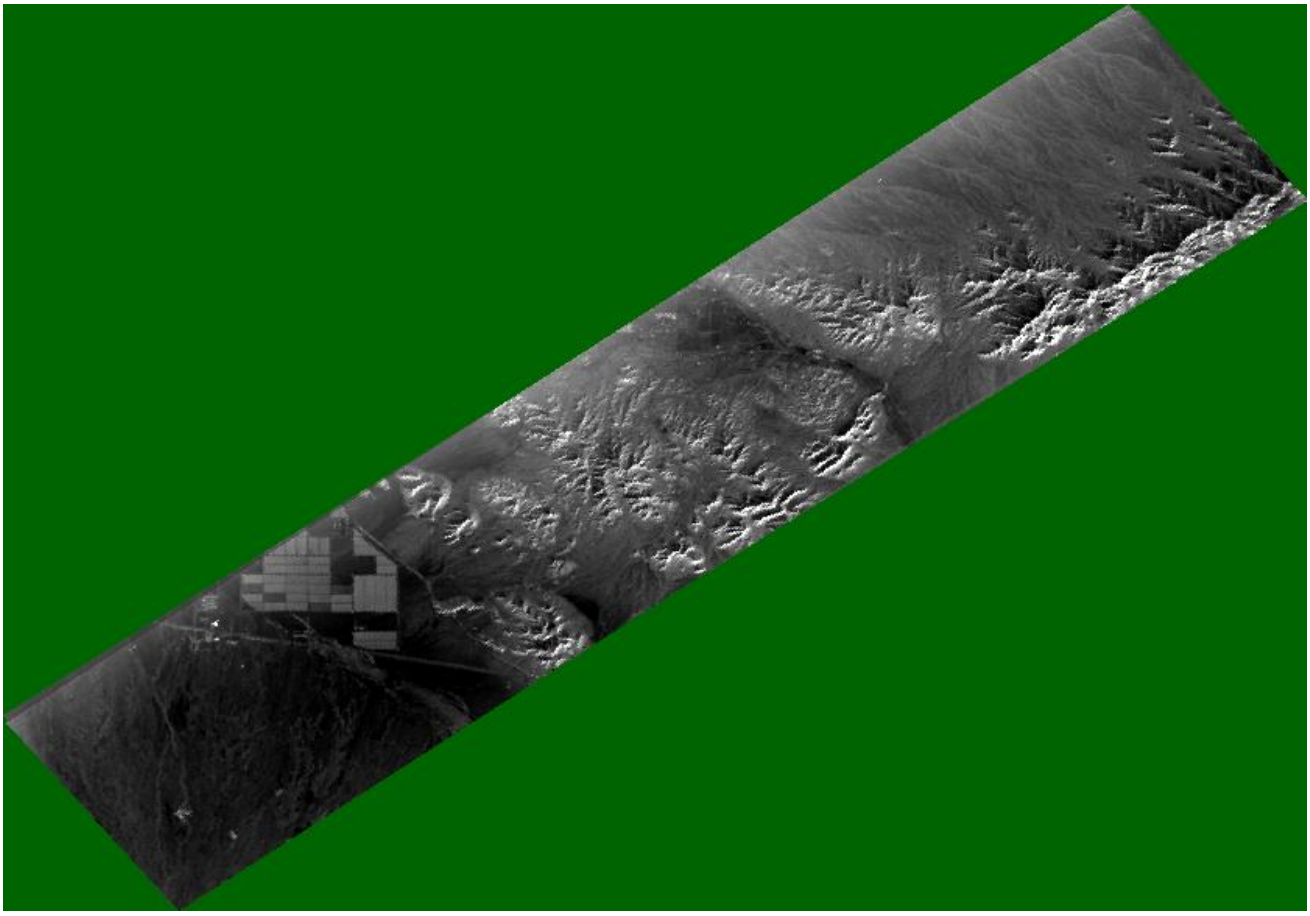
The ERDAS IMAGINE StereoSAR Digital Surface Model (DSM) software is designed to create a surface model from a radar stereo-pair. Note that this is a surface model as opposed to an elevation model. A surface model is calculated from the imaging return from the material that reflects the incident radiation, be it radar, optical or lidar. This reflective surface could be tree canopy, a building roof, road surfaces, ground cover or the Earth itself. This is, thus, different from a DEM which models the surface of the Earth i.e., ground elevation.

To create an accurate model requires that the SAR stereo-pair meet the geometric requirements of the software. These are discussed in detail in the Operational Radar Tour Guide but, in brief, these are the basic requirements:

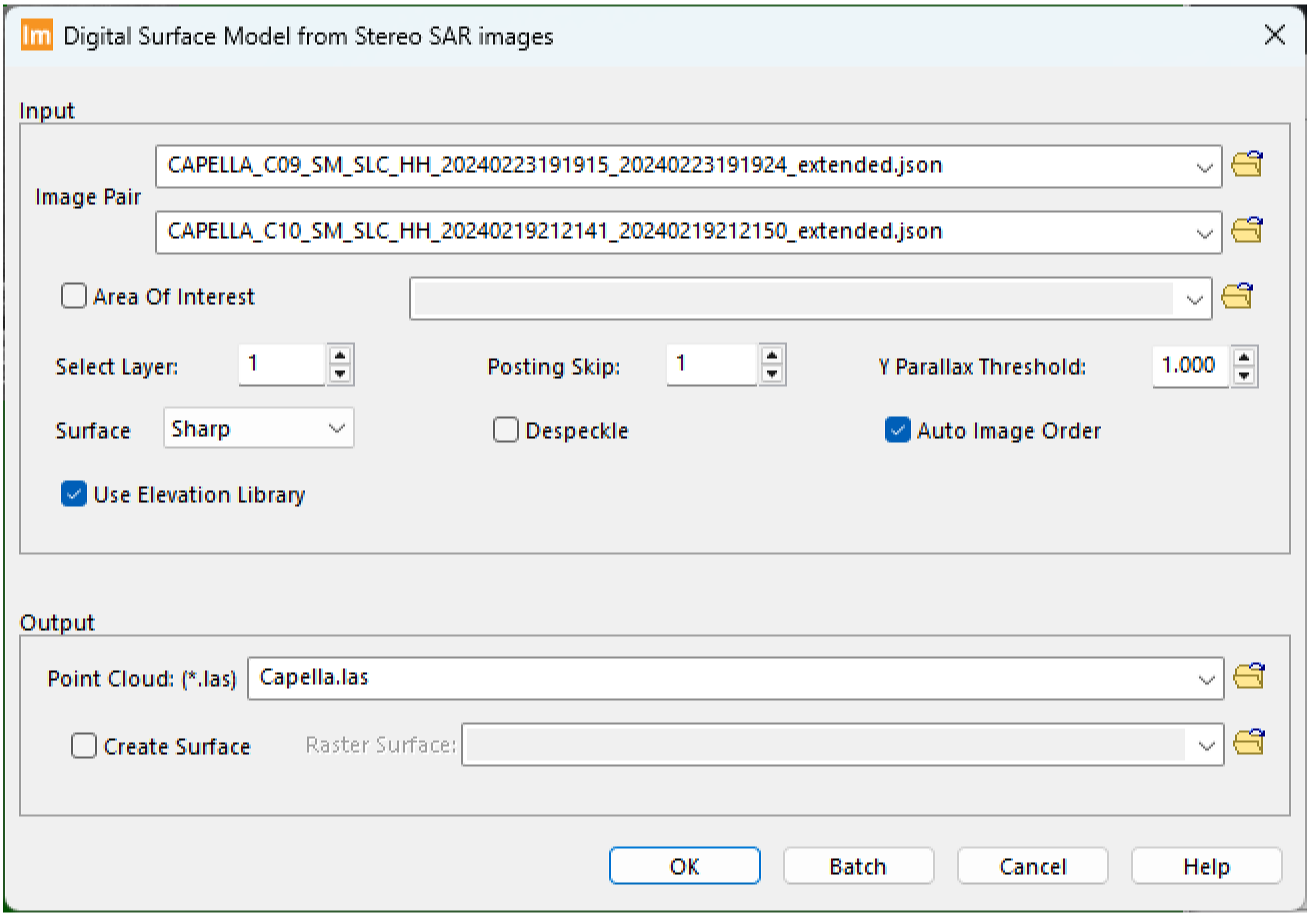
1. The incidence angle difference between the two images is best if between 8 and 12 degrees. As the difference gets larger or smaller the accuracy of the resultant DSM will slowly decrease. This is terrain dependent, with mountainous areas having tighter constraints than a smoother undulating terrain.
2. Larger incidence angles have less layover effect than smaller, more nadir, viewing angles. Thus a 50-60 degree stereo-pair would be less affected than a 20-30 degree pair. However, for mountainous areas, a larger incidence angle will result in a greater shadow problem. Resolving this may necessitate creating DSMs using opposite side viewing.
3. Both images of the stereo-pair must be acquired from the same orbit configuration: Ascending or Descending. A stereo-pair with opposite viewing directions will have insurmountable layover and shadow distortions.
4. In addition, the ground cover of the study area must be considered. The radar signal is impacted by changes in vegetation. Thus, depending on the nature of the ground cover, growing season, etc., scenes for a stereo-pair should be temporally close.

IMAGINE StereoSAR DSM software is designed to be automatic with minimal analyst requirements and inputs. The main requirement is that the scenes must conform to the constraints (viewing angles, etc.) discussed above.

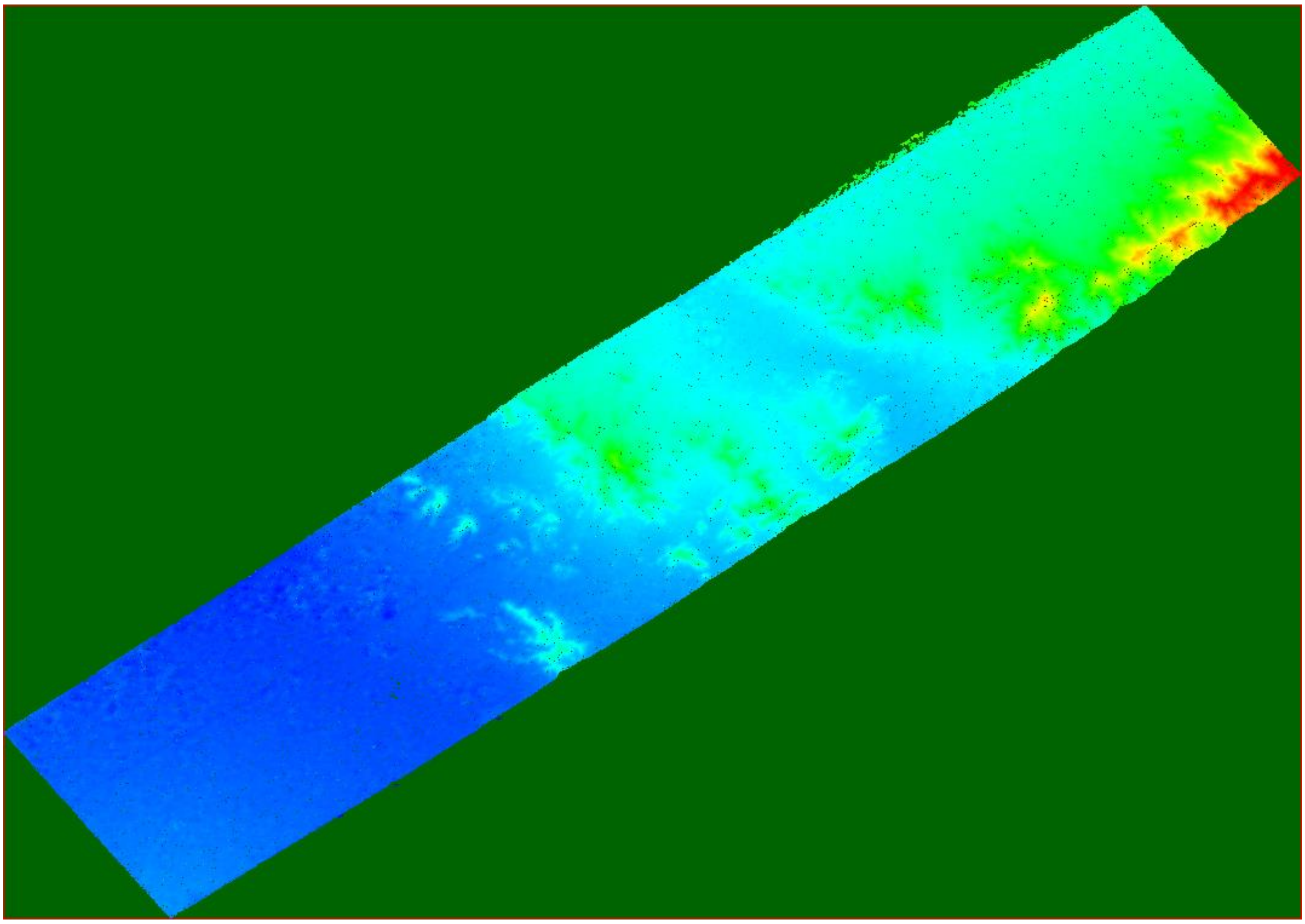
Two Capella SLC images with high overlap and incidence angles of 44.7 and 38.4 degrees:



The simplified, easy to use StereoSAR DSM interface:



The resulting DSM point cloud:



Normalized Difference Water Index (NDWI) for water bodies

For many years ERDAS IMAGINE has provided an extensive library of Indices for deriving different types of information from multispectral imagery, such as vegetation health, mineral locations, snow and ice extent, etc. One of these has been the Normalized Difference Water Index (NDWI).

This original NDWI was intended to assist with monitoring changes in water content of leaves, using near-infrared (NIR) and Mid infrared (MidIR) wavelengths, proposed by Gao in 1996. Applications include forest canopy stress analysis, leaf area index studies in densely foliated vegetation, plant productivity modelling, and fire susceptibility studies. As such the naming of this index caused confusion in that it was primarily intended as a vegetation health index, not one for identifying water bodies. The wavelength requirements (1,240 nm) also limited its use to sensors such as AVIRIS, MODIS and many hyperspectral sensors.

Consequently, the existing index has been renamed as the Normalized Difference Water Index for Vegetation and a new NDWI has been introduced. The new Normalized Difference Water Index for Water approach was first proposed by McFeeters in 1996 to detect surface waters in wetland environments and to allow for the measurement of surface water extent. Its wavelength requirements make it suitable for use with Sentinel-2, WorldView, most sensors with RGBN bands and many others.

H-Type USB Stealth Mouse

Support for the H-Type Stealth Mouse as a digitizing device is added for stereo viewers (Stereo Analyst, Stereo Point Measurement tool and Terrain Editor) providing you with an additional input device choice.

System requirements for ERDAS IMAGINE 2023 Update 2

ERDAS IMAGINE

Computer/processor	64-bit: Intel 64 (EM64T), AMD 64 or equivalent (four or more logical processors are strongly recommended) Support for the Advanced Vector Instruction Set (AVX) is required for the following capabilities: • Artificial Intelligence operators in Spatial Modeler (as indicated by their specific Help pages) • Hexagon Smart Point Cloud (HSPC) formats handling
Memory (RAM)	16 GB minimum, more is strongly recommended
Disk space	• 11 GB for software • 7 GB for optional example data • Data storage requirements vary by mapping project[1] • SSD drives strongly recommended for both software and data
Operating systems[2, 3, 4]	• Windows 10 Pro (64-bit, version 1607 or higher) • Windows 10 Enterprise (64-bit, version 1607 or higher) • Windows Server 2019 (64-bit) • Windows Server 2022 (64-bit) • Windows 11 Enterprise (64-bit) • Windows 11 Professional (64-bit)
Software	• OpenGL 4.2 or higher is required for Globe Views (this typically comes with supported graphics cards[5]) • Java Runtime 1.7.0.80 or higher — IMAGINE Objective requires JRE and can utilize any installed and configured JRE of version 1.7.0.80 or higher; Microsoft Visual C++ 2010 x64 Redistributable is also required. • Python 3.8.x through 3.12.x can optionally be used for writing Python scripts that reference ERDAS IMAGINE libraries or access operators that are available in the Spatial Modeler • Python 3.8.x through 3.12.x are optionally supported for executing Python scripts in the Spatial Model Editor • Microsoft DirectX 9c or higher • .NET Framework 4.7.2 or higher • OpenCL 1.2 with a device that supports double precision (cl_khr_fp64) if wanting to GPU accelerate NNDiffuse and other operators (most functions should fall back to the CPU if a suitable GPU is not present) • An NVIDIA card with CUDA Compute level of 3.5 or greater is recommended for use with Deep Learning and 5.0 or greater for other Spatial Modeler operators (most functions should fall back to the CPU if a suitable GPU is not present)
Recommended graphics cards for stereo display[6]	• NVIDIA Quadro P6000, P5000, P4000 and P2000 • NVIDIA Quadro M6000, M5000, M4000 and M2000 • NVIDIA Quadro K5200, K5000, K4200, K4000, K2200, K600 and K420 • NVIDIA Quadro RTX 4000 • NVIDIA Quadro RTX A4500 and RTX A5000
Recommended stereo display monitors	• 120 Hz (or above) LCD Monitors with NVIDIA 3D Vision Kit • 3D PluraView system from Schneider Digital[7] • Vision Engineering CONTOUR 3D stereoscopic GIS display

Recommended stereo glasses and emitter kits	• NVIDIA 3D Vision Kit • 3DTV Universal Emitter
Peripherals	All software installations: • One Windows-compatible mouse with scroll wheel or equivalent input device • Printing requires Windows-supported hardcopy devices[8] Software security (Hexagon Geospatial Licensing 2023) requires one of the following: • Ethernet card • One USB port for hardware key Advanced data collection requires one of the following hand controllers:[9] • TopoMouse or TopoMouse USB • Immersion 3D Mouse • MOUSE-TRAK • Stealth 3D (Immersion), S3D-E type, Serial Port • Stealth Z, S2-Z model, USB version • Stealth V, S3-V type (add as a serial device) • H-Type Stealth Mouse • 3Dconnexion SpaceMouse Pro[10] • 3Dconnexion SpaceExplorer mouse[10] • Z/I Mouse
Software interoperability	• ERDAS IMAGINE can be safely installed on a computer that has GeoMedia 2022 or GeoMedia 2023 installed; however, for greatest compatibility, it is highly recommended to install matching versions (including updates) • ERDAS IMAGINE 2023 requires GeoMedia 2023 for live linking (order of installation does not matter) • ERDAS IMAGINE 2023 Update 2 can read web services from an ERDAS APOLLO 2022 catalog, but not from ERDAS APOLLO 2023. Support for ERDAS APOLLO 2023 should be released in a later update. • ERDAS IMAGINE can interact with File Geodatabases (but ArcGIS installation is not required for File Geodatabase access)
Database engines	• PostgreSQL 13.2 with PostGIS 3.1.1: PostGIS can be used to store GeoMedia Features (.pfp) • Oracle Server 19c (12.2.0.3) 64-bit: Oracle Server 19c can be used to store Oracle Spatial Features (.ogv) (requires Oracle Spatial), as well as GeoMedia Features (.ofp) • Microsoft SQL Server 2019 64-bit: Microsoft SQL Server 2019 can be used to store GeoMedia Features (.sfp)
Internet connection	• ERDAS IMAGINE LiveLink for Google Earth Engine requires an active internet connection to operate. It has no offline mode.

ERDAS IMAGINE system requirements notes

[1] Disk I/O is usually the slowest task in geospatial data processing. Faster hard disks improve productivity. Reading data from one disk, writing temporary data to a second disk and writing data to a third disk improves performance. Disk arrays improve productivity, but some RAID options slow performance. Network disk drives are subject to network limitations.

[2] Server operating systems are not supported for IMAGINE Photogrammetry, ORIMA or ERDAS ER Mapper.

[3] The 3D stereo viewing and peripheral requirements of IMAGINE Photogrammetry limit its operating system options.

[4] Includes ERDAS ER Mapper support.

[5] Windows provides a generic OpenGL driver for all supported graphics cards; however, an OpenGL-optimized graphics card and driver are recommended for these applications.

[6] Graphics cards certified with previous versions of IMAGINE Photogrammetry and ORIMA may also be compatible but are not certified in the current version. Drivers must not be newer than R418. NVIDIA dropped 3D Vision support for drivers released after R418 U4 (425.31), which was released on April 11, 2019.

[7] Stereo monitors certified with previous versions of IMAGINE Photogrammetry and ORIMA may also be compatible but are not certified in the current version.

[8] HP-RTL drivers are recommended. Windows 64-bit print servers require 64-bit print drivers.

[9] Stealth S-Mouse (S2-S model) and MOUSE-TRAK are the only supported hand controllers in Stereo Analyst for ERDAS IMAGINE.

[10] 3Dconnexion mice are supported in IMAGINE Photogrammetry.

Issues resolved: ERDAS IMAGINE 2023 Update 2

IMAGINE Essentials

Support ticket	Summary
00246789	ERDAS IMAGINE failed to edit the datum of an existing ECW file
00242298	Top portion of raster image becomes NoData after calibration.
00265758 00364686	The Breakpoint Editor in ERDAS IMAGINE 2023 Update 1 is not saving changes to the lookup table (LUT) in the same way it used to in IMAGINE 2022.
00262175	EPSG 6257 has to be added properly to ERDAS IMAGINE
00250157	Affine resample doesn’t work in ERDAS IMAGINE 2023 Update 1
00209053	ERDAS IMAGINE sometimes crashes when opening certain .laz files, or it's very slow to display
00337490	Pulldown menus do not work in ERDAS IMAGINE 2022 for custom application created with the IMAGINE Developers Toolkit 2016
00358147	Polygons (with composite geometry holes) are missing from a Geodatabase
00303215	WMS data does not display by default (because of scale-based display rules)

IMAGINE Advantage

Support ticket	Summary
00335437	Not getting correct information when exporting DTED : Partial Cell Indicator
00310505	SAR Metadata node not transferred when importing data
00263448	mosaicprocesspro.exe does not read list of input images from file
00241659	Thematic data loses color table after ortho resampling.
00283718	Terrain Prep Tool outputs empty DEM when input is 3D ASCII XYZ points and Shapefile breaklines

IMAGINE Expansion Pack

Support ticket	Summary
00373709	AutoSync APM does not start correctly when input and reference images use pyramids with differing numbers of levels.
00140256	StereoSAR DEM’s Coregistration tool displays input control points in the wrong location

IMAGINE Professional

Support ticket	Summary
00022783	Threshold Distance Histogram GUI does not update the Chi-Square cells from the Spectral Angle Mapper Classification.

IMAGINE Objective

Support ticket	Summary
00208999	IMAGINE Objective - can't collect training samples from Shapefile

Spatial Modeler

Support ticket	Summary
00256373	Raster generated with the Point Cloud To Raster operator has a strip of area that has a salt and pepper appearance
00319244	Preview from Apply AOI causes error “Invalid level specified from Warp”
00226259	Spatial Model using the ‘Image Segmentation FLS’ operator runs in ERDAS IMAGINE 2022, but fails in 2023

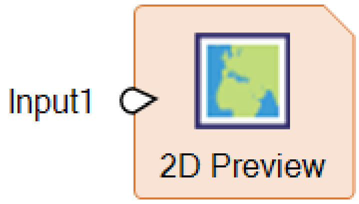
New technology for ERDAS IMAGINE 2023 Update 1

Spatial Modeler

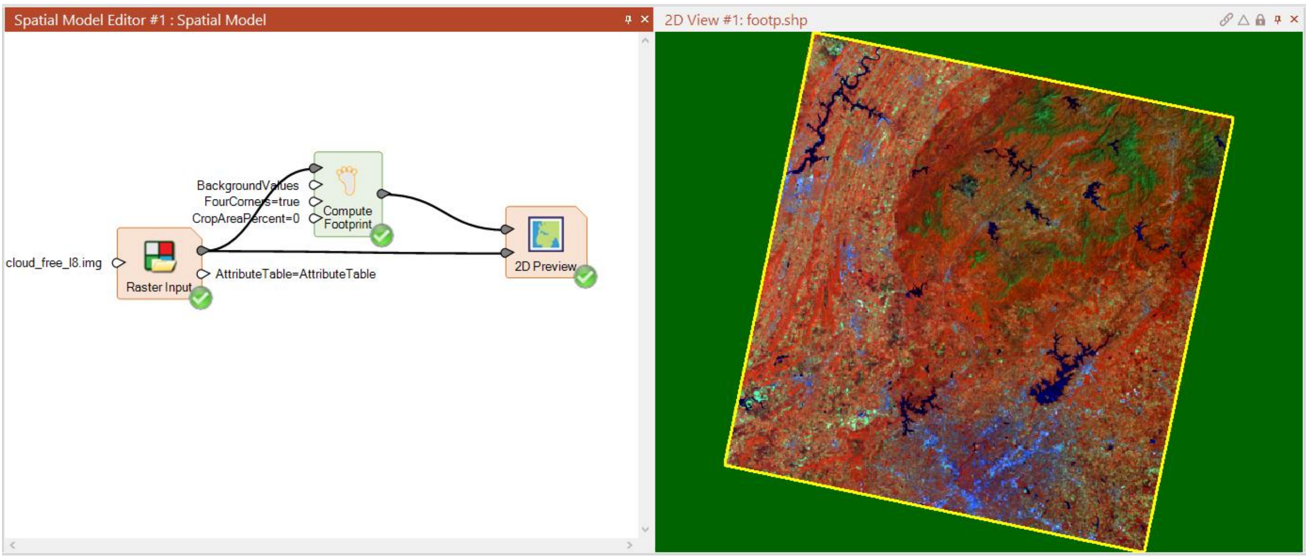
Update 1 extends the capabilities offered by Spatial Modeler including new and improved operators, tabular previews for tables and matrices and support for Python 3.12.0.

Updated Spatial Modeler operators in ERDAS IMAGINE 2023 Update 1

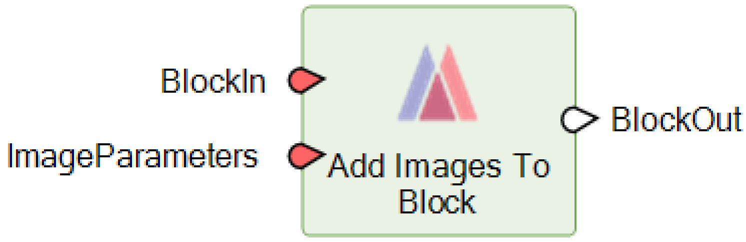
2D Preview



2D Preview now supports IMAGINE.Geometry directly, such as the output from Compute Footprint:



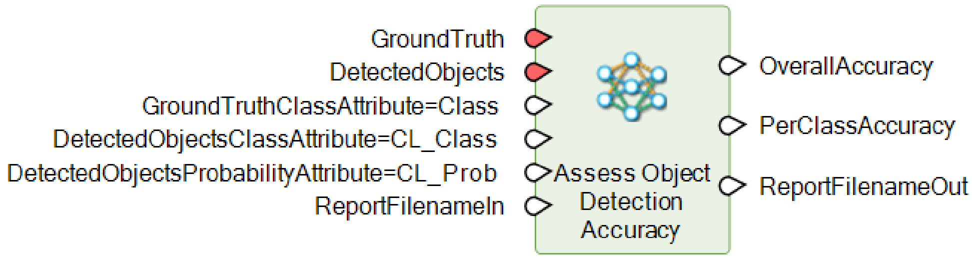
Add Images To Block



This operator adds images to a block. The images must be of the same sensor model types as those for which the block was created. As part of this process, interior orientation of the images is solved. The status of the exterior orientation of the images is also set according to the value specified in the parameter dictionary.

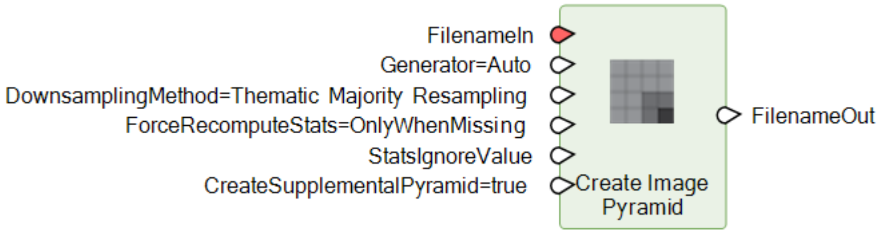
In previous versions this operator would fail if one (or more) of the input images was inappropriate for adding to the block. In this release this behavior has been changed so that if an inappropriate image is encountered a warning is issued, the image is skipped, and the process continues. This prevents the overall Spatial Model from failing unexpectedly.

Assess Object Detection Accuracy



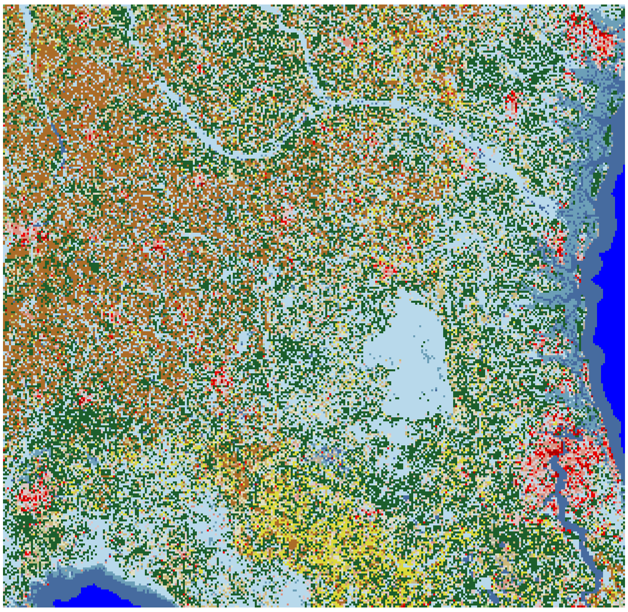
The Assess Object Detection Accuracy operator has been modified to take a list of features (since the Detect Objects Using Deep Learning operator now can generate a list of features and Assess Object Detection Accuracy needs to work through the list).

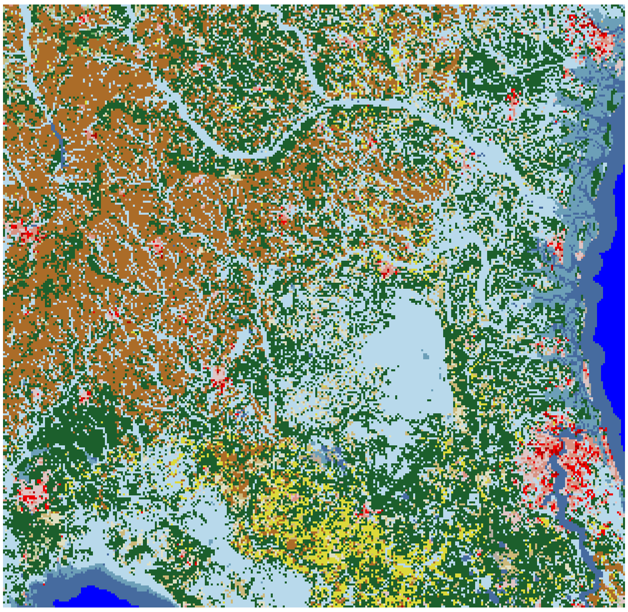
Create Image Pyramid



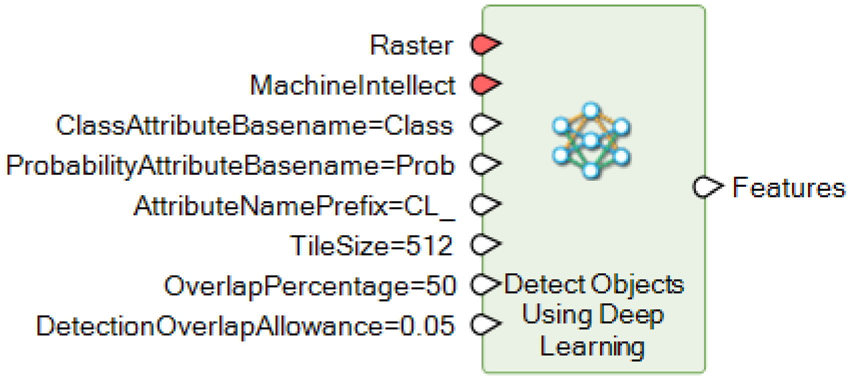
The Create Image Pyramid operator has been enhanced to provide a Thematic Majority Resampling option as an alternative to Nearest Neighbor (or Max Pixel Decimation) when creating pyramids for thematic raster data. This same option is available wherever pyramids can be created, such as the View/Edit Image Metadata dialog.

In specific cases the Thematic Majority Resampling option can result in smoother looking data when zoomed out. Below is a screenshot showing two copies of the same landcover image (13473 x 12129 pixels) displayed fit to frame with Nearest Neighbor resampled pyramids on the left and the new Thematic Majority Resampling on the right:



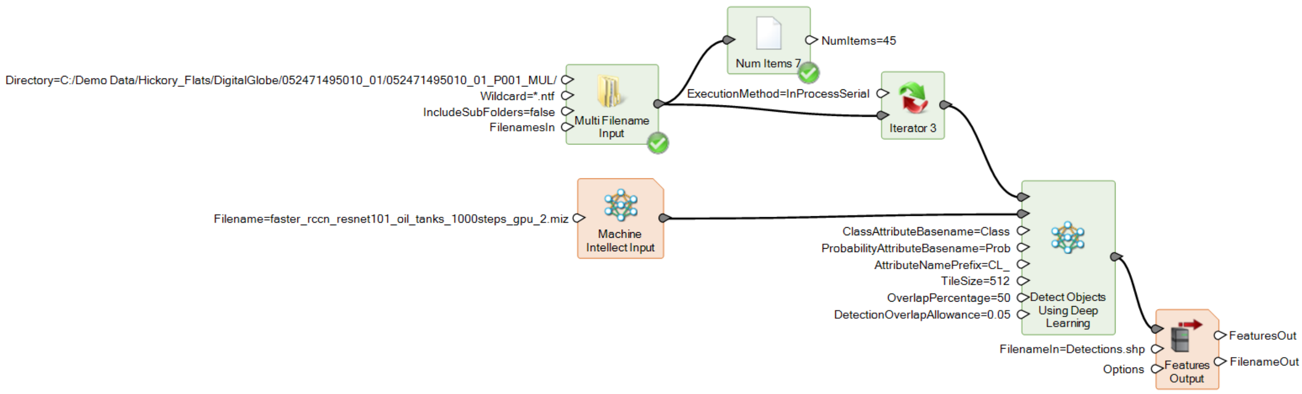


Detect Objects Using Deep Learning

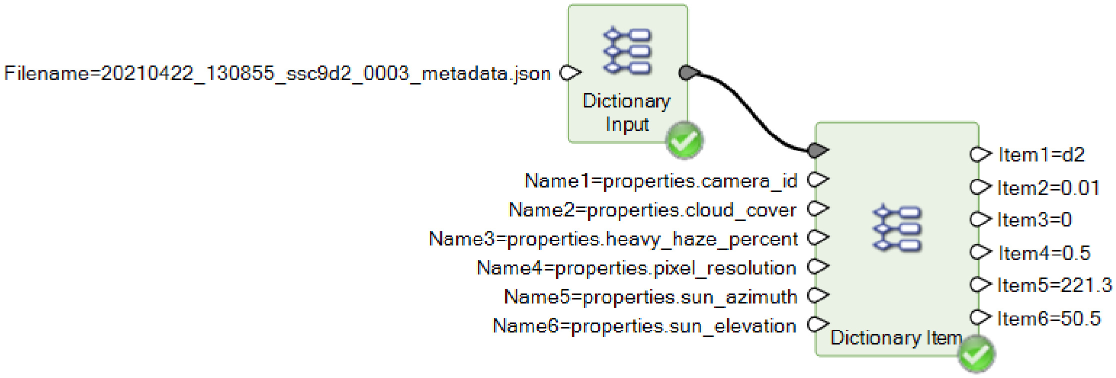


Detect Objects Using Deep Learning will now accept a list of rasters for input.

For example, the Spatial Model below reads the file names of all NITF images in a source directory, passes the list of file names to an Iterator which reads each one to a list of rasters and feeds that list of rasters to the Detect Objects Using Deep Learning operator in order to identify trained objects in each input image.

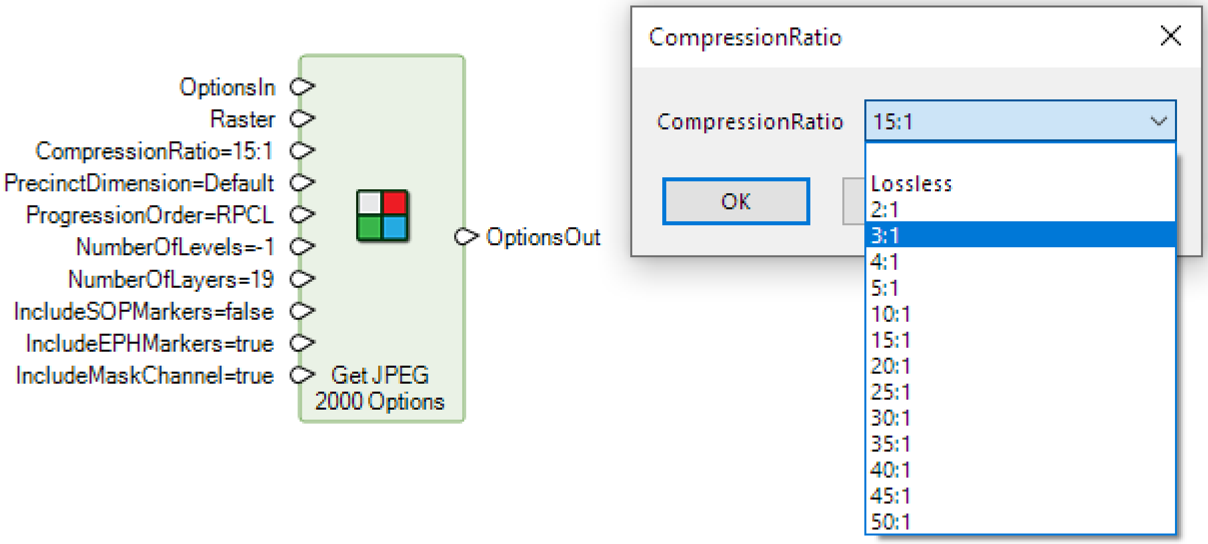


Dictionary Input



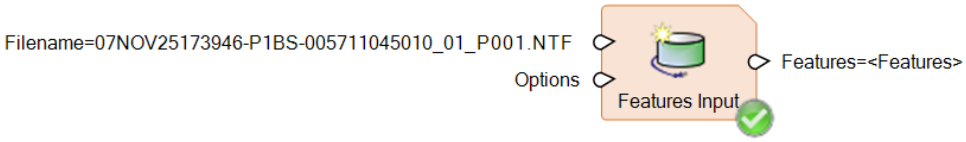
The Dictionary Input operator now reads JSON input files, which are sometimes supplied with satellite imagery to list metadata that can be useful in processing the data.

Get JPEG 2000 Options



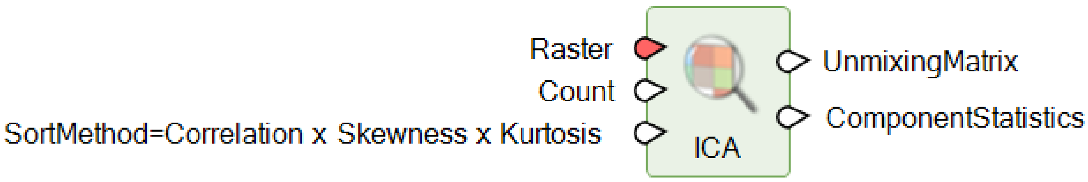
The Get JPEG 2000 Options operator now supports a wider range of pre-supplied options on the **CompressionRatio** port, as well as the ability to type arbitrary ratios via the Properties panel. This makes it possible to apply visually lossless compression ratios.

Features Input

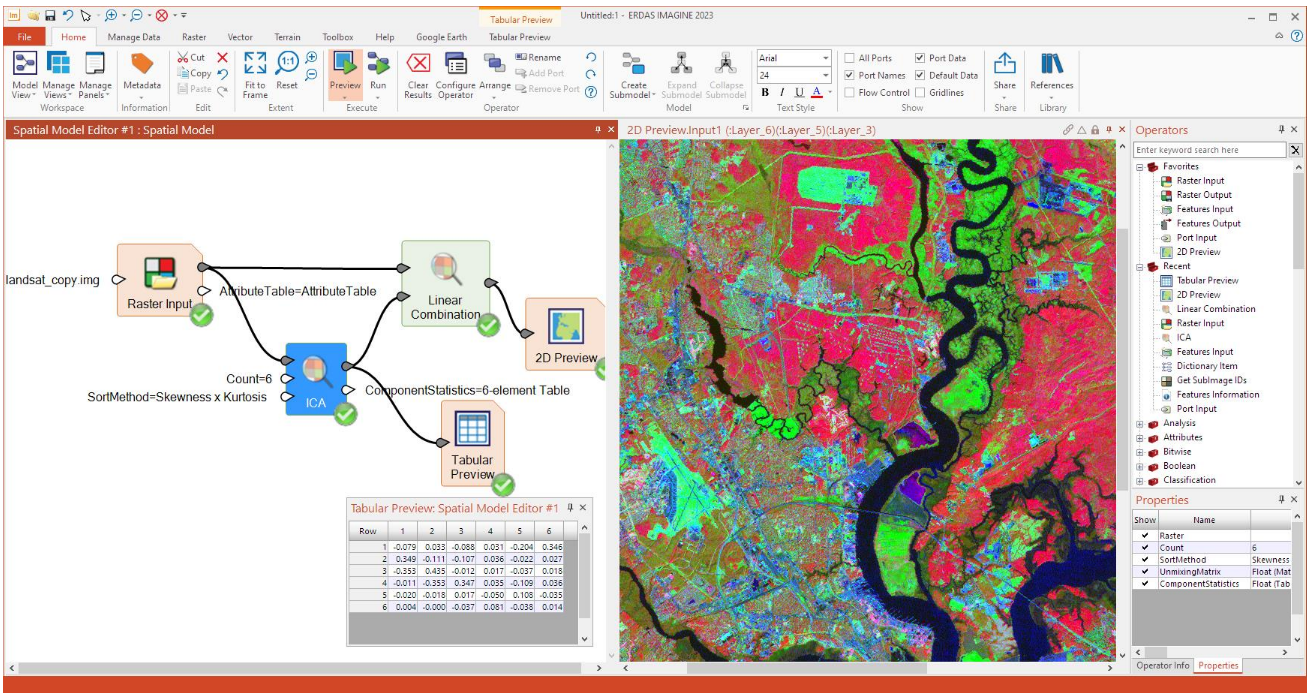


Features Input will now extract features embedded in NITF files, such as the image footprint, cloud locations, etc.

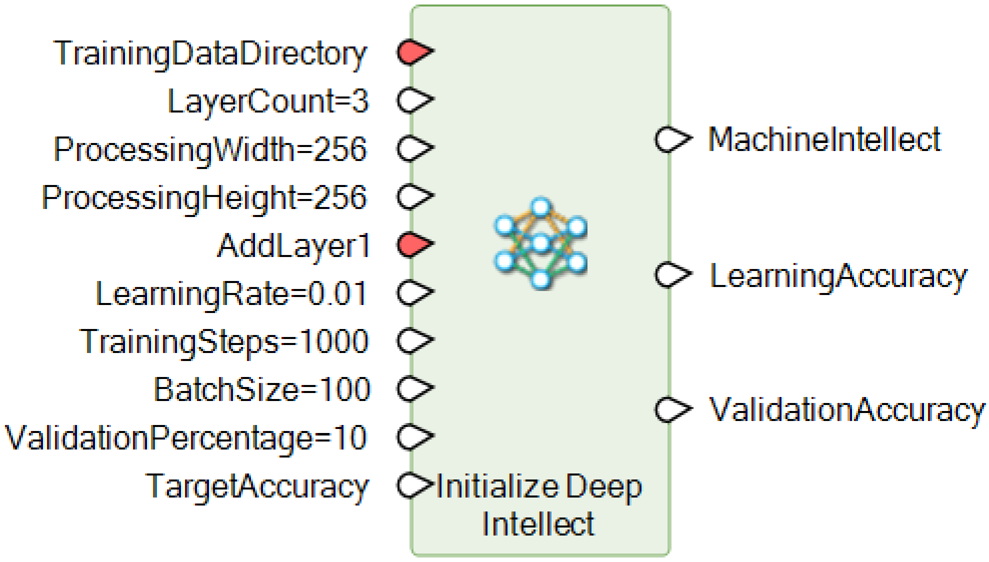
ICA



The Independent Component Analysis (ICA) operator has been overhauled to work with larger (multigigabyte) input images.

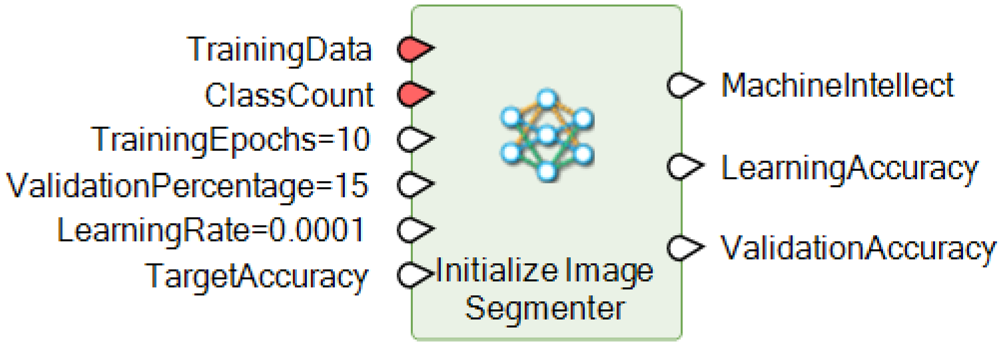


Initialize Deep Intellect



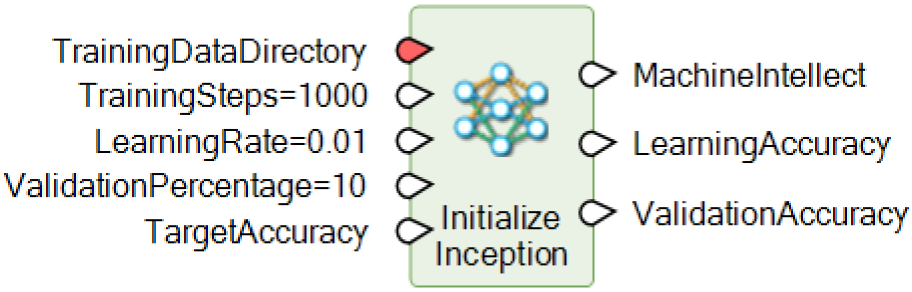
Initialize Deep Intellect now has a user-definable accuracy threshold at which to cease iterations.

Initialize Image Segmenter



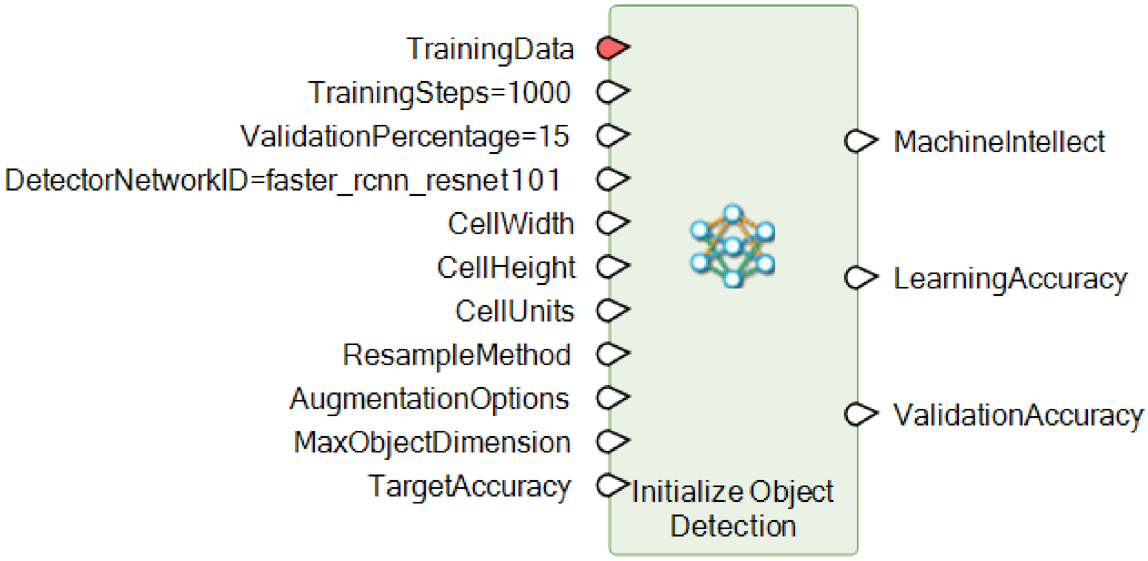
Initialize Image Segmenter now has a user-definable accuracy threshold at which to cease iterations.

Initialize Inception



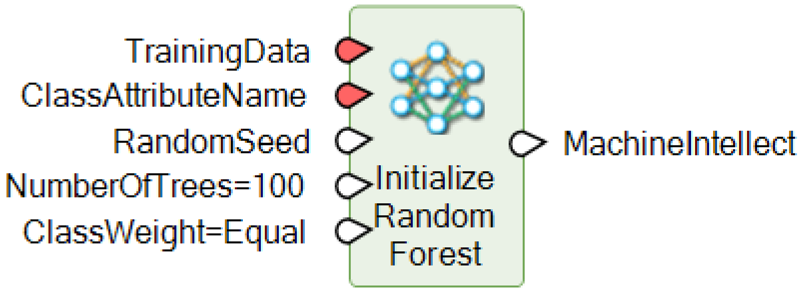
Initialize Inception now has a user-definable accuracy threshold at which to cease iterations.

Initialize Object Detection



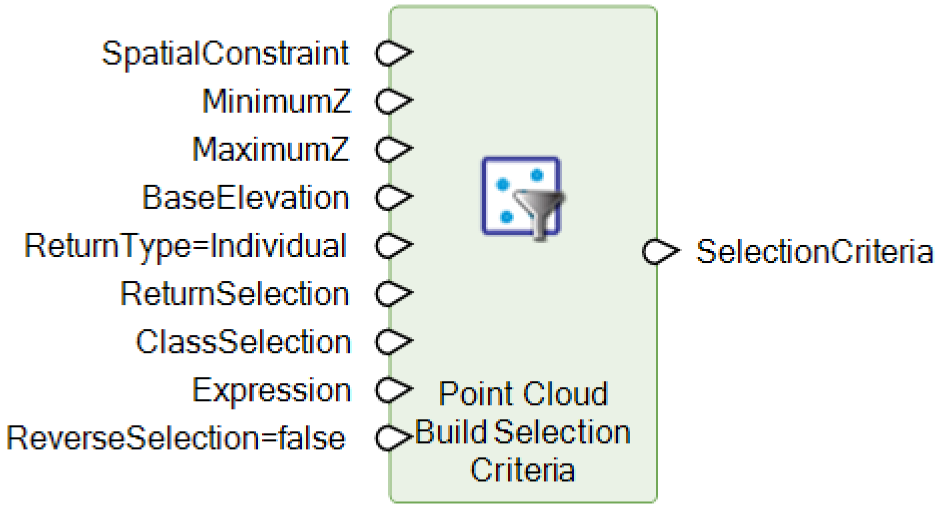
Initialize Object Detection now has a user-definable accuracy threshold at which to cease iterations.

Initialize Random Forest



Initialize Random Forest has been extended to specify the number of trees and set the class weights, enabling tuning of model robustness.

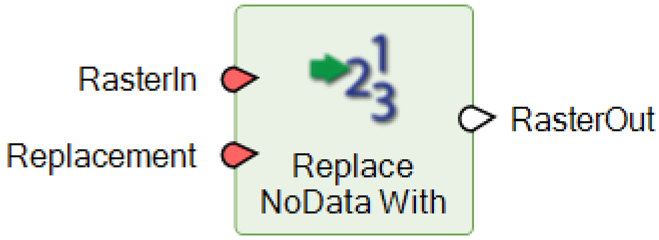
Point Cloud Build Selection Criteria



In order to simplify the process of setting **MinimumZ**/**MaximumZ** ranges specific to each **SpatialConstraint** geometry, the Point Cloud Build Selection Criteria operator has been enhanced.

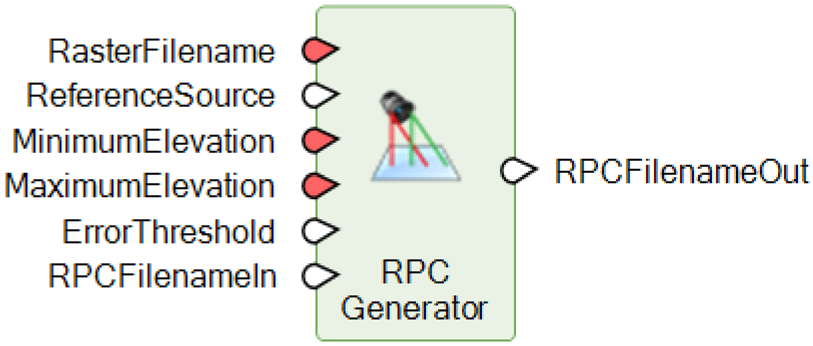
You may now define an elevation constraint for each feature on port **SpatialConstraint** by providing an IMAGINE.List of IMAGINE.Double on port **MaximumZ** and/or **MinimumZ**, where the number of values in the list is the same as the number of features. If there are more values than features, the excess values are ignored. If there are fewer values than features, only the spatial constraint is applied. Alternatively, you may provide an attribute field name for **MaximumZ** and/or **MinimumZ**, in which case the attribute values are combined with each corresponding feature geometry to define the 3D spatial constraint.

Replace NoData With



Replace NoData With no longer attempts to fill beyond the image's extent with the replacement value(s), i.e., outside the image remains NoData. This provides for more consistent processing when multiple datasets are being processed that have differing geospatial extents.

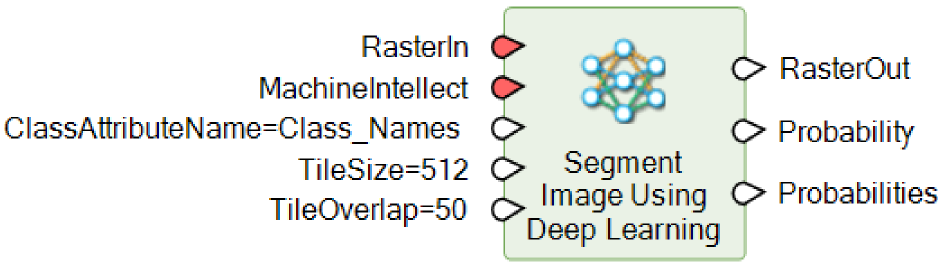
RPC Generator



The RPC Generator operator now accepts a block as input on the **ReferenceSource** port.

Previously, when using the RPC Generator operator to create an RPC file to go alongside an image file the image file had to already be associated with a 3D geometric model. This made it difficult to build a suitable Spatial Model if the model itself was going to build the 3D geometric model (i.e., if it did not already exist). Now that information can be stored in a block as part of the Spatial Model and that block is referenced for the 3D geometric model information.

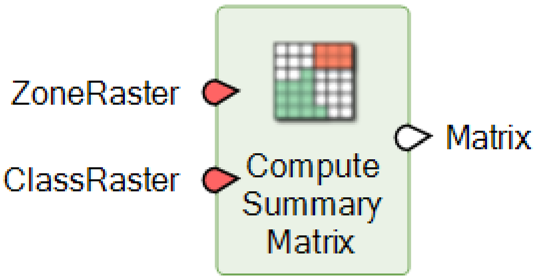
Segment Image Using Deep Learning



Segment Image Using Deep Learning will now accept a list of rasters for input.

New Spatial Modeler operators in ERDAS IMAGINE 2023 Update 1

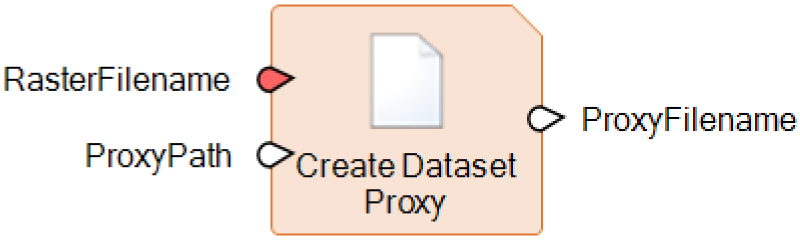
Compute Summary Matrix



Compute Summary Matrix returns a matrix containing a cross tabulation of the two input rasters. In the returned matrix, row-column position [i, j] contains the number of pixels that have value i in **ZoneRaster** and value j in **ClassRaster**. The values in **ZoneRaster** are referred to as zones and the values in **ClassRaster** as classes.

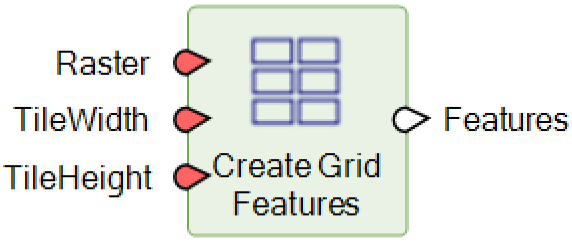
The **Matrix** output by this operator is intended as input to the Zonal Summary <Statistic> operators in order to compute the <Statistic> of the classes of **ClassRaster** in the zones of **ZoneRaster**.

Create Dataset Proxy



Use Create Dataset Proxy to create an Unrestricted Access Image (UAI) file for a raster dataset in a specified directory, optionally with a specified name. The primary use of a UAI file is to provide writeable access to a raster dataset that is stored in a read-only location or is not itself writeable for some other reason. If the File Formats → Image Files (General) → Access Read-only Image Datasets through Writable Proxy Datasets preference is turned on, UAI files are created automatically in a default location when attempting to perform an edit operation on a read-only dataset. Use this operator if you want control over when or where a UAI file is created and used.

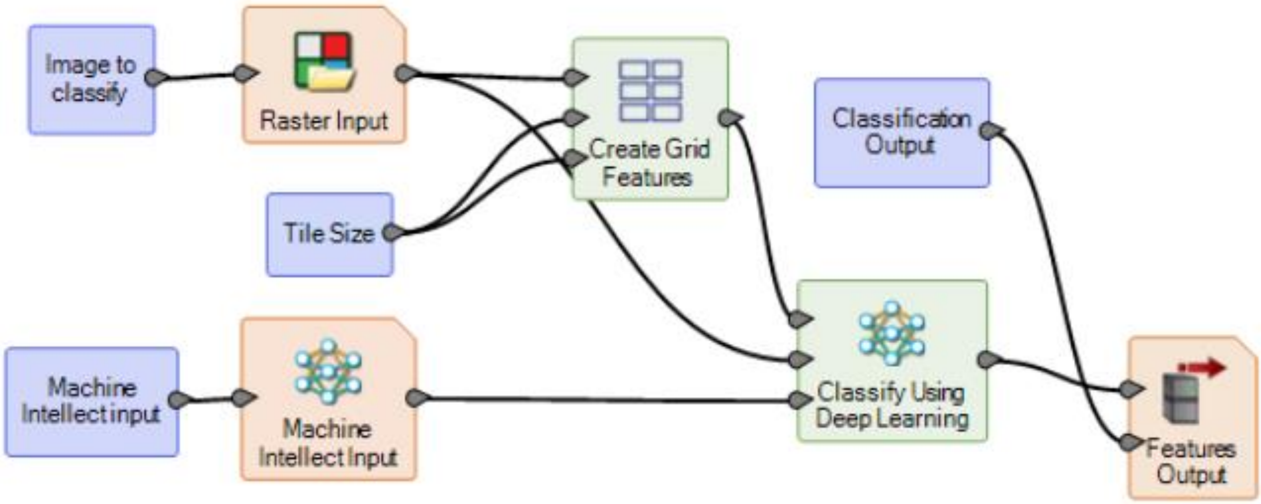
Create Grid Features



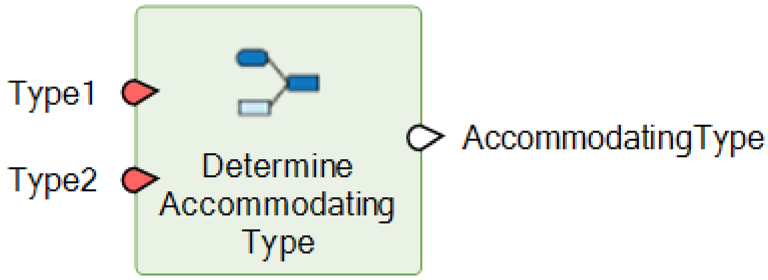
This operator generates a grid of rectangular feature tiles over the extent of the input raster stream.

The output features stream will contain ceil(raster width/**TileWidth**) x ceil(raster height/ **TileHeight**) area geometries. The geometries will butt join. The features on the right and bottom edges of the grid may extend beyond the extent of the input raster.

Below is an example of using the Create Grid Features operator to create the tiling structure required for classifying an image using deep learning.



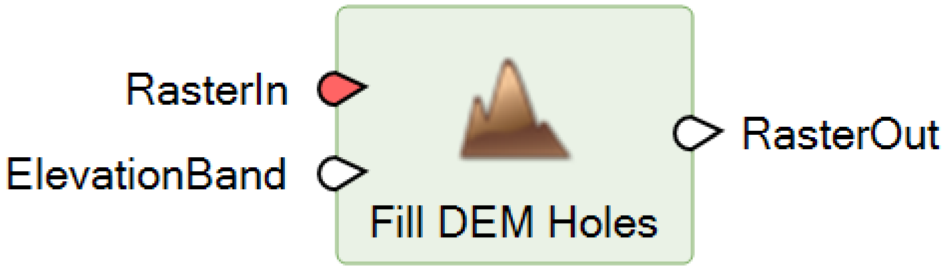
Determine Accommodating Type



Determine the minimum NumberType that will accommodate all possible values of all NumberTypes in the list.

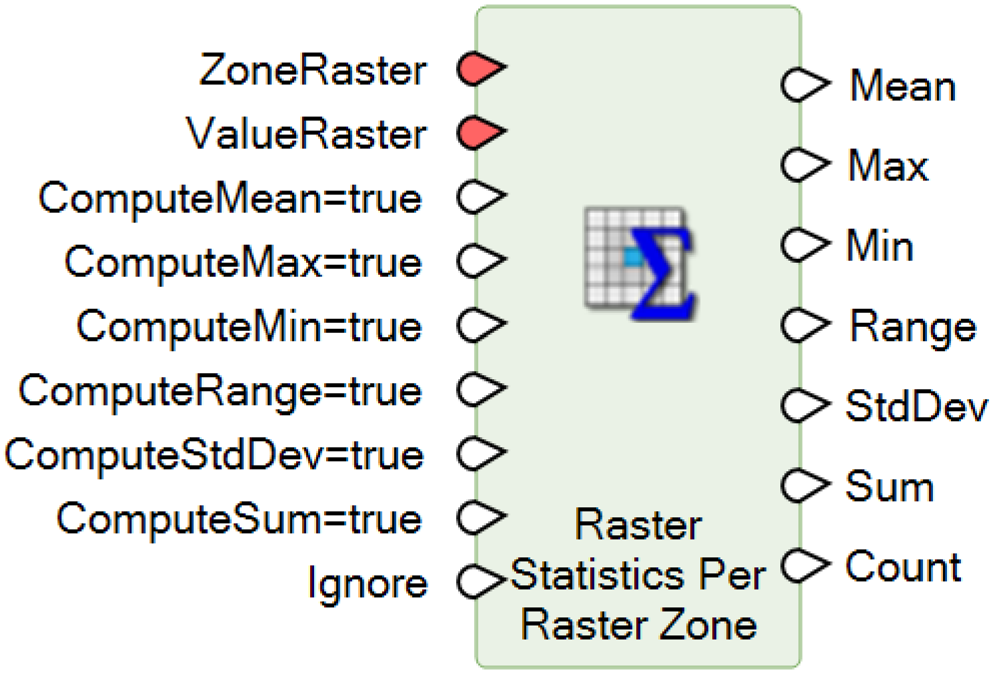
For example, if **Type1** contains "u4" and **Type2** contains "u8," "u8" is set as the **AccommodatingType** because all possible unsigned 4-bit values and all possible unsigned 8-bit values can be represented in an unsigned 8-bit value. However, if **Type1** contains "u8" and **Type2** contains "s8," "s16" is set as the **AccommodatingType**, because all possible unsigned 8-bit values and all possible signed 8-bit values cannot be represented by either an unsigned 8-bit value or a signed 8-bit value; it requires a signed 16-bit value to be able to represent the full range of both signed and unsigned 8-bit values.

Fill DEM Holes



The operator identifies NoData pixels or areas, otherwise known as holes, in the input raster. It uses TIN interpolation between pixels that border the NoData areas, to determine the DN values used to update NoData pixels. Only NoData pixels within the convex hull of the TIN are modified. All other pixel DN values are passed through unchanged.

Raster Statistics Per Raster Zone

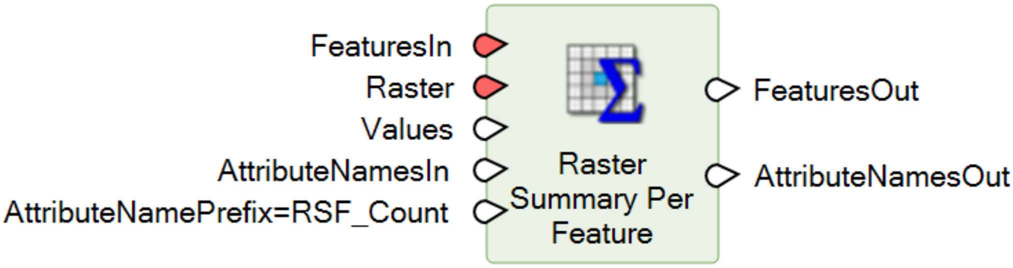


This operator computes selected statistical measures of all pixel DN values from **ValueRaster** that are overlaid by each zone (i.e., DN value) of **ZoneRaster**. Row i of the returned Table contains the statistical measure from **ValueRaster** of all pixels that have value i in **ZoneRaster**. By default, all supported statistics are computed and output.

If required, the output Tables can be attached to the **ZoneRaster** raster stream as attribute fields by using the Create Column(s) and Attach Attributes operators.

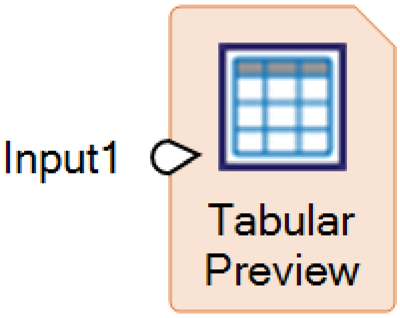
This operator provides a more efficient alternative to using multiple Zonal <Statistic> operators when multiple statistical measures are required.

Raster Summary Per Feature



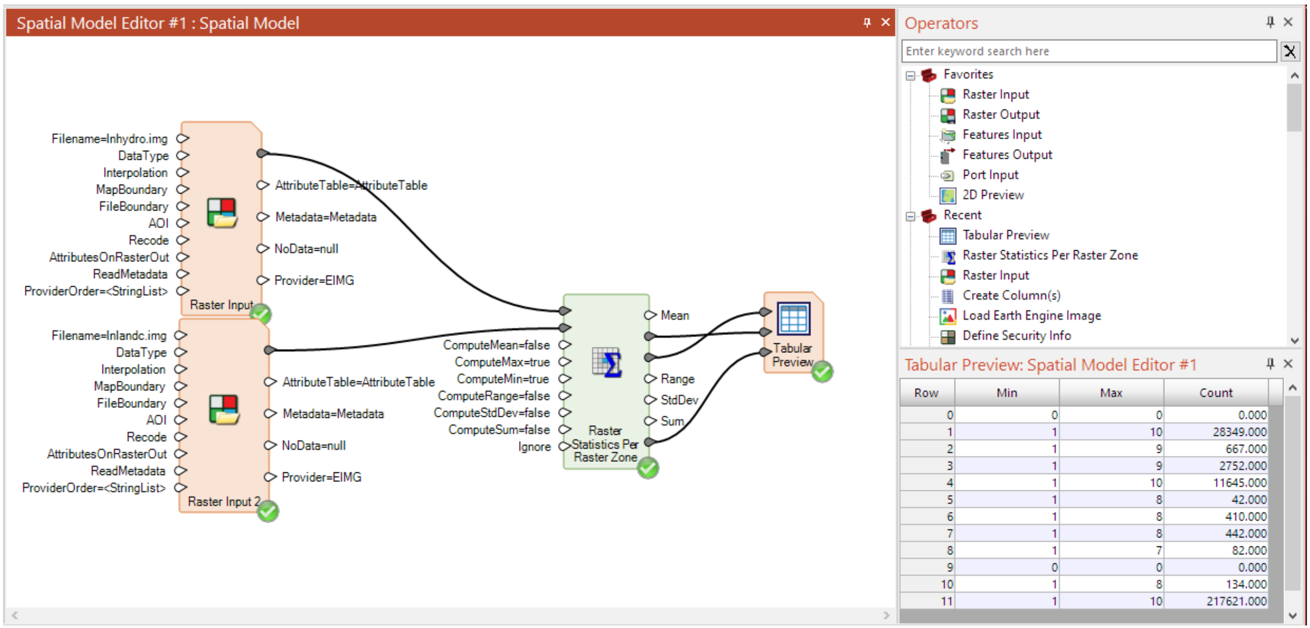
This operator computes the count of each DN value of the pixels of the input raster that are contained within a feature geometry (using the Primary Geometry field). The computed counts are added as attributes to the feature stream, one per value discovered in the input raster. The attributes are named by combining the value held by port **AttributeNamePrefix** and "<N>" where N is the pixel value. For example, by default, the name of the first column is "RS_Cnt1."

Tabular Preview



View AttributeTable, Table, Matrix or Scalar data on the fly in a tabular view. If multiple Tables or multiple Scalars are attached to the Tabular Preview operator, each Table or Scalar is represented as a column in the Tabular Preview pane. If an AttributeTable is being previewed, the Tabular Preview pane shows all columns for a single layer. You can choose which layer to preview on the Tabular Preview ribbon. Only a single AttributeTable or single Matrix can be previewed with each Tabular Preview operator. AttributeTables, Tables, Matrices and Scalars cannot be combined. If any Table or Table in an AttributeTable has a Bin Function, all Tables must have the same Bin Function.

The example below shows how this operator could be used to display a tabular view of three Tables output from Raster Statistics Per Raster Zone:



Spatial Model Editor improvements in ERDAS IMAGINE 2023 Update 1

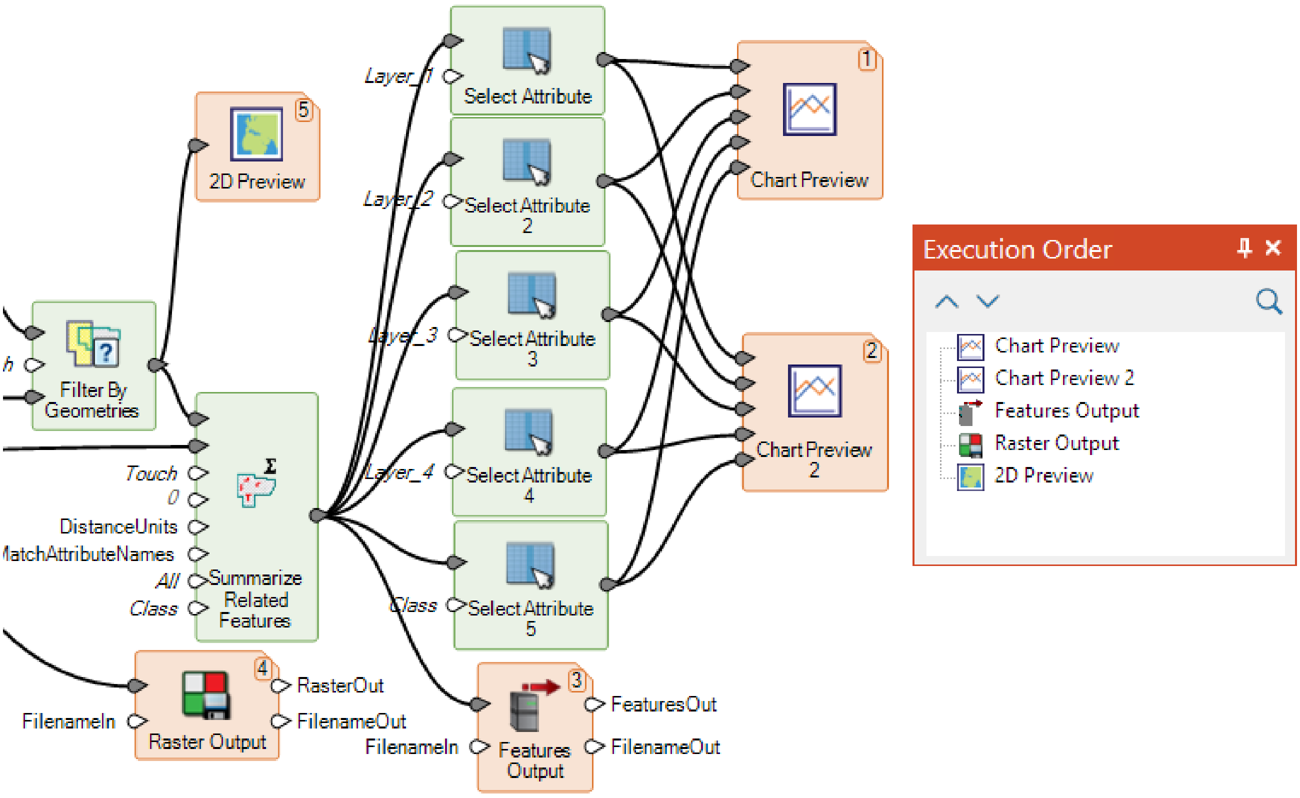
Execution Order

A new panel has been added to the Spatial Model Editor called Execution Order.

Spatial Modeler uses a pull model type of observer pattern where the end of the model chain requests data from its parent(s) and waits until the requested data is provided. This allows the chain to process only the data necessary to satisfy the request, so a subset of the input data may be used. Because of this, all execution of a model starts at the tail end of the chain. Any operator that does not have one or more of its output ports connected to another operator is a tail. A model may have multiple tails. In some cases, it is useful to recognize tail operators and know in which order multiple chains in a model will be executed.

The Execution Order panel shows the order in which execution of operators in the current model will be initiated and allows you to change the order.

When the Execution Order panel is active, each tail operator in the current model will have a number shown in the upper-right corner. This number indicates the order in which the operator will be executed.



Format support in ERDAS IMAGINE 2023 Update 1

DS-EO support

DS-EO imagery can be orthorectified using RPCs for quick and accurate georeferencing.

RPF (CADRG and CIB)

RPF products containing multiple images default to opening as a mosaic for easier access.

On Demand Data (ODX)

ODX was originally introduced in ERDAS IMAGINE 2023 as a space-saving concept based on the capability of a Spatial Model to provide on-the-fly processing and preview of raster data. These “virtual datasets” helped minimize overuse of valuable disk space. However, the concept was originally limited to raster data. ERDAS IMAGINE 2023 Update 1 has expanded support to features and point clouds.

General ERDAS IMAGINE 2023 Update 1

The following section highlights some other new capabilities added to ERDAS IMAGINE.

Normalized Difference Red Edge (NDRE) index

NDRE has become a common vegetation health index, especially in precision agriculture, replacing the use of NDVI. So, while it could already be easily applied via the Indices dialog, it has been added as a standard named option.

The normalized difference red edge index (NDRE) is a vegetation index used in remote sensing for measuring the chlorophyll content in plants, which can be an indicator of crop health, biomass, etc. It is similar to the NDVI but uses the ratio of Near-Infrared and Red Edge wavelengths as follows:

$$NDRE=(NIR-RE)/(NIR+RE)$$

The Red Edge is the part of the electromagnetic spectrum centered around 715 nm.

The Red Edge light of the electromagnetic spectrum isn’t as strongly absorbed by chlorophyll pigments as Visual Red light in vegetation, so the light can penetrate deeper into the canopy. Therefore, NDRE not only estimates the chlorophyll content of the outermost layer of a vegetation canopy, but also for that at lower levels, making it ideal for detecting the health of later-stage crop growth or of dense forest ecosystems.

Added support for EPSG 5682, 5683, 5684 and 5685 (DB_REF)

Several EPSG codes used by engineering survey and topographic mapping for railway applications in Germany (DB_REF) have been added.

Saudi Arabian National PCS references (SANSRS)

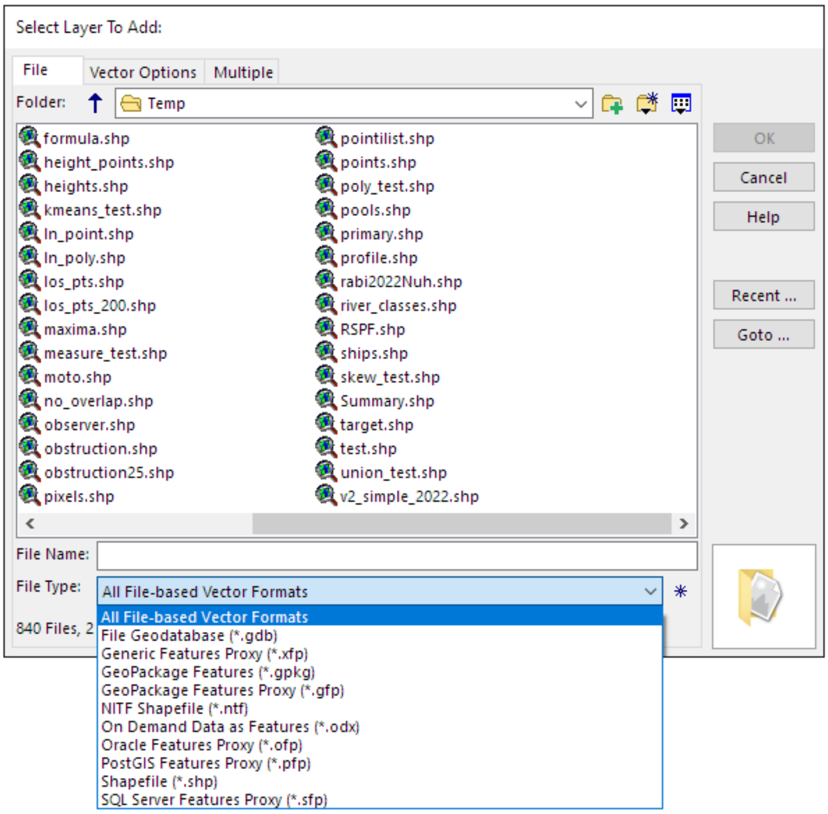
Support has been added for the new Saudi Arabian National PCS references (SANSRS).

New Colombian Cartographic system

Support for EPSG code 9377 (CTM12): new Colombian Cartographic system has been added.

All File-based Vector formats

The file chooser now has an option in the File Type menu for “All File-based Vector formats,” which can also be set as the default filter when opening vector data. This makes it quicker and easier to search for you vector data.



Direct-read NITF embedded shapefiles (CSSHPA and B) as vector layers

Previously, the CSSHPA & CSSHPB DES shapefiles embedded in some NITF files had to be extracted as physical files before they could be displayed or used. Now they can be read directly from the NITF file without conversion.

Save As… to LAZ format

Previously point cloud data could only be saved to LAS format. Now it can also be saved to LAZ.

Issues resolved: ERDAS IMAGINE 2023 Update 1

IMAGINE Essentials

Support ticket	Summary
00187940	Displaying Pleiades Neo data via Image Chain causes “bad allocation” and endless Spatial Modeler errors if Preferences are set to request a band for display that is not present in the image.
00210282	Send to Word outputs corrupt image from thematic Image Chain data which has large extents of NoData
00171523	NetCDF file does not open with coordinate information
00188965	Reprojection from EPSG: 32630 to EPSG: 27700 is not correct

IMAGINE Advantage

Support ticket	Summary
00208579	Recode Operation giving different results in Versions of 2018, 2022 and 2023
00197675	Export ADRG dialog closes unexpectedly when changing Chart Series or Country Code values
00146814	Ribbon ICA (Independent Component Analysis) model fails with larger (> 90GB) input image
00022744	MosaicPro Active Area option is incorrectly calculating image area
00217270	Color table of input thematic data not applied to MosaicPro output
00138780	Wide strip of data missing from recent Capella SLC image after orthorectification
00194171	Subset of LAZ format point cloud results in incorrect output
00175664	Convolution tool does not recognize NoData pixels
00188661	Perform Affine Resample outputs empty raster
00215404 00186503 00185573	ERDAS IMAGINE 2023 (version 16.8) is presenting a resample problem when using Control Points tool

IMAGINE Expansion Pack

Support ticket	Summary
00192642	AutoSync interface becomes unusable (missing icons) after APM

IMAGINE Photogrammetry

Support ticket	Summary
00190625	Adding TIFF to block file crashes IMAGINE if auto-associate preference for elevation library is on
00195107	PRO600 Snapping issue while collecting lines in stream mode
00217191	Invalid Units error: Photogrammetry

IMAGINE Professional

Support ticket	Summary
00113276	Accuracy Assessment does not enforce minimum number of points when all classes are not selected
00023289	Classifyisodata fails when output file name or path has parenthesis
00023307	New/Open Zonal Change project resets the custom layout to default

IMAGINE Objective

Support ticket	Summary
00023237	Zonal Mean calculation discrepancy exists between Objective VOP Zonal:Mean cue and IMAGINE Thematic Zonal Attributes

Spatial Modeler

Support ticket	Summary
00181926	Classify Using Machine Learning model fails in ERDAS IMAGINE 2023
00023102	Initialize Random Forest fails with big data (266863 features)
00187852	Spatial model preview fails with error "invalid vector<T> subscript"
00190141	Importing 'OpenCV' and/or 'Numpy' in the Python Operator throws errors
00187068	"Unsupported moniker" error reported from Spatial Modeler 2023

New technology in ERDAS IMAGINE 2023

This section describes the enhancements provided by the original release of ERDAS IMAGINE 2023

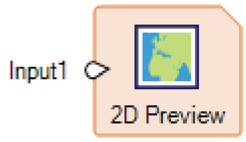
Spatial Modeler

Preview operators

The Preview operator has been renamed 2D Preview and three additional types of Preview operators have been added: 3D, Chart and Web.

Previews are tools provided by the Spatial Modeler Editor that are generally used for visualizing intermediate and terminal outputs of a Spatial Model during development to quickly determine if the processing appears to be producing the expected result. Rather than having to wait for the data to be processed in its entirety and outputs written to a disk, Previews can pull just the data required through the model to display the selected extent and scale. For example, if previewing raster via 2D Preview results in a 2D View with a Fit to Window extent, only the appropriate pyramid level of the input data will be used for processing. Conversely, if the data is displayed at 1:1 raster pixels to screen pixels, only the pixels that fall within the extent of the 2D View will be processed and displayed. In this manner, the amount of data processed is minimized so results can be viewed more quickly and analyzed for the impacts of changing upstream parameters. Note that this relies on there being no upstream operators that are global in nature (i.e., that require the entire extent of the data to be processed at full resolution to produce results, such as the Clump or Intersect Features operators.) If there are global operators present, the benefits of processing only the displayed scale and extent will be diminished, and previewing may be slower if your incremental changes to the model are upstream of any global operators.

2D Preview



When you click on the Preview button on the Home tab, the data connected to a 2D Preview operator is displayed in a 2D View. If there is an empty 2D View, the layer will be created in that 2D View; if there is no empty 2D View, a new 2D View is created. A separate layer will be created for each item on each Input port on the 2D Preview operator, and a separate 2D View is used for each 2D Preview operator in the model. Any port that does not contain data will be ignored; however, if no input ports contain data, the operator will give a warning and no preview will be created.

If the data connected to the Input port is raster, an Image Chain layer is created and one of the Image Chain context tabs will be available for the layer, depending on the type of the raster. If the data connected to the Input port are Features or PointCloud, the layer in the 2D View will have a Preview context tab.

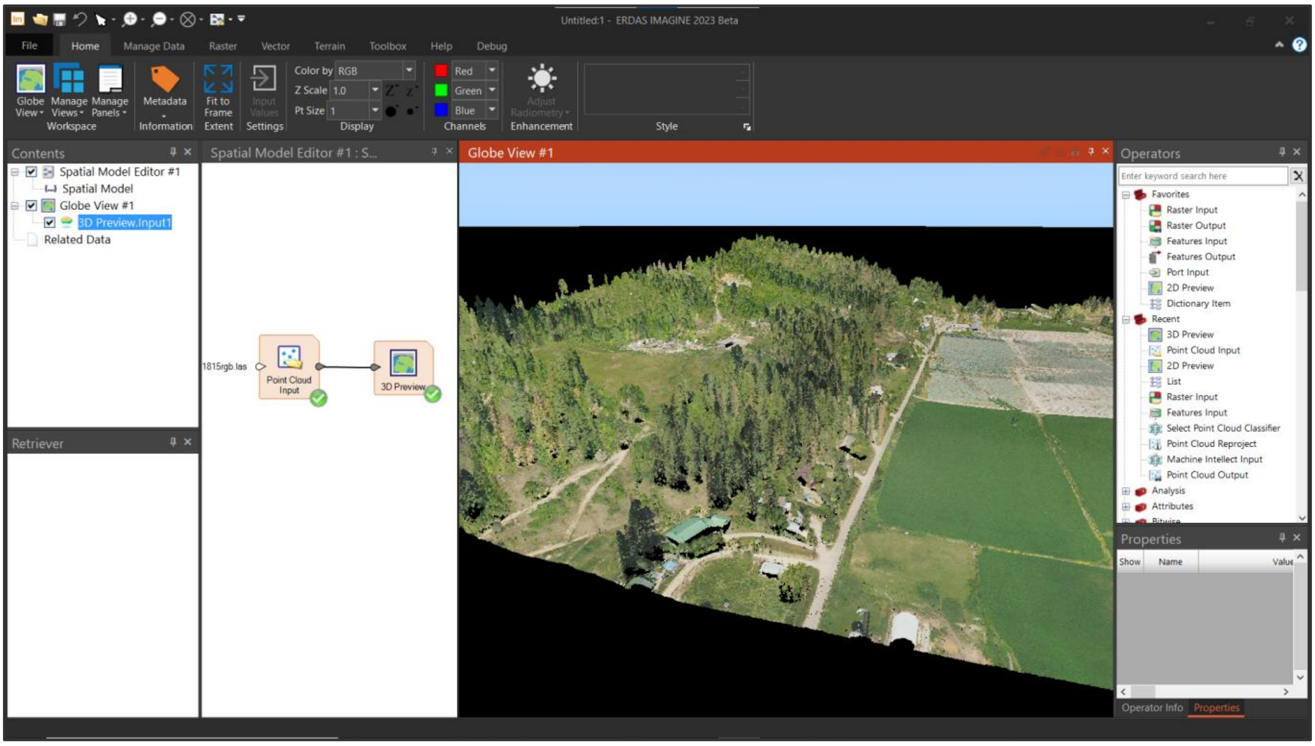
PointCloud data can also be previewed in 3D by adding a 3D Preview operator to your model.

2D Preview now accepts List of IMAGINE.Features and List of IMAGINE.PointCloud (along with the already supported List of IMAGINE.Raster) to facilitate previews of outputs from Iterators and others.

3D Preview

. 3D Previews are designed to enable efficient review of point cloud information flowing through a spatial model. For example, you may wish to visualize the results of removing noise from point cloud data in the Globe View. This will give you the opportunity to assess if the noise points have been removed from the data and, if necessary, to tweak the parameters.

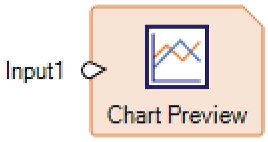
When you click on the Preview button on the Home tab, the data connected to the 3D Preview operator is displayed in a Globe View. A separate layer will be created for each item on each Input port on the 3D Preview operator, and a separate Globe View is used for each 3D Preview operator in the model. Any port that does not contain data will be ignored; however, if no input ports contain data, the operator will give a warning and no preview will be created.



Limitations:

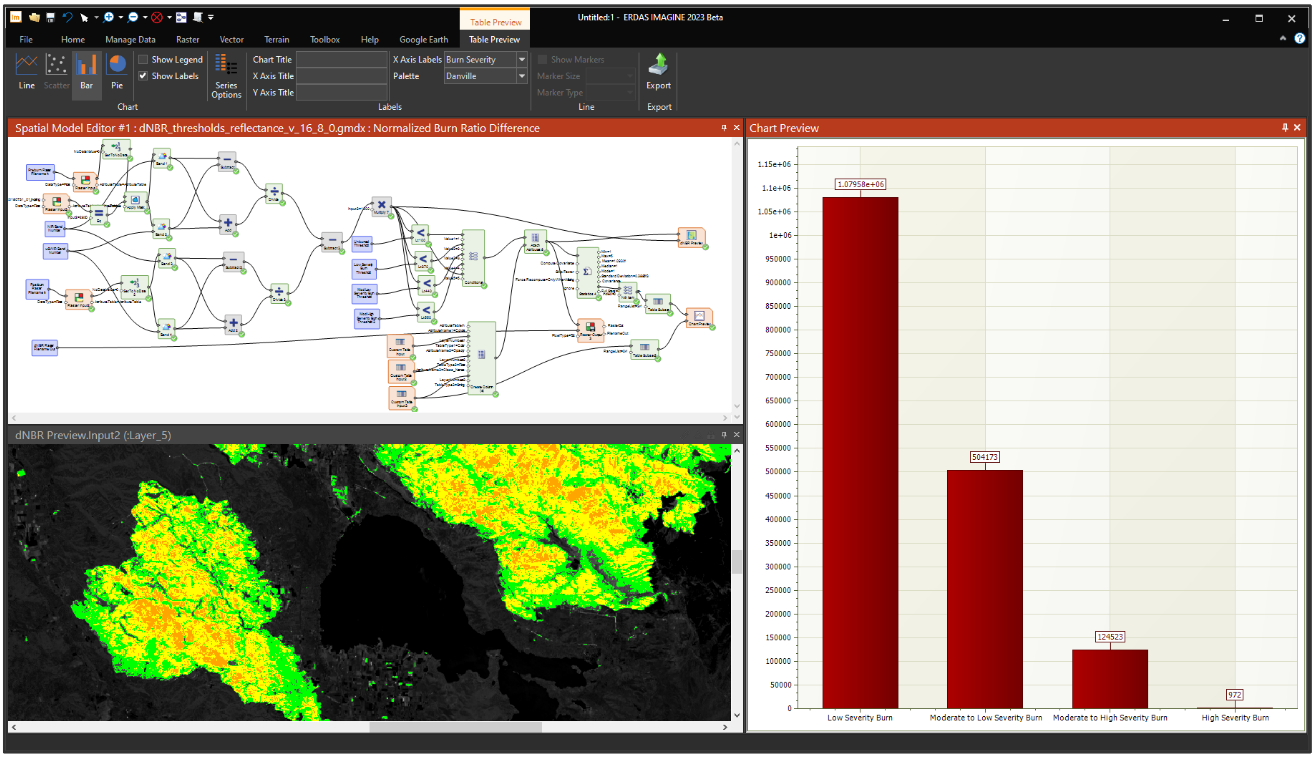
Globe Views require a graphics card that supports OpenGL 4.2 or higher. If the requirements are not met, the Globe View will fail to open with a message advising you of the nature of the shortcoming(s).

Chart Preview



View Table data on the fly in a Chart.

Previews are tools provided by the Spatial Modeler Editor that are generally used for visualizing intermediate and terminal outputs of a Spatial Model during development to quickly determine if the processing appears to be producing the expected result. Chart Previews are designed to provide an efficient method for reviewing tabular information in a spatial model. For example, you might wish to visualize if the distribution of classes in landcover data appears to be correct by sending the histogram information to a Table, inputting that to a Chart Preview and generating a pie or bar chart of the relative areas covered by each class. Altering an upstream parameter in the Spatial Model might then cause a change in the visible distribution of classes in the pie chart.

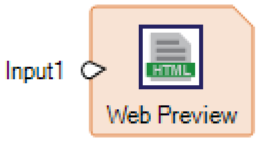


When you click on the Preview button on the Home tab, the data connected to the Chart Preview operator is displayed in a Chart Preview pane. The data from all Input ports on the Chart Preview operator will be displayed in a single Chart Preview pane, and a separate Chart Preview pane will be created for each Chart Preview operator in the model. Any port that does not contain data will be ignored; however, if no input ports contain data, the operator will give a warning and no preview will be created.

The values to be graphed are usually the numeric values present in one or more input Tables. String type tables can also be input, but their use is primarily limited to labeling the corresponding numeric values from a second Table. Color type tables can be used for the color palette of line, scatter or bar charts or for the colors in pie charts.

The screenshot below shows a Spatial Model that analyzes polygonal geometries to be used as training areas for supervised classification. The pixels falling within the extent of each class polygon are mapped as band-to-band scatterplots in the Chart Tools to visually explore the separability of the training data.

Web Preview



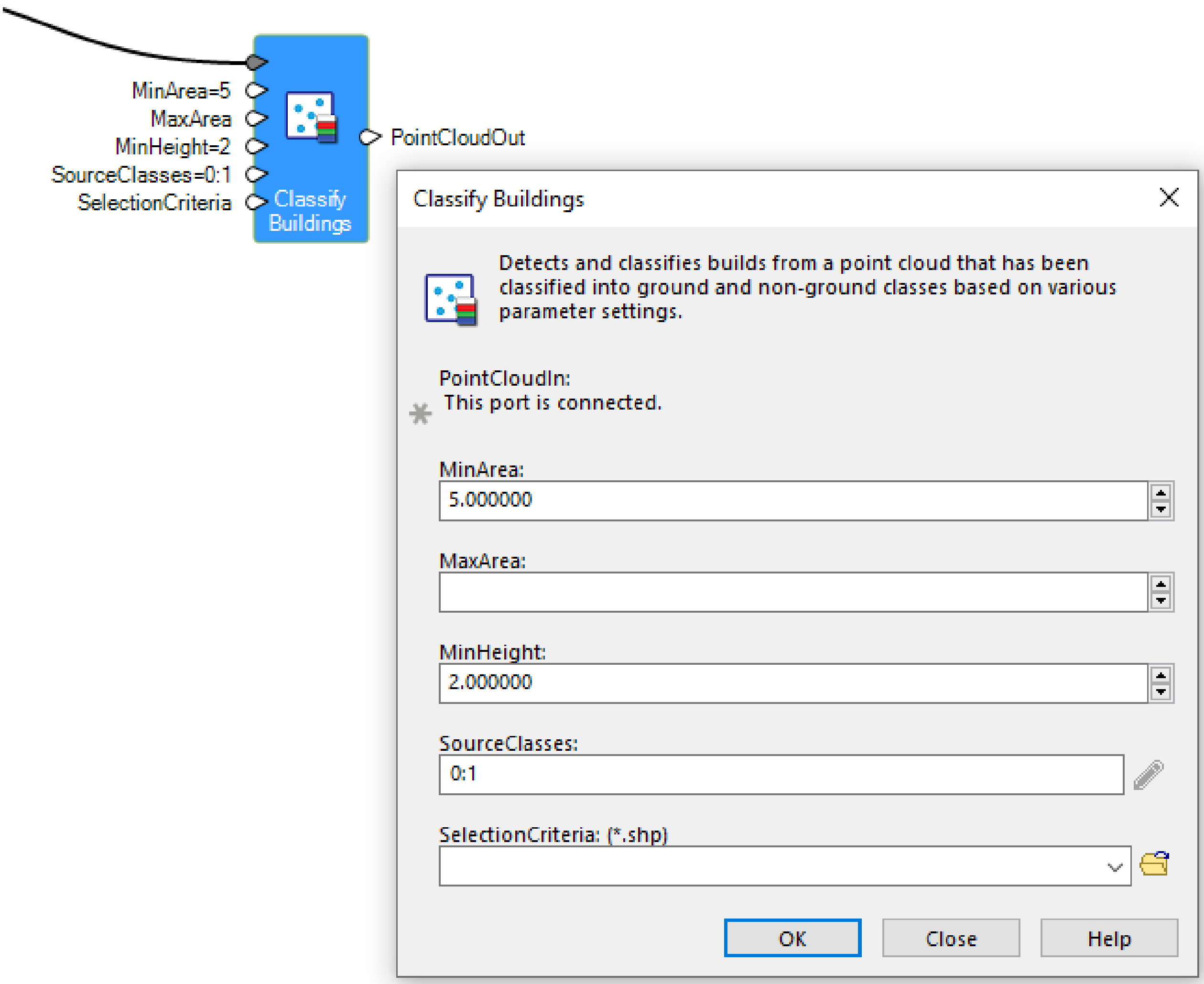
The Web Preview operator enables you to view a file in an embedded browser pane. When you click on the Preview button on the Home tab, the files specified by the connections to the Web Preview operator are opened in an embedded browser pane. A separate browser pane will be created for the file(s) on each Input port of each Web Preview operator in the model. Any port that does not contain data will be ignored; however, if no input ports contain data, the operator will give a warning and no preview will be created.

Any file type that is supported by a web browser – such as GIF, MP4 and PDF – can be viewed.

Auto-generate an operator user interface (UI) for those that lack dedicated ones

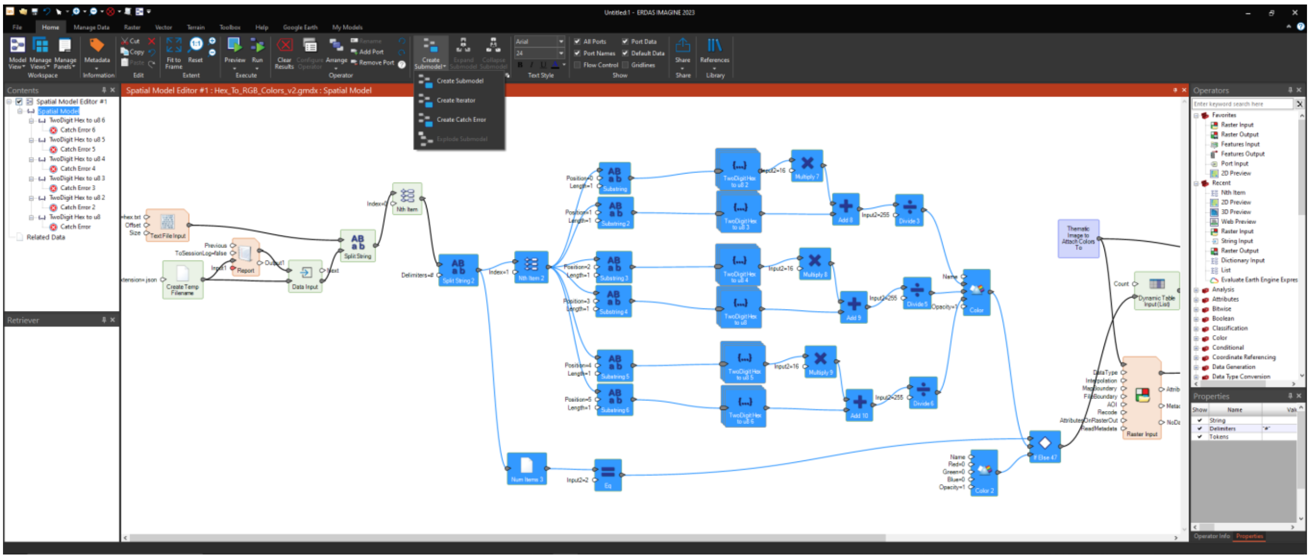
In a prior release, a function was added to change the cursor if it passed over an Operator that could start a dedicated dialog if double-clicked. Confusion among users remained as to when an operator could be double-clicked.

Consequently, the 2023 release extends the capability so that all operators can be double-clicked to start a configuration dialog. If they do not have a dedicated dialog for editing properties, then one will be auto-generated in a similar fashion to the auto-generated dialogs when a Spatial Model is Run or Previewed.

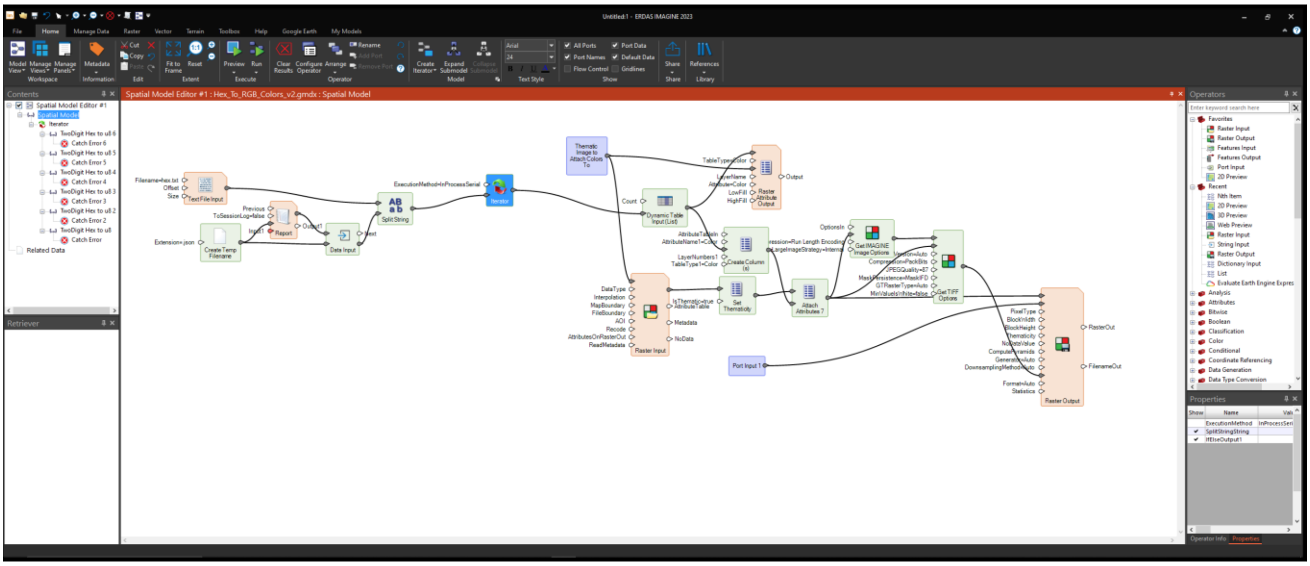


"Create Iterator" and "Create Catch Error" options added to Ribbon tab

When prototyping a Spatial Model, especially when creating sections that require use of an Iterator or other sub-model, it's often easiest to build and test that section of the model using just one of the inputs, such as simulating a single iteration. Once the model is working correctly with one item you can turn that section into an Iterator. With the 2023 release this is made easier with the addition of "Create Iterator" and "Create Catch Error" options that apply to the selected operators:



…helps create:



Spatial Models as On-Demand Data (odx)

On-Demand Data is a space-saving concept based on the capability of a Spatial Model to provide on-the-fly processing. Consider a situation where you have a 1GB Landsat multispectral image. From this image you might want to derive: an NDVI; a Leaf Area Index (LAI); a pan-sharpened true-color view; a Snow and Ice Index; an atmospherically corrected surface reflectance version; a land/water mask; a Tasseled Cap; the first three Principal Components; and a change detection when compared to a different date image. Creating all those products as output files would be easy using individual spatial models, but each results file would consume considerable additional disk space and time.

Instead, On-Demand Data (.odx) can be created for each product. These "virtual datasets" consume negligible space and time. Only the original input Landsat multispectral image file needs to be stored on disk. The other products are derived on the fly from the Spatial Model that defines the On-Demand Data. But the On-Demand Data can still be used like other input datasets in ERDAS IMAGINE (or other applications that use Spatial Modeler) to be viewed in a 2D View, fed into an Unsupervised Classification, exported to an exotic file format or fed into another Spatial Model.

Looking closer at one of the examples above, you might want to take the Landsat multispectral image and calculate an LAI to look at vegetation health in the area. However, LAI requires the input data to be Reflectance data (in the unitless 0-1 approximate range.) The Landsat file has been corrected to Surface Reflectance by the data provider (USGS), but it was also scaled and offset to unsigned 16-bit integer for ease of dissemination (along with a metadata file that provides information such as the multiplier and offset for each band to convert back to floating point surface reflectance values). It is therefore easy to build a small Spatial Model that takes in the Landsat image, reads the metadata file, extracts the multiplier and offset values and applies these to the DN values to produce true floating point surface reflectance values suitable for feeding to the LAI function. But you don't want to create a floating-point version of the Landsat image on disk simply to feed it into the LAI function and produce a third dataset. So rather than (or in addition to) adding a Raster Output operator in the Spatial Model to convert to Surface Reflectance, a Preview operator is added at the point where the required data would be produced by the model. If a Preview operator is selected in the Spatial Model Editor, an option will be enabled to Create On-Demand Data. This will prompt you to provide a name of the On-Demand Data file (a small .odx file on disk that contains the Spatial Model and pointers to the input data). This On-Demand Data file can then be treated like any other geospatial dataset supported by the software. For example, the On Demand Data file defining how to convert DNs to Surface Reflectance could be fed as input to the LAI option in the Indices dialog.

You may wish to avoid the use of On-Demand Data in some cases. Spatial Models that include global operations may not be suitable for efficient display in a 2D View, for example. Or if the data will be re-used multiple times, such as needing the Landsat Surface Reflectance values as input to several different processes. In these cases, it might be more efficient to produce an actual dataset on disk.

Create an On-Demand Data File

If a Spatial Model contains one or more Preview operators an On-Demand Data file can only be created corresponding to one selected Preview operator. However you could create multiple Virtual Datasets, one for each Preview.

A single Preview operator can also have multiple input ports. In this instance, the On-Demand Data derived from that Preview will be considered a multilayer dataset. The File Chooser will present the Sub-Images tab (or another appropriate tab) so that the desired layer can be selected. If displayed in a 2D View the On-Demand Data will populate a Related Data panel.

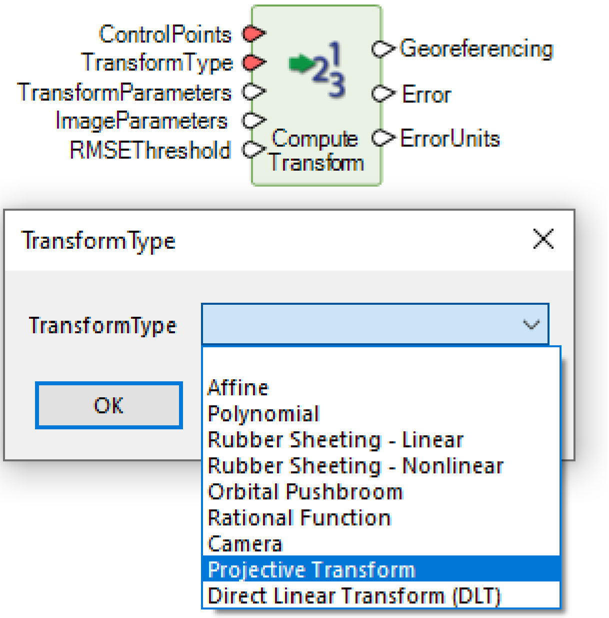
On-Demand Data is also not necessarily limited to raster data.

Open an On-Demand Data File

In a File Selector, click the File Type dropdown arrow to select On-Demand Data (*.odx).

Updated Spatial Modeler operators

Compute Transform

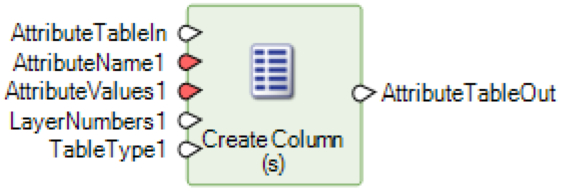


This operator takes control points and their measurements in an image and creates a transform from the image coordinate system to the control point coordinate system.

ERDAS IMAGINE 2023 has been updated to support the creation of a wider range of 3D orthorectification transform types including:

- Camera
- Projective Transform
- Direct Linear Transform (DLT)

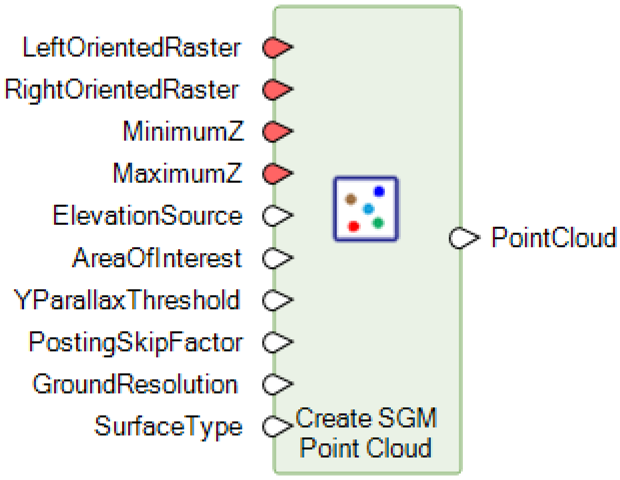
Create Column(s)



A column is defined by a name and a Table of values. The Attribute Table set on the **AttributeTableOut** port can be added to a Raster or Features using other operators, such as to add color definitions or class names for thematic pixels.

The operator has been updated to automatically adjust the number of rows in cases where the number of rows defined on **AttributeTableIn** and **AttributeValues*n*** differs.

Create SGM Point Cloud



This operator extracts a point cloud from a stereo image pair using Semi-Global Matching (SGM) and saves it to an output point cloud.

It has been enhanced with a new **GroundResolution** port to account for high vs. low resolution input images.

Accepted values are:

- Auto
- High
- Low

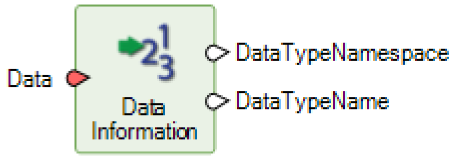
In High-resolution images, features such as buildings, utility poles, tree canopies, road markings and other infrastructure details are clearly visible. Images with a ground resolution of 50 cm or smaller are considered High-resolution images.

The point cloud creation process attempts to extract the elevations of these features as accurately as possible.

If set to Auto (the default), the operator determines if the input images are High-resolution or not.

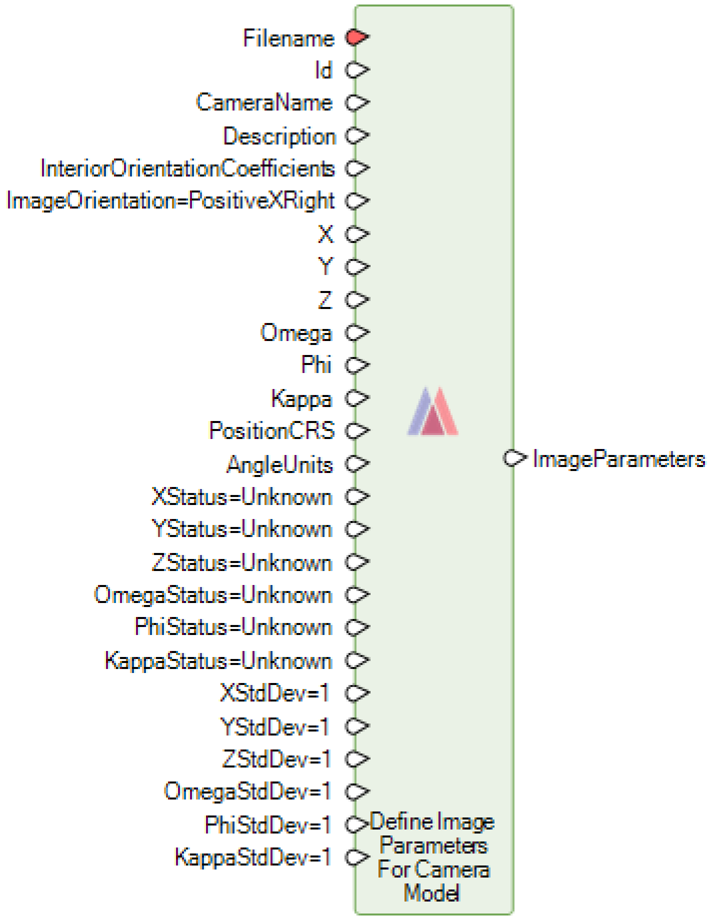
This option is applicable only when **SurfaceType** is "Sharp."

Data information



This operator provides basic information about a data stream. Use of this operator in sub-models has been made easier via better error messaging.

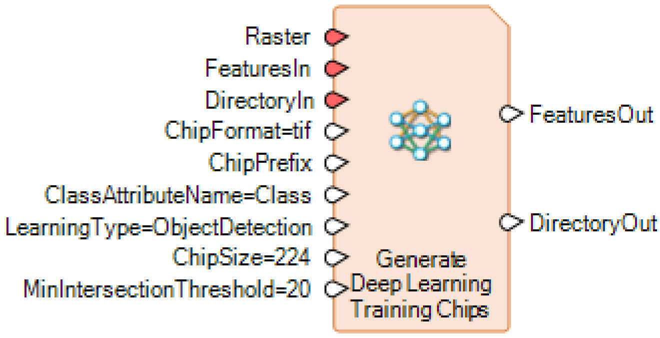
Define Image Parameters For Camera Model



This operator creates a set of image parameters that are used when adding a camera sensor model to an image.

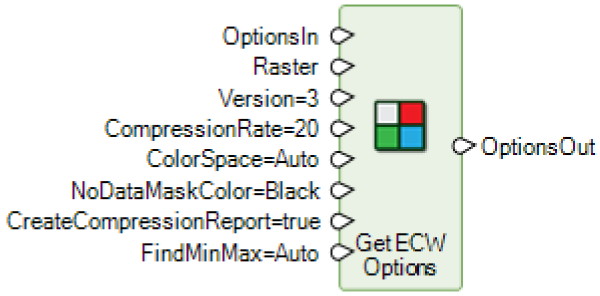
The **CameraName** port has been modified to optionally take an input dictionary in which the full set of parameters that describe the frame or digital camera that generated the data may be specified. If these image parameters are added to a block, the camera parameters will also be added if they do not already exist.

Generate Deep Learning Training Chips



Generate Deep Learning Training Chips is now capable of generating image chips for use in Semantic Segmentation.

Get ECW Options



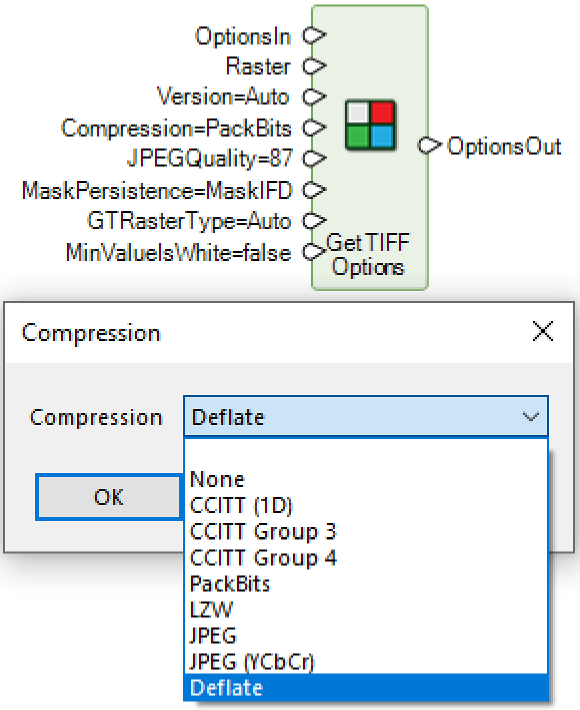
This operator creates the format-specific output-option dictionary for Enhanced Compressed Wavelet (ECW) format. This can then be fed to the Raster Output operator to control how an output ECW file might be generated.

When compressing 16-bit imagery to ECW it is important to provide the minimum and maximum DN values of the input data to avoid over-compression of the output. For example, if the data effectively has an actual bits per pixel of 11 bits, but has to be stored as 16 bit, it is important to specify the range is 0-2047, not 0-65535. Consequently, a **FindMinMax** port has been added.

If set to True and there are no statistics available on the input **Raster**, this port option causes the software to go through the pixels to find the minimum and maximum values. This improves compression and image quality, but will take more time.

When Auto is selected, it is equivalent to a **FindMinMax** value of False for 8-bit output and **FindMinMax** value of True for 16-bit output.

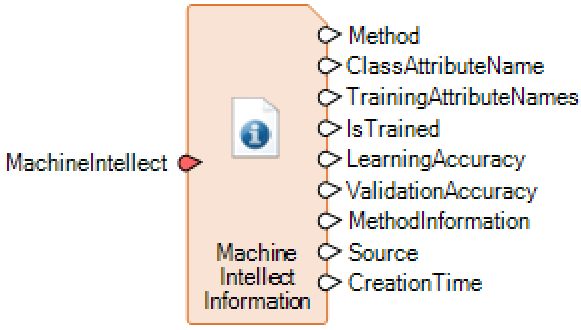
Get TIFF Options



The Get TIFF Options operator creates the format-specific output-option dictionary for Tagged Image File Format (TIFF) Support.

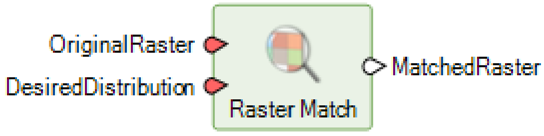
An option has been added for reading and writing using Adobe Deflate compression.

Machine Intellect Information



This operator provides basic information about a machine intellect and has been extended to provide information about imported deep learning models (see the Import operators detailed below.)

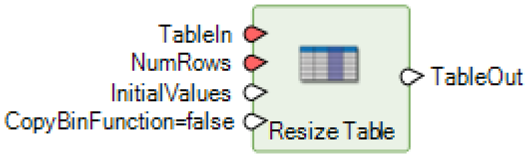
Raster Match



This operator is used to change the distribution of pixel values in a Raster (**OriginalRaster**) to be like a given distribution (**DesiredDistribution.**) **DesiredDistribution** can either be another Raster or a Table (such as an ideal histogram distribution.)

The operator has been corrected to now work with 16-bit imagery.

Resize Table



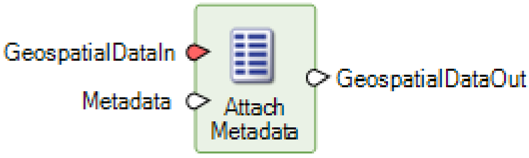
This operator has been enhanced to copy the (appropriately adjusted) bin function when resizing a Table.

The **CopyBinFunction** port specifies whether to copy the bin function (if present) from **TableIn** to **TableOut**.

If **CopyBinFunction** is true and **TableIn** has a Direct or Linear bin function, the bin function on **TableIn** will be extended or truncated (depending on whether rows are added or removed) and set on **TableOut**.
If **CopyBinFunction** is false or **TableIn** has no bin function, **TableOut** will have no bin function.
If **CopyBinFunction** is true and the bin function on **TableIn** is not Direct or Linear, the operator will fail.

New Spatial Modeler operators

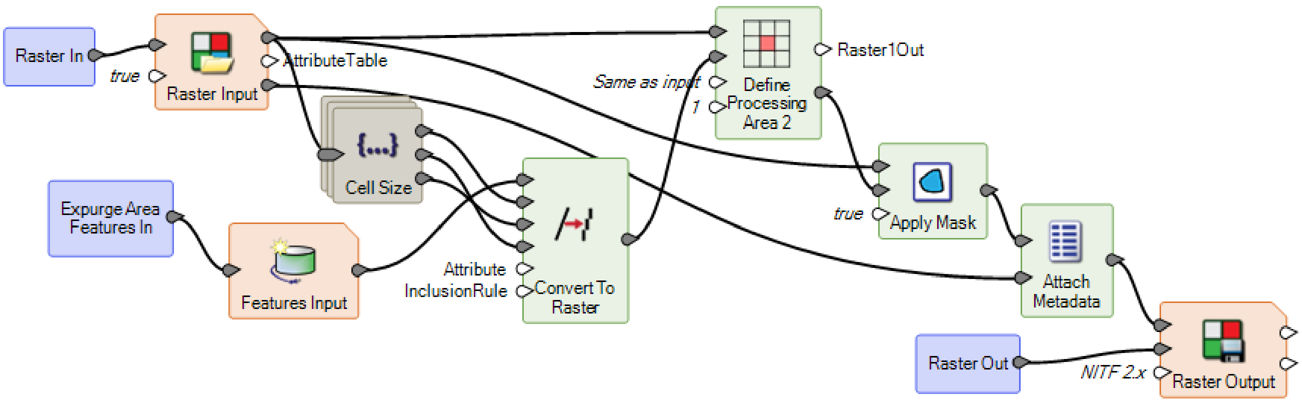
Attach Metadata



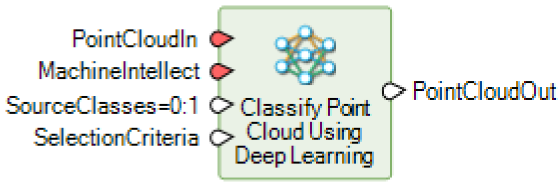
This operator attaches metadata to **GeospatialDataIn**. Any existing metadata is replaced. If no metadata is supplied, **GeospatialDataOut** will have no attached metadata.

Only the metadata is affected by this operator. Properties that might be removed by other streaming operators due to alteration of the streamed data, such as the Attributes or Vertical CRS of a Raster, are fully retained.

Below is an example of using the Attach Metadata operator to reattach the metadata from the input raster to the output raster, for example when performing a Raster Expurgation operation.



Classify Point Cloud Using Deep Learning

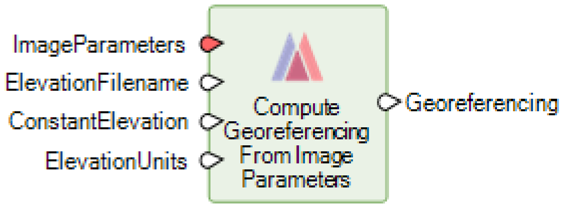


This operator uses deep learning to classify points from the input point cloud.

Only those points whose current class value matches one of the value(s) specified in the **SourceClasses** port will be considered for classification. By default, class values 0 (Never Classified) and 1 (Unclassified) are considered as source classes.

The points from **SourceClasses** can be filtered further by providing additional criteria using the **SelectionCriteria** port. All points input via **PointCloudIn** will be output to **PointCloudOut** even if they were not considered for classification.

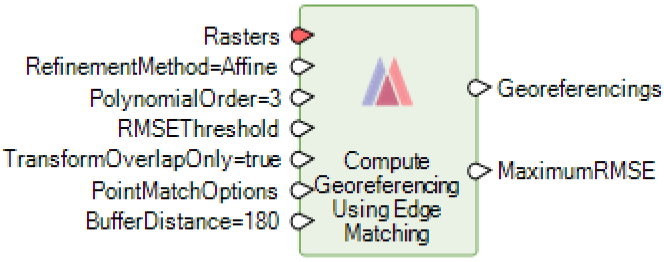
Compute Georeferencing From Image Parameters



This operator creates Georeferencing (of type IMAGINE.CoordinateOperation) from an image parameter dictionary that is based on a sensor model.

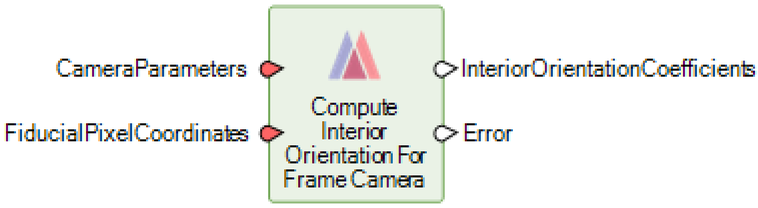
The Georeferencing can then be passed to the Associate Georeference operator, for example, to orient the image with the model. The georeferenced image can be used for downstream processes such as point matching with a reference image using Generate Control Points Based On Reference.

Compute Georeferencing Using Edge Matching



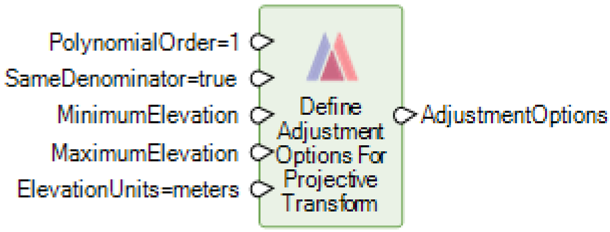
This operator accurately aligns the overlap area between adjacent georeferenced rasters to each other using edge matching. This will generate refined georeferencing for all rasters because the required shifts will be divided between the rasters. Edge matching may also be a good choice when the overlap area is small and it is undesirable to apply the same transform that is suitable for the overlap region to the entire raster.

Compute Interior Orientation For Frame Camera



This operator produces coefficients that can be used to construct an affine transformation that converts from millimeters in film space (Fx, Fy) to pixels in pixel space (Px, Py) of the sensor.

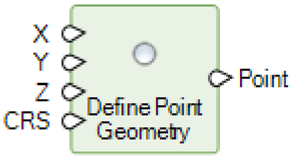
Define Adjustment Options For Projective Transform



This operator defines adjustment options for a 2D/3D projective transform model for use with the Compute Transform operator (see above). If the input points to Compute Transform are 2D, a 2D transform will be created. If the points are 3D, the transform will be 3D.

The Projective Transform model is based on a Rational Polynomial Functions and provides a powerful modeling capability for multiperspective satellite images such as Landsat, SPOT and QuickBird. While the best way to transform and orthorectify is to use sensor-specific models, the projective transform can be used in situations where no sensor model and ephemeris information are available.

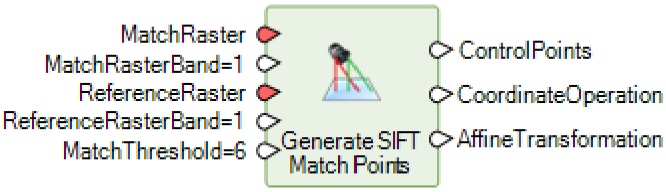
Define Point Geometry



This operator creates a Point geometry from a set of input values representing the coordinate.

The coordinate reference system given to the operator will determine the coordinate inputs that must be set (3D requires X Y and Z; 2D requires X and Y; 1D requires Z). If no coordinate reference system is given, an unknown CRS will be used whose dimensionality will be determined by which coordinate inputs are set.

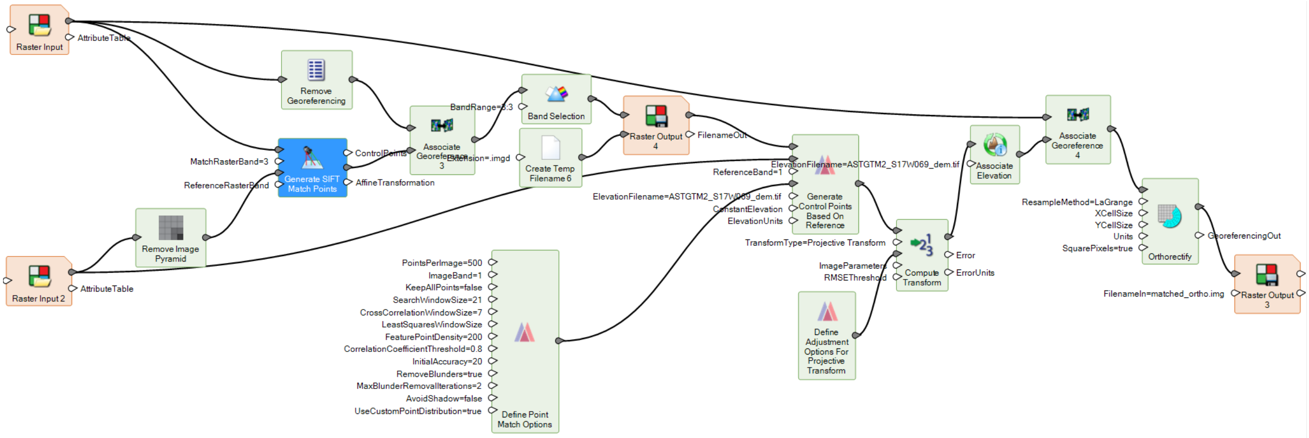
Generate SIFT Match Points



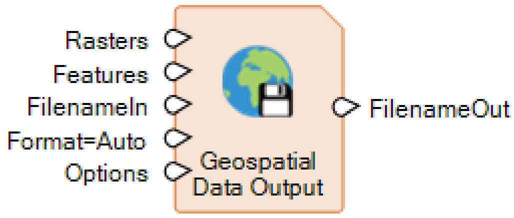
This operator uses a variant of the Scale Invariant Feature Transform (SIFT) algorithm to automatically identify common points (scale and orientation invariant features) that occur in both an input image (which is usually unreferenced) and a reference image. Using these point matches, a transformation is generated that maps the coordinate space of the input image to that of the reference. This technique is commonly used as part of a process to georeference unreferenced imagery based on a geospatially accurate reference image base or to simply co-register two images. It is also commonly used as an initial step in a more complex photogrammetric matching approach.

In addition to generating an affine transformation that can be easily applied to the input image, the automatically generated feature points can also be output to a ControlPoints dictionary so that they can be used in more complex geometric models if desired.

Below is an example of a Spatial Model that uses Generate SIFT Match Points to create an initial location for the unreferenced input image and then further refines the geometry model by using Generate Control Points Based On Reference and then Compute Transform with the new Projective Transform option to then ortho correct the image.



Geospatial Data Output

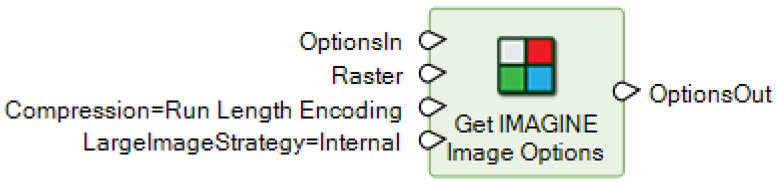


This operator saves geospatial data to a new file-based container. The formats supported by Geospatial Data Output allow for multiple Rasters and/or Features to be part of the same dataset (referred to as a container). Single Rasters may be saved to a larger selection of formats (including non-container formats) via the Raster Output operator. Likewise, single Features may be saved to a larger selection of formats (including non-container formats) via the Features Output operator.

Container formats currently include GeoPackage.

Names for the individual layers in the output container file can be set using the Set Data Name operator.

Get IMAGINE Image Options



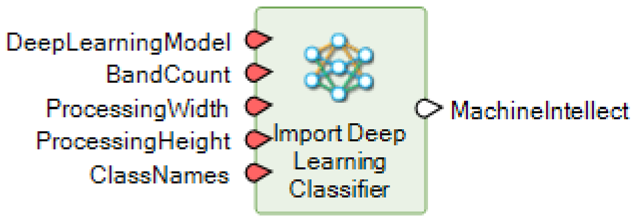
This operator creates the format-specific output-options dictionary for IMAGINE Images.

The primary reason for introducing this operator is to provide control over the new **LargeImageStrategy** for .img files. This provides a new way of storing data Internal to a .img file that includes the ability to losslessly compress data even if it is over 2GB (rather than using an uncompressed .ige external file), as well as being able to store NoData location information as a mask (rather than using a value). However, these new capabilities may not be supported by other software applications, so the option to store using the older External formatting is also provided.

These options work as described below:

LargeImageStrategy	NoData Handling	Comments
External	A mask will not be persisted; NoDataValue handling is unaffected.	When the uncompressed raster size is greater than 2GB, a binary-formatted .ige file is produced.
Internal	A mask will be persisted; NoDataValue handling is unaffected.	When the uncompressed raster size is greater than 2GB, .img files with 64-bit offsets that are not compatible with versions of ERDAS IMAGINE prior to 2023 (version 16.8) and possibly other packages are produced.

Import Deep Learning Classifier



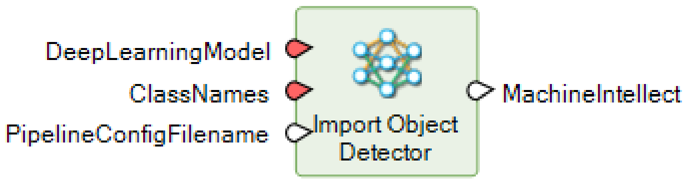
This operator imports TensorFlow deep learning classifiers created with other software.

A classification model is a deep learning model that accepts a batch of rasters and produces a batch of probabilities that each image is of a given class. The model operates on rasters of size ProcessingWidth x ProcessingHeight.

Classification models should accept one array of shape (BatchSize, ProcessingWidth, ProcessingHeight, BandCount) as input, representing the input raster to classify. They should produce an array of shape (BatchSize, ClassCount) floating point numbers as output, representing the probability that the input is of that class.

The model should be a TensorFlow 2 model, saved using tensorflow.keras.model.save or downloaded from TensorFlow Hub.

Import Object Detector

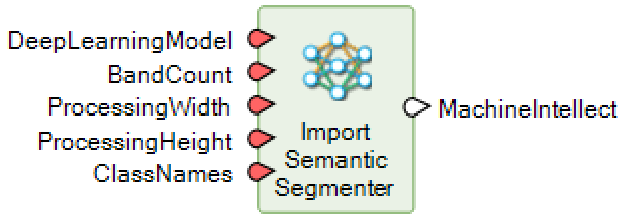


This operator imports TensorFlow deep learning object detector created with other software.

An object detector is a deep learning model that accepts a batch of rasters and produces a batch of detected objects.

Object detectors should accept one array of shape (BatchSize, ProcessingWidth, ProcessingHeight and 3) as input representing the input raster in which to detect objects. ProcessingWidth and ProcessingHeight are defined by the Detect Objects Using Deep Learning operator.

Import Semantic Segmenter

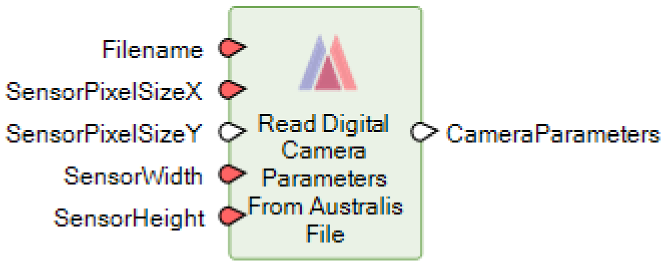


This operator imports TensorFlow deep learning segmentation models created with other software.

A segmentation model is a deep learning model that accepts a batch of raster tiles and produces a batch of rasters where each pixel represents the probability that the input pixel is of a given class. The model operates on rasters of size ProcessingWidth x ProcessingHeight.

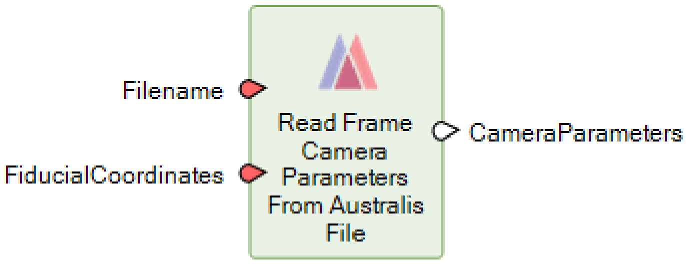
Segmentation models should accept one array of shape (BatchSize, ProcessingWidth, ProcessingHeight and BandCount) as input representing the input raster tile to segment and produce an array of shape (BatchSize, ProcessingWidth, ProcessingHeight and ClassCount) floating point numbers as output representing the probability that each input pixel is of that class.

Read Digital Camera Parameters From Australis File



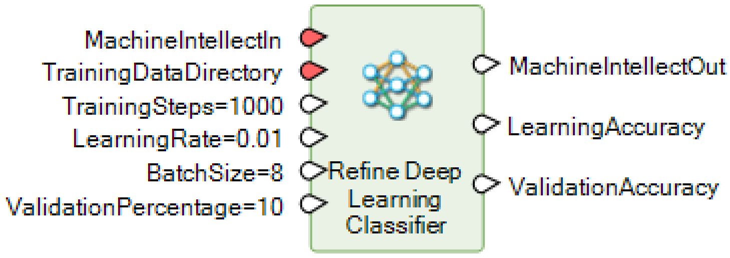
This operator reads digital camera parameters needed for setting up a Digital Camera from an Australis camera file.

Read Frame Camera Parameters From Australis File



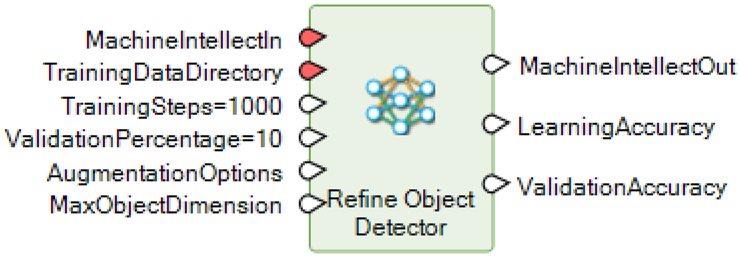
This operator reads frame camera parameters needed for setting up a Frame Camera from an Australis camera file.

Refine Deep Learning Classifier



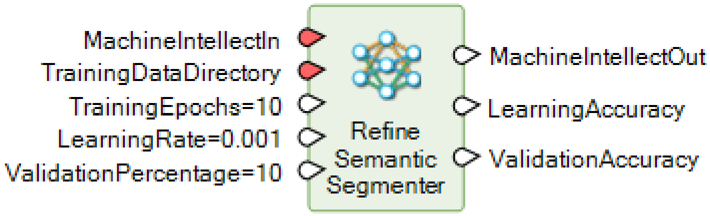
This operator refines the training on an existing deep learning classifier by supplying more training samples. This can be used to improve the overall accuracy or to tailor training for a new environment. The classes and band count in the training data must match the input machine intellect.

Refine Object Detector



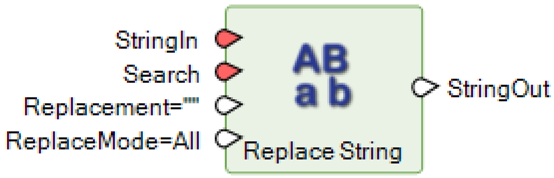
This operator refines an existing object detector by supplying more training samples. This can be used to improve the overall accuracy or to tailor the training for a new environment. The classes in the training data must match the input machine intellect.

Refine Semantic Segmenter



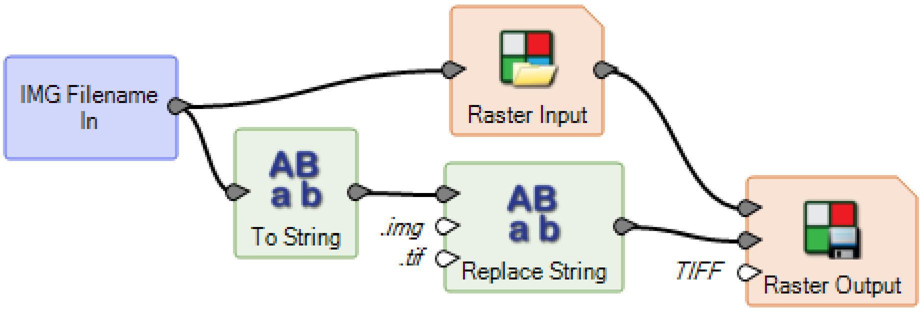
This operator refines the training on an existing Semantic Segmentation machine intellect by supplying more training samples. This can be used to improve the overall accuracy or to tailor the training for a new environment. The classes and band count in the training data must match the input machine intellect.

Replace String

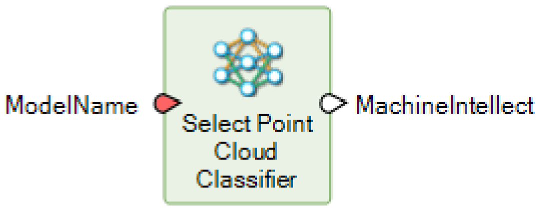


Replace String searches **StringIn** for occurrences of **Search** and replaces them with **Replacement**. You can search and replace the first, last or all occurrences of **Search**. The search is performed case-insensitive (based on the default locale), but the replacement is done with **Replacement** exactly as it is supplied to the operator.

Below is an example of how this operator might be used in a model to convert an input IMG file to TIFF and create the output TIFF file in the same directory as the input IMG.



Select Point Cloud Classifier

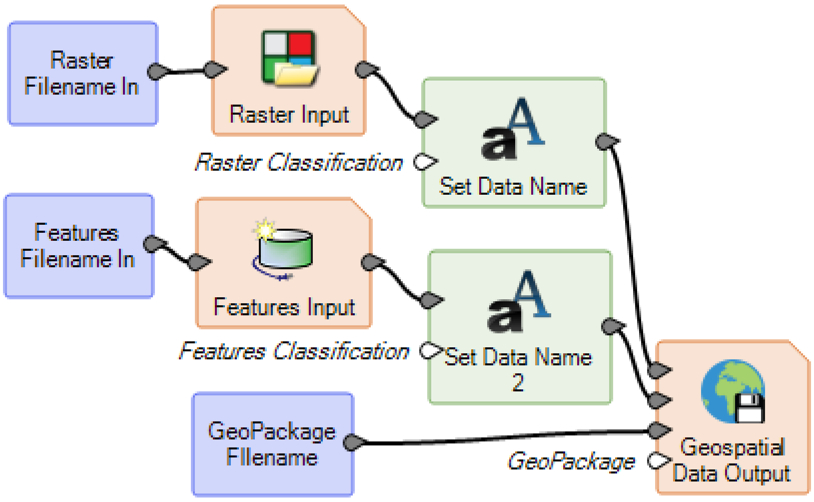


Spatial Modeler provides a selection of trained machine intellects for point cloud classification for different types of datasets/domains, for example, Airborne LiDAR. The operator allows you to select one of these intellects for use with the Classify Point Cloud Using Deep Learning operator.

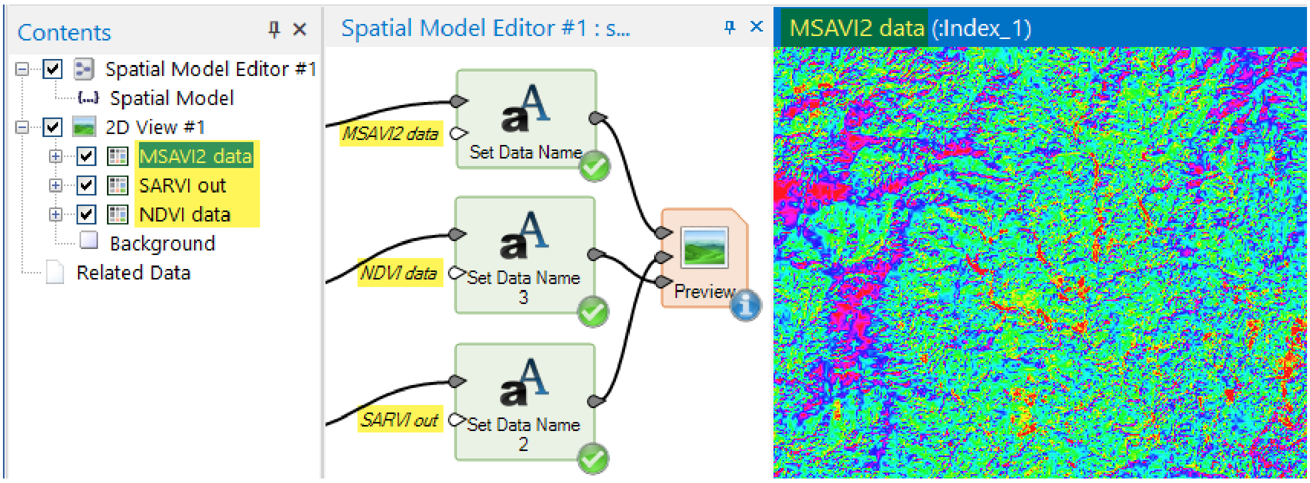
Set Data Name

This operator sets the name property of the data, if supported. This may be used by Output operators to set the identifier of the data. For example, in Geospatial Data Output (see above), the name will be used as the table name when creating a GeoPackage. Another use of this operator is to set the name that will be displayed for the Layer in the Table of Contents when Previewing a dataset.

Below is an example of a model that outputs both a Raster and Features to a single GeoPackage file using the Geospatial Data Output operator. The Set Data Name operator is used to set the name that will be used for the layer-specific table when creating the GeoPackage.



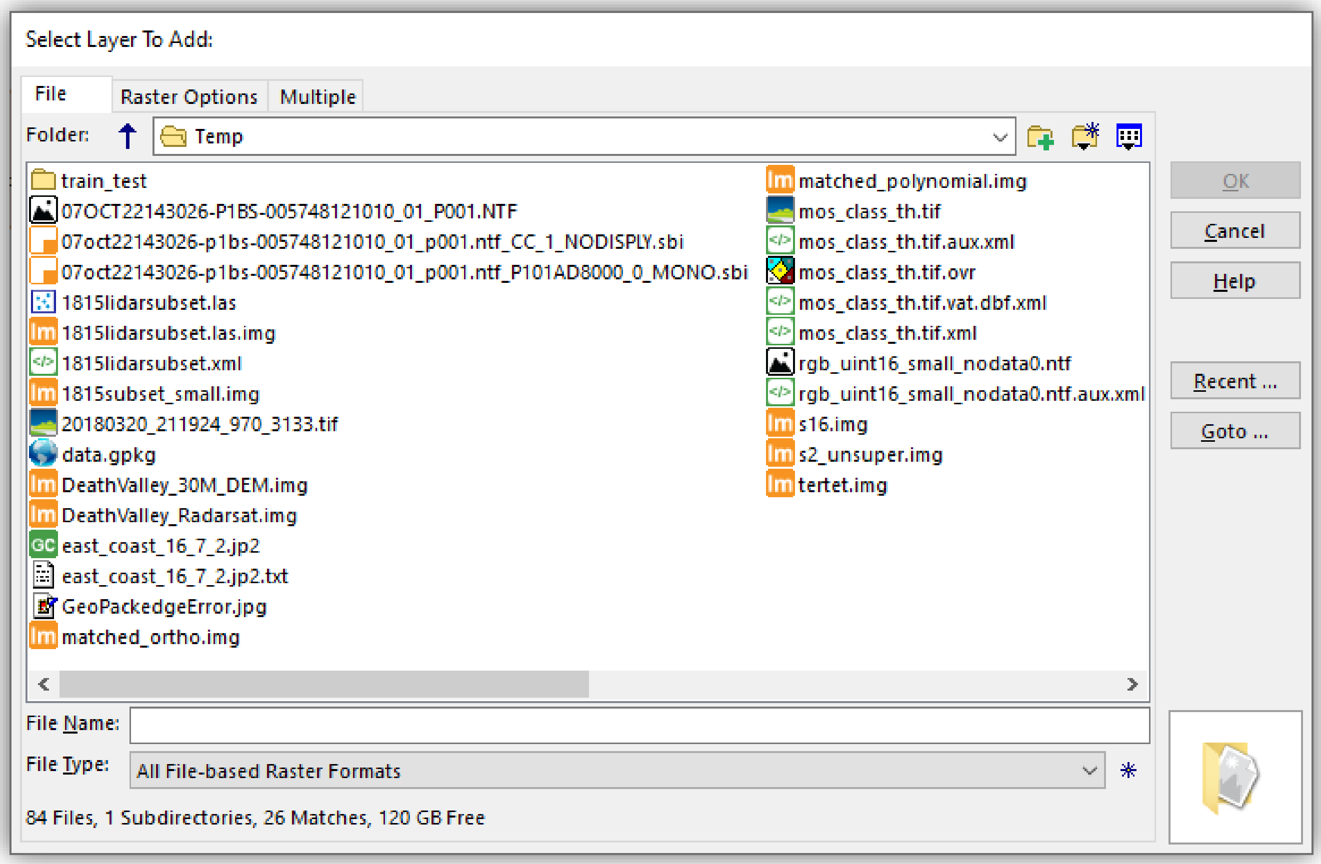
Below is an example of a model that previews three Raster data streams. By using the Set Data Name operator to set the name on the stream before feeding it to 2D Preview, the Table of Contents for the 2D View then identifies each dataset with the given name, making it easy to tell which dataset is which. The title bar of the 2D View also shows the dataset name of the top layer.



Format support

Select Layer To Add dialog displays improved icons for supported formats

In prior releases the Select Layer To Add dialog would show some files, such as .jp2 files, with generic icons such as the “file” page icon, giving the false impression that the file might not be supported. This release has updated numerous file extensions to show more appropriate icons.



IMAGINE Image File format (IMG)

The IMAGINE Image File Format (IMG) has been extended to provide support for lossless compressed IMG files larger than 2GB (rather than using an uncompressed .ige external file), as well as the ability to persist a NoData mask within the IMG file (rather than using a value).

However, these new capabilities may not be supported by other software applications yet, so the option to store using the older External formatting is provided via a Preference. See the IMAGINE Image Files (Native) preference category.

The screenshot below shows the same large thematic image stored first as an External format IMG and second as a new Internal format IMG. Note that the External formatting results in a set of three files that in total are 2.5 times larger than the two files created using the new Internal formatting.

east_coast.ige	2/10/2006 11:42 PM	IGE File	24,108,306 KB
east_coast.img	6/14/2022 3:03 PM	ERDAS IMAGINE D...	17 KB
east_coast.rrd	11/13/2019 3:47 PM	RRD File	2,020,578 KB
east_coast_big_image.img	3/10/2023 3:41 PM	ERDAS IMAGINE D...	9,683,395 KB
east_coast_big_image.img.pyrx	3/10/2023 3:41 PM	PYRX File	145,092 KB

Hexagon Smart Point Cloud (HSPC)

HSPC is a powerful, robust format for handling point cloud data.

ERDAS IMAGINE now directly supports point clouds in the Hexagon Smart Point Cloud format (HSPC). Certain Hexagon scanners allow exporting point clouds in HSPC, a 3D-tiled format enabling customers to avoid additional conversion steps into formats such as LAS. Advantages compared to OGC 3D tiles are better compression of the tile payload and precise encoding of point locations.

Meta Raster Format (MRF) and Limited Error Raster Compression (LERC) format read

Meta Raster Format (MRF) is an image and data storage format designed for fast access to imagery within a georeferenced tile pyramid at discrete resolutions. It was originally developed at NASA's Jet Propulsion Laboratory.

Limited Error Raster Compression (LERC) is a format for compressed raster image data and is often used to store compressed raster data in the MRF format.

Support for reading both of these has been added.

Sentinel-3 SLSTR Level-2 WST read

The SLSTR Level-2 WST product respects the Group for High Resolution Sea Surface Temperature (GHRSST) L2P specification and includes a single SST field derived from the best performing single-coefficient SST field in any given part of the swath, plus a number of supporting data fields providing context for the SST fields.

Adobe Deflate in GeoTIFF

Deflate is a lossless data compression technique that uses a combination of LZ77 and Huffman coding. Support has been added to both read TIFF files using this compression as well as to create TIFFs using Deflate.

Better handling of container formats such as GeoPackage

Container formats are a type of storage format that allows more than a single dataset to be stored. For example, GeoPackage format allows multiple vector and raster datasets to be stored in a single GeoPackage. Prior releases supported reading of data from these container formats.

Spatial Modeler 2023 has been enhanced to enable the writing of multiple datasets into container formats such as GeoPackage (via the Geospatial Data Output operator).

USGS Spectral Library Version 7 support

USGS Spectral Library Version 7 has been added to the available spectral libraries associated with the hyperspectral processing tools in ERDAS IMAGINE 2023.

General ERDAS IMAGINE

New Home tabs

Functionality in the ERDAS IMAGINE 2023 ribbon interface has been reorganized such that the Home tab is now context sensitive to the type of active View or Pane. Consequently, there are five discrete Home tabs. These make the appropriate tools for interacting with the active View much more readily available to the user.

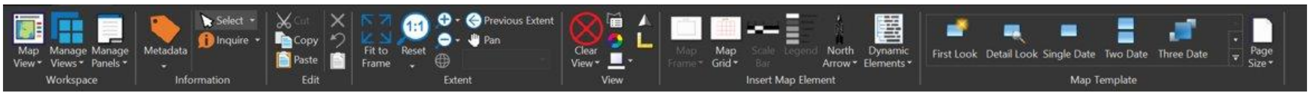
In addition, all the Home tabs have an initial menu (in the Workspace group) that enables any other type of View to be started (or to start another one of the currently active type.)

Every Workspace contains a Home tab, consisting of tool tab groups for Workspace management and information, as well as groups specific to the current View, such as editing, View extent, etc. The tools are context sensitive, meaning that whether tools are active or what they do may depend on the current layer or the currently selected item.

2D View

The 2D View displays raster, vector, point cloud and annotation data in a 2D View window. The Home tab for a 2D View is as shown above.

Map View

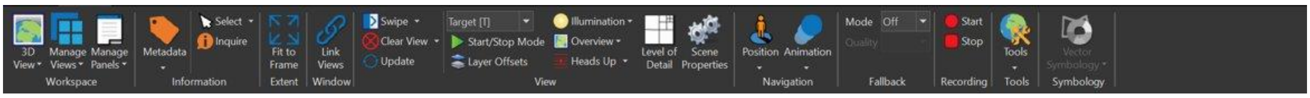


The Home tab for a Map View contains tools for creating a Map Composition in the Map View.

A map composition or map file is a file created with several components of the ERDAS IMAGINE Workspace, usually to be output to hard copy. The map composition contains map frames and map annotation. Map composition files have the extension .map. They can be created, edited and viewed in the Map View.

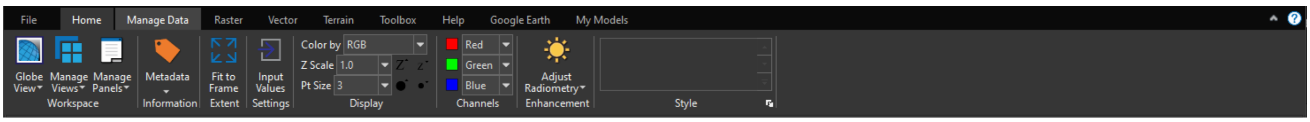
The Home (Map View) tab and the Drawing tab are the primary tabs for working in the Map View. Other frequently used tabs while creating map compositions are the respective raster tabs such as Multispectral and Panchromatic.

3D View



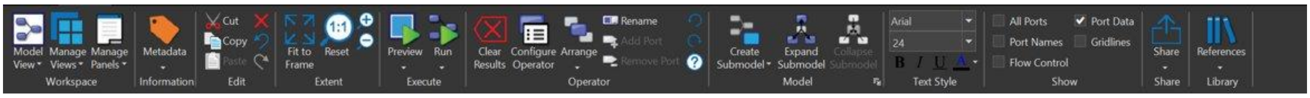
The Home tab for a 3D View contains tools for working in the 3D View that renders 3-dimensional DEMs, raster overlays and vector and/or annotation feature layers. The functions of VirtualGIS are rendered through the 3D View. The tools on the Home (3D View) tab are context sensitive, meaning that whether tools are active or what they do may depend on the current view, the current layer or the currently selected item.

Globe View



The Home (Globe View) is a new Home tab with tools specific to the 3D Preview operator in Spatial Modeler. The Home tab for a Globe View contains tools for working in the Globe View that renders 3-dimensional point cloud layers.

Model View



The Home (Model View) tab provides all the tools necessary for interacting with the Spatial Model Editor.

DHHN2016 vertical datum

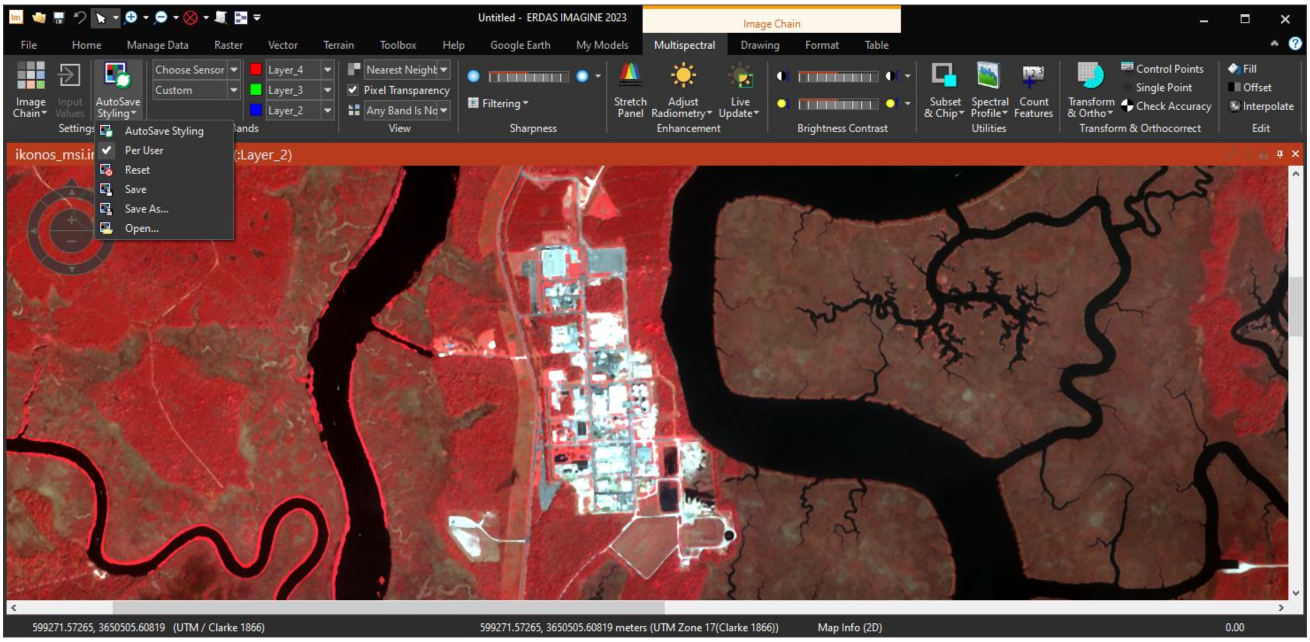
Support for the German DHHN2016 vertical datum transformation grid has been added.

Option to save Image Chain styling parameters (with Global/Per User option)

In prior releases, changes made to the styling of an image displayed using the Image Chain rendering engine (which is now the default behavior) were automatically saved so that if that image were re-opened (by the same user) the display would default to the last saved styling.

In some instances, this caused confusion, especially if the styling last applied were wrong, such as a failed attempt to improve the contrast and brightness distribution of the image.

Consequently, ERDAS IMAGINE 2023 introduces a new menu in the Settings group of the specific Image Chain's tab to enable saving modes to be configured. The user can decide whether styling should be automatically saved or not (which then requires the user to intentionally select an option to save a specific styling) and if that styling should apply just for this user or for all users of that image file. If the latter is selected, the .icp styling file is saved in the same folder as the image file and uses the same base file name, without the format extension.

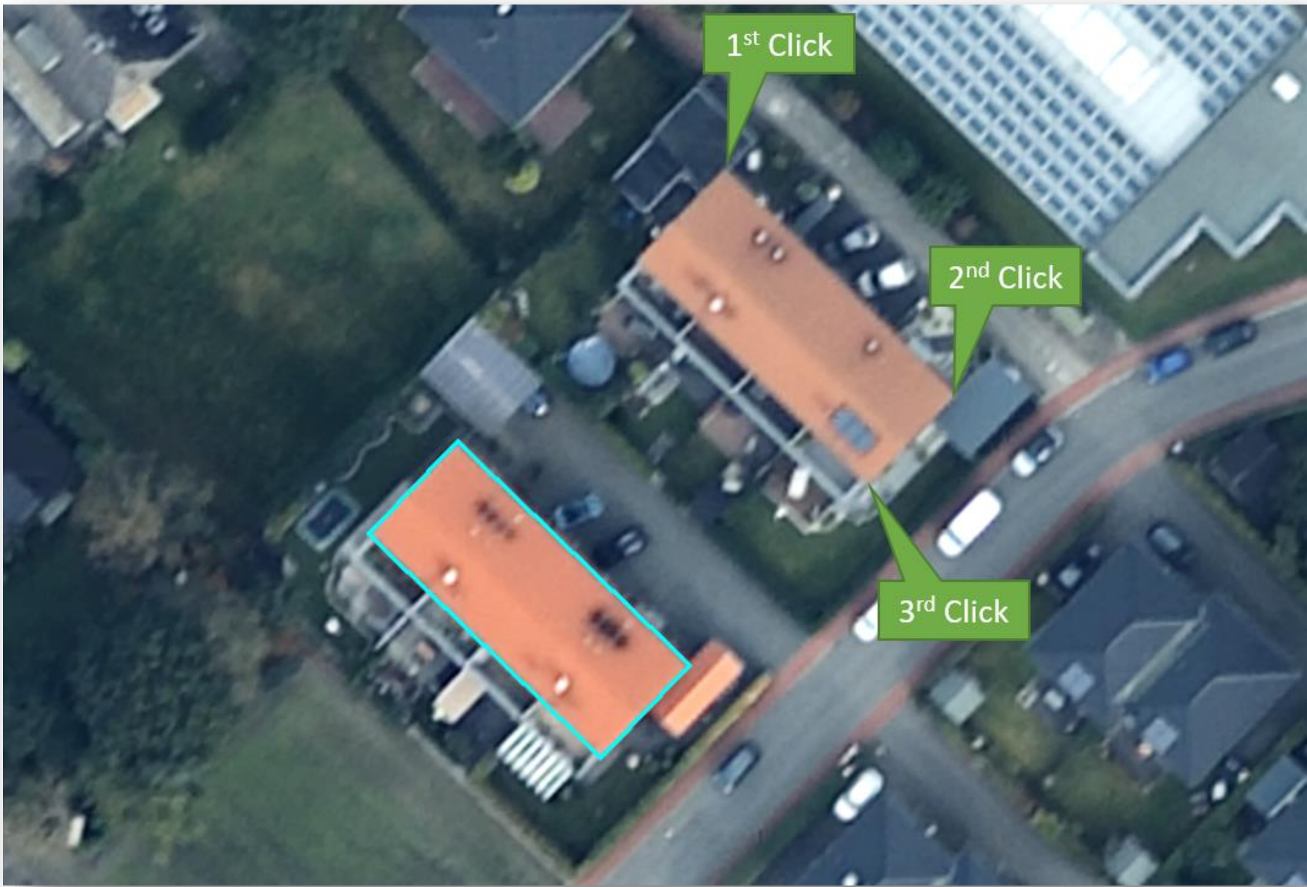


Stealth Mouse support in Stereo Analyst for IMAGINE

Support is added for Stealth Mouse V-Type and Z-Type devices. This aligns Stereo Analyst's Stealth Mouse support to the other viewplex-based stereo applications, namely Stereo Point Measurement tool (SPM), Terrain Editor(TE), ORIMA and PRO600.

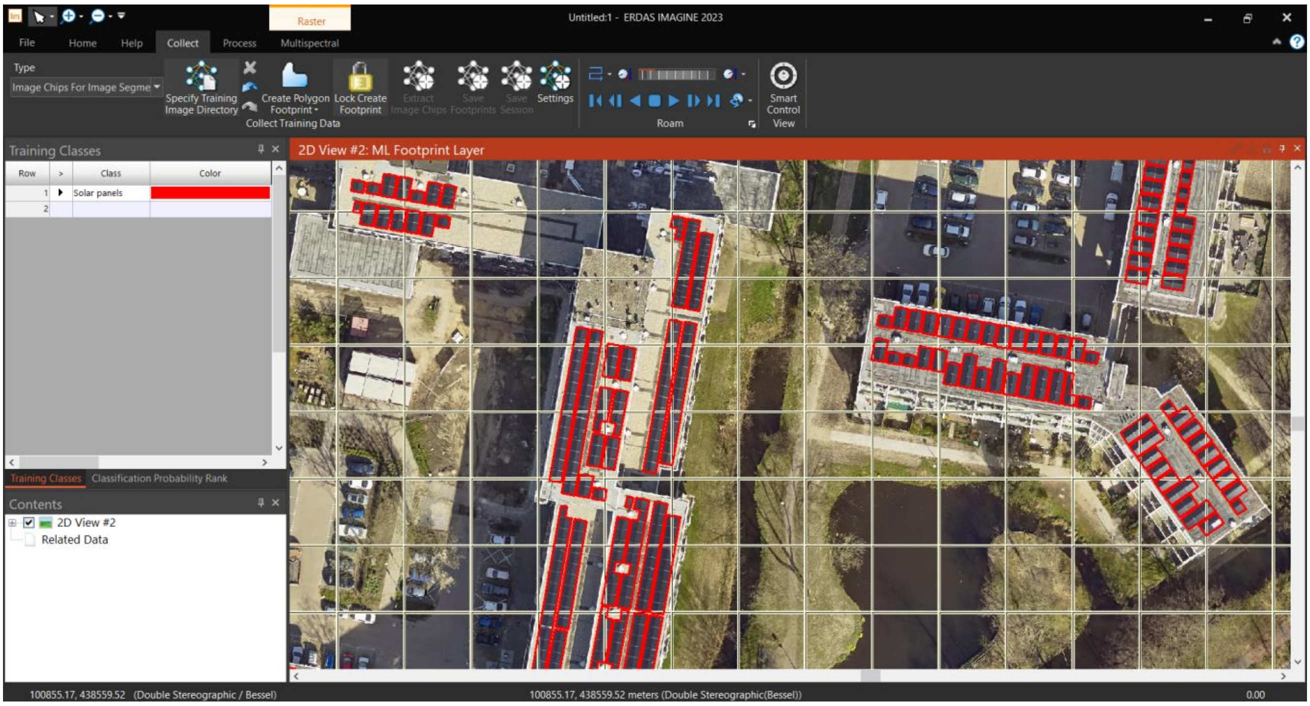
Three-Click Rotated Rectangle edit tool supported for vector layers (i.e., feature collection)

The Three-Click Rotated Rectangle digitizing tool that was introduced with ERDAS IMAGINE 2022 Update 2 has been enhanced to work with vector layers in addition to the previously supported Annotation and AOI layers.



Machine Learning Layout improvement to support Semantic Segmentation

Collection of training data for Semantic Segmentation has been added to the Machine Learning Layout. The training data for Semantic Segmentation are image chips that are labeled at the pixel level. This means each pixel of the image chip is annotated (attributed) with the class it represents. The Machine Learning Layout provides a framework for digitizing/labeling different classes and generating the image chips based on the labels.



Resample dialogs now use Spatial Modeler

Resample dialogs (e.g., Raster tab > Geometry group > Geometric Calibration menu > Orthorectify without GCPs) have been reimplemented to replace the previous executable program with a Spatial Model. This produces several benefits, including significant improvement in performance and more consistent handling of NoData.

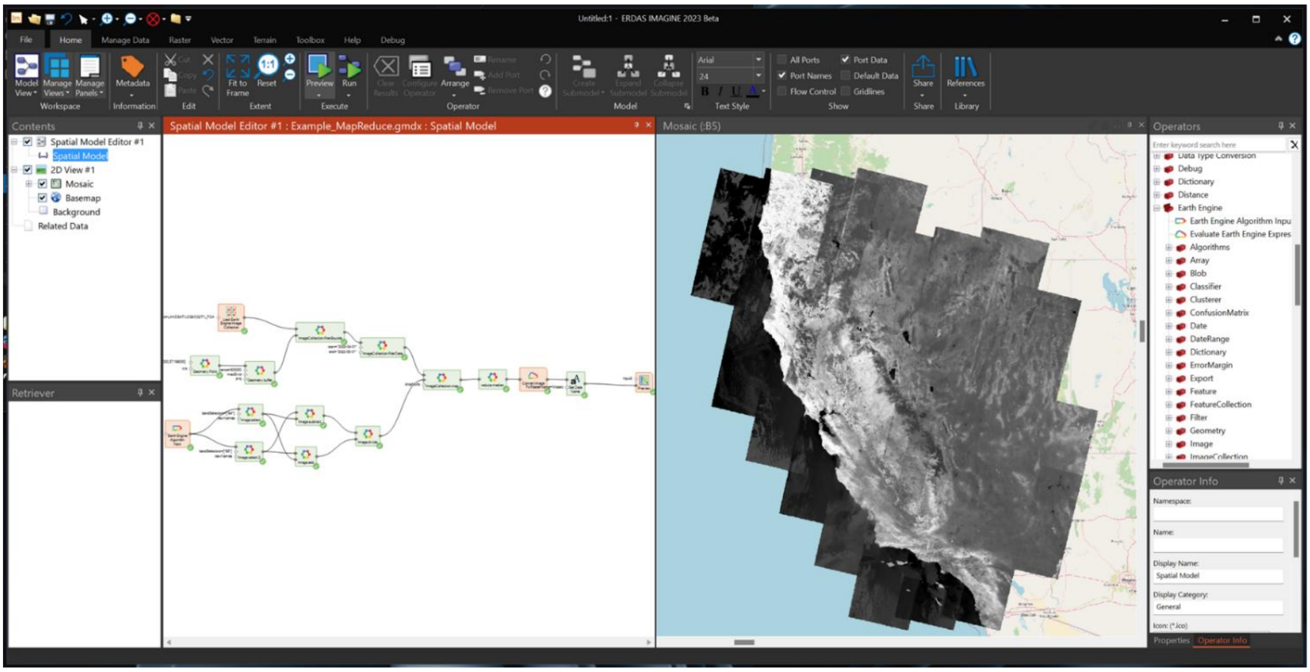
Replacement Sensor Model (RSM) refinement support in IMAGINE Photogrammetry

Support for refinement of RSM has been added. The RSM ground-to-image functions consist of one or multiple sets of rational polynomials that may have geometric errors. The geometric errors can be reduced/removed by applying polynomial corrections (refinements) to the original RSM model based on ground control points and tie points.

ERDAS IMAGINE LiveLink for Google Earth Engine

Google Earth Engine is a combination of a planetary scale collection of geospatial data and massive compute capability that can change the way people approach geospatial information analysis. The holdings include all the imagery from programs like Landsat, MODIS, Sentinel and others. This imagery spans the Earth and in many cases covers the last 40 years. Coupled with this data is a rich geoprocessing engine that puts the processing close to the data in the cloud. The combination of data and processing makes it possible to perform actions such as mapping change over time for the whole world in ways that were impractical in the past.

ERDAS IMAGINE LiveLink for Google Earth Engine is a new product available for use with ERDAS IMAGINE that pairs the intuitive, easy-to-use graphical modeling environment of ERDAS IMAGINE's Spatial Model Editor with the vast data holdings and online geoprocessing capabilities of Google Earth Engine.

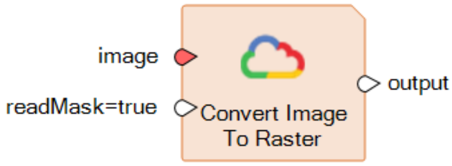


The screenshot above shows a spatial-temporal query being used to select all the images within a 600-km radius of San Francisco in 2022, the NDVI computed for each, and all combined into a single output. Just locating and acquiring all this data would have been a large, time-consuming task in the past.

LiveLink for Google Earth Engine introduces 12 new operators that provide the interface with the Google Earth Engine Cloud services. In addition to these 12, an additional 850 or so Earth Engine operators arranged in about a dozen categories are created dynamically within Spatial Modeler using Google Earth Engine's REST API. These operators can be assembled like any other Spatial Modeler operator to create a model that runs on Google's cloud infrastructure.

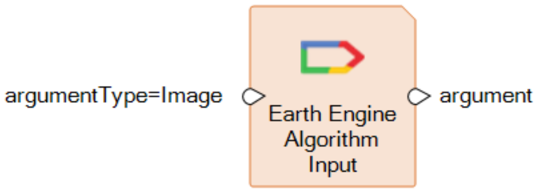
New interface Operators provided by LiveLink for Google Earth Engine

Convert Image To Raster



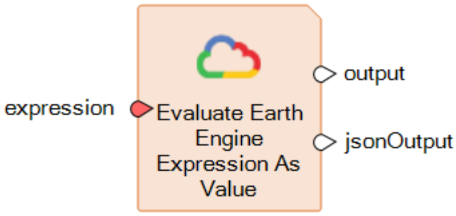
This operator uses the Earth Engine Image expression that is connected to the image port to create an IMAGINE.Raster on the output port. This enables raster tiles to be read by any of the Spatial Modeler operators that take an IMAGINE.Raster as an input. All the computation defined by the input Image expression is performed on the Google cloud infrastructure. Computation is only performed for the requested tiles at the requested resolution.

Earth Engine Algorithm Input



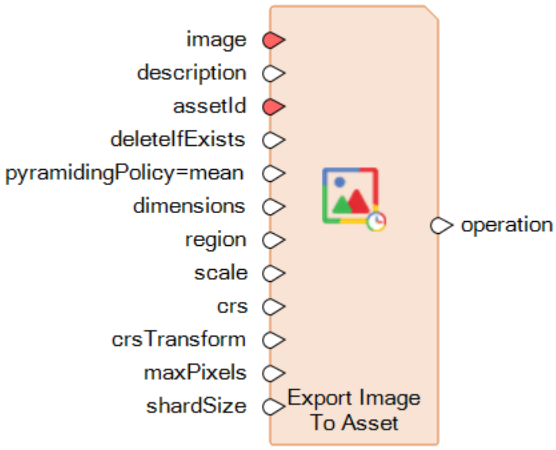
This operator is a section of the Spatial Model built from Earth Engine operators that is to be used as an input to an operator such as ImageCollection.map. The Algorithm must begin with an Earth Engine Algorithm Input that defines the number and order of inputs to be used.

Evaluate Earth Engine Expression As Value



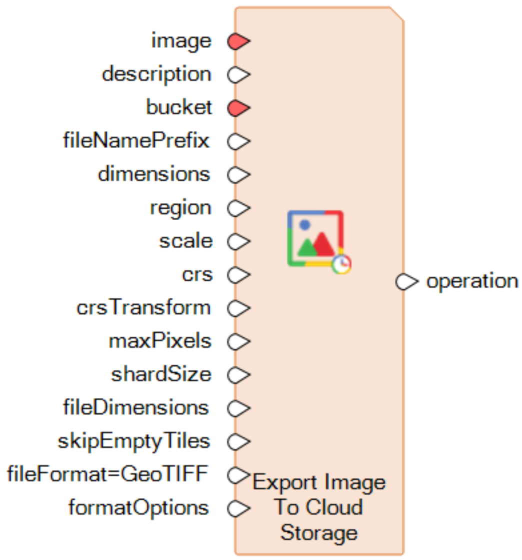
This operator evaluates the Earth Engine Expression and returns its value as the appropriate type. For example, an expression that results in an integer will return an IMAGINE.Int on the **output** port, a double will return an IMAGINE.Double and a string will return an IMAGINE.String. Expressions evaluating non-value types such as an Earth Engine Image Collection or an Earth Engine Image return metadata about the expression as an IMAGINE.Dictionary. The JSON value returned from the expression evaluation can be viewed on the **jsonOutput** port. This can be useful during the development of the model to understand the results of an operation.

Export Image to Asset



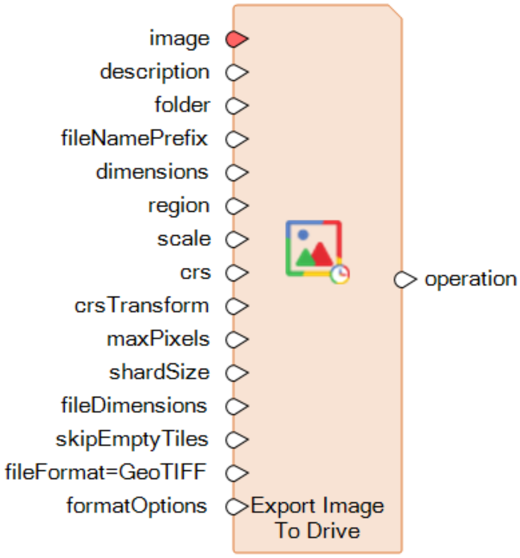
This operator lets you create a task to evaluate an Earth Engine Image expression and save the results as an asset in the Earth Engine Project space. Assets in the Project space can be directly used in further Earth Engine computations. The resulting asset ID is visible in the Project folder in the Earth Engine Catalog Browser.

Export Image to Cloud Storage



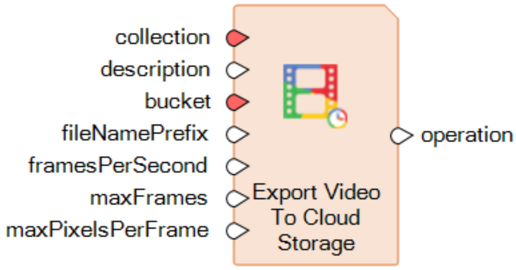
This operator lets you create a task to evaluate an Earth Engine Image expression and save it as one or more files in the named bucket in Google Cloud. The account to which the bucket belongs must have given access to the Earth Engine project being used.

Export Image to Drive



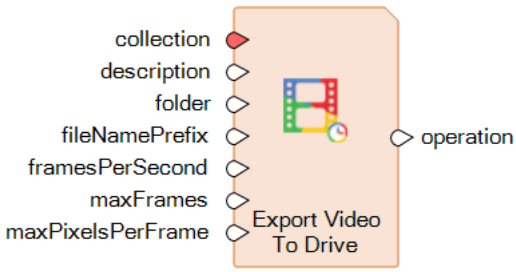
This operator lets you create a task to evaluate an Earth Engine Image expression and save it as one or more files in the named folder in Google Drive. The account folder must be accessible to the Earth Engine project being used.

Export Video to Cloud Storage



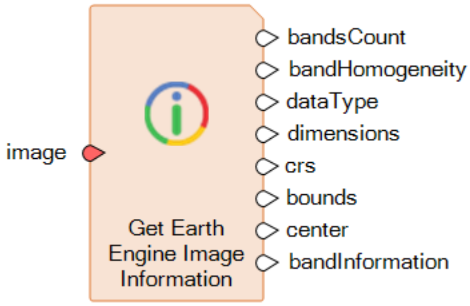
This operator lets you create and start a task to evaluate and export an Earth Engine Image Collection expression as an MP4 formatted video file saved to Google Cloud storage. The collection must only contain RGB images in the final CRS and resolution. See Creating Videos From Image Collections for details.

Export Video To Drive



This operator lets you create and start a task to evaluate and export an Earth Engine Image Collection expression as an MP4 formatted video file saved to Google Drive storage. The collection must only contain RGB images in the final CRS and resolution. See Creating Videos From Image Collections for details.

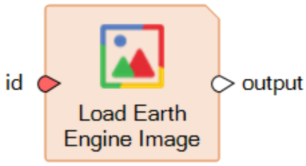
Get Earth Engine Image Information



This operator evaluates the Earth Engine Image expression given on the **image** port and returns several pieces of useful information about the first band of the resulting image. Earth Engine images are not constrained to have homogenous bands. The outputs **dataType**, **dimensions**, **crs**, **bounds** and **center** are presented for the first band only, given that typically the bands are homogenous; however, the full set of metadata is returned on the **bandInformation** port.

This operator is very useful as a debugging tool during the development of a model using the Earth Engine image operators.

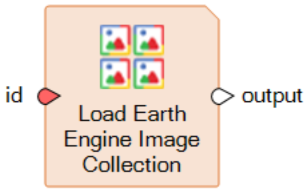
Load Earth Engine Image



This operator selects a single image from the Earth Engine catalog of data and makes it available as an EarthEngine.Image for use by any of the Earth Engine operators that take an EarthEngine.Image as an input.

The operator can be configured using the Earth Engine Catalog Browser to select an image asset ID.

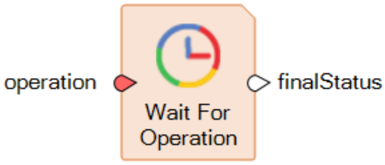
Load Earth Engine Image Collection



This operator selects an image collection from the Earth Engine catalog of data and makes it available as an EarthEngine.ImageCollection for use by any of the Earth Engine operators that take an EarthEngine.ImageCollection as an input.

The operator can be configured using the Earth Engine Catalog Browser to select an image collection asset ID.

Wait For Operation



This operator waits for the batch operation identified by the input to complete, and upon completion the final status of the operation is made available.

Issues resolved: ERDAS IMAGINE 2023

IMAGINE Essentials

Support ticket	Summary
00137306	Problems opening and displaying some COSMO-Skymed Second Generation SAR images has been corrected.
00140747	ERDAS IMAGINE would crash if 'Breakpoint Editor' was selected in the Help tab's 'Search Commands window' after a raster image is displayed with the 'Image Chain' rendering engine.
00091311	When displaying multi-band imagery using the legacy Raster rendering engine and attempting to use and save changes made using the Data Scaling tool, ERDAS IMAGINE would sometimes crash.

IMAGINE Advantage

Support ticket	Summary
00165651	Export Raster Product Format (RPF) was not writing Significant, Production and Currency Date attributes to the CADRG headers correctly.
00022850	Occasionally, with some datasets, MosaicPro was introducing small gaps in seam areas between overlapping image frames. This has been corrected and should no longer occur.
00106612	In previous versions of ERDAS IMAGINE it was not possible to drag the MosaicPro application window from the primary monitor to a secondary (or other) display.
00091332	A customers specific set of imagery could not be successfully processed using MosaicPro. This has been corrected.
00080609	MosaicPro was crashing during seamline generation when a specific TIFF file was sent to the top of the stacking order. The crash did not occur if this TIFF image was not moved to the top of the stacking order. The problem did not happen if you imported the TIFF file to IMG format and replaced it in the photogrammetric project. This has been corrected.
00059579	Certain customer data, loaded in a specific order into MosaicPro could cause a crash when mosaicking. This has been corrected.
00023137	When mosaicking specific panchromatic images in JPEG 2000 format and writing the output to JPEG 2000 format the mosaicked image might contain several black 512 pixel x 512 pixel squares. The black squares had a pixel value equal to 1. The problem did not happen when writing the output to IMG or ECW format.
00091994	Attempting to orthorectify some NITF images to a GeoTIFF output file would fail with an "Invalid arguments" message. This has been addressed by moving the resampling processes to use Spatial Modeler.

IMAGINE Expansion Pack

Support ticket	Summary
00023261	When resampling data in AutoSync the map and projection information from the input raster wasn't being written directly to the JPEG 2000 format output image. Spatial information was written to the associated .aux file instead. This could be an issue when the JP2 raster is used outside of the ERDAS IMAGINE environment (e.g. ArcGIS). This is related to IM-55223: "Map coordinate and projection information isn't written to output JPEG 2000 image," which fixed the problem in other areas of ERDAS IMAGINE (e.g. MosaicPro, Subset, etc.) but the issue remained when outputting JPEG 2000 data from AutoSync.

IMAGINE SAR Interferometry

Support ticket	Summary
00099763	Documentation issues relating to the Differential InSAR input step have been corrected.

IMAGINE SAR Feature

Support ticket	Summary
00108773	The Radar Image Chain now correctly disallows changing the resampling method, while other Image Chains (such as Panchromatic) allow the resampling method to be selected and correctly applied to SAR imagery.

IMAGINE Photogrammetry

Support ticket	Summary
00065305	With the introduction of IMAGINE Photogrammetry 2022 Update 1 contrast adjustment had the effect of blacking out imagery with a dynamic range greater than 8-bit. This occurred when using Photogrammetry > Classic Point Measurement > (Right click) > General Contrast.
00151158	When an ISAT Project was imported into an IMAGINE Photogrammetry project the Exterior Orientation angle units were incorrectly translated and were always changed from angular degrees to radians.
00154591	The button mapping file was missing for the 3D Softmouse input device.

IMAGINE DSM Extractor

Support ticket	Summary
00063859	The IMAGINE DSM Extractor process could fail to execute correctly using some specific datasets.
00048344	When using input images with irregular boundaries (e.g. study-area AOIs) the DSM extraction process could sometimes fail to complete.

IMAGINE Professional

Support ticket	Summary
00023382	The Grid Generation Tool was producing an empty shapefile on systems where the Japanese system locale was set.
00023228	The Machine Learning Layout was failing to load Object Footprints With Image Chips

ERDAS ER Mapper

Support ticket	Summary
00034079	ERDAS ER Mapper was failing to write output file from an algorithm if the target format was JPEG 2000
00023194	ERDAS ER Mapper was writing the wrong datum and projection for GDA2020

IMAGINE Objective

Support ticket	Summary
00023502	Occasionally IMAGINE Objective might crash after selecting ‘Accept’ to save training in projects with multiple Raster Pixel Processor cue metrics (e.g. SFP, Identity) using different project raster variables.
00023501	IMAGINE Objective project would generate unexpected ROC Segmentation results from RPP Identity metric if the input data were Float (rather than u8 or u16.)
00023236	The Raster Pixel Processor pixel cue probability layer required for the ROC “Line Threshold and Clump” plug-in layer could not be generated in IMAGINE Objective. Consequently “Line Threshold and Clump” has been removed from the software to avoid confusion.
00023238	The IMAGINE Objective VOP cue would sometimes not save projection parameters to the output shapefile.

Spatial Modeler

Support ticket	Summary
00084429	Creating image chips with the Create Dice Boundaries operator in combination with Define Processing Area and Iterator operators would sometimes fail. It was correlated to the number of image chips being produced in parallel. When using 512x512 pixels for each chip it worked. But choosing 224x224 pixels it failed. Then ERDAS IMAGINE might crash if you attempt to close the Spatial Model Editor by clicking the X in the upper-right corner. The process should no longer cause a crash and should complete successfully in most cases.
00115281	The Raster Statistics per Feature operator could, under certain circumstances, return incorrect (frequently 0) ComputeSum statistics.
00086755 00100698	With some input datasets the Convert To Features operator could cause the error “Trying to change a running meter from modal to non-modal is not supported!” to be reported. After closing the error window ERDAS IMAGINE would remain frozen and needed to be restarted, but the output feature data was created correctly.
00086713	The Convert to Raster operator could sometimes fail to convert some large polygons or islands associated with multi-polygons (resulting in NoData pixels instead.)
00130377	When producing multiple Features sets inside an Iterator and passing that List of Features to an operator such as Union Features a crash could occur.
00132145	The Initialize Image Segmenter operator was failing to run successfully if raster with more than 3 bands was passed in.

00112336	The Add Attributes By Location operator could return empty attribute fields when working with Geopackage vector data.
00090607	Deep Learning Object Detection was creating much worse result from classification if using training data based on using the "Image Chips With Footprints" method compared to "Footprints only."

About Hexagon

Hexagon is the global leader in digital reality solutions, combining sensor, software and autonomous technologies. We are putting data to work to boost efficiency, productivity, quality and safety across industrial, manufacturing, infrastructure, public sector, and mobility applications.

Our technologies are shaping production and people-related ecosystems to become increasingly connected and autonomous – ensuring a scalable, sustainable future.

Hexagon's Safety, Infrastructure & Geospatial division improves the resilience and sustainability of the world's critical services and infrastructure. Our solutions turn complex data about people, places and assets into meaningful information and capabilities for better, faster decision-making in public safety, utilities, defense, transportation, government and physical security.

Hexagon (Nasdaq Stockholm: HEXA B) has approximately 24,500 employees in 50 countries and net sales of approximately 5.4bn EUR. Learn more at hexagon.com and follow us @HexagonAB.

Copyright

Terms of Use

Disclaimers

Additionally, some text, graphics, PDF documents, and/or other accompanying material may not have been translated.

Links To Third Party Websites

This Document may provide links to third party websites for your convenience and information. Third party websites will be governed by their own terms and conditions. Hexagon does not endorse companies or products to which it links.

Third party websites are owned and operated by independent parties over which Hexagon has no control. Hexagon shall not have any liability resulting from your use of the third party website. Any link you make to or from the third party website will be at your own risk and any information you share with the third party website will be subject to the terms of the third party website, including those relating to confidentiality, data privacy, and security.

Hexagon provides access to Hexagon international data and, therefore, may contain references or cross references to Hexagon products, programs and services that are not announced in your country. These references do not imply that Hexagon intends to announce such products, programs or services in your country.

Revisions

Hexagon reserves the right to revise these Terms at any time. You are responsible for regularly reviewing these Terms. Your continued use of this Document after the effective date of such changes constitutes your acceptance of and agreement to such changes.

Questions

Contact us with any questions regarding these Terms.