ESRI
EXTERNAL

Oriented Imagery Catalog Management

User Documentation v2.3 | May 11, 2020

380 New York Street
Redlands, California 92373-8100 USA
909 793 2853
info@esri.com
esri.com

esri | THE SCIENCE OF WHERE™

This document applies to Oriented Imagery, currently provided as a prototype and for testing only. The functionality has not been exhaustively tested and is not currently covered under ArcGIS Support. Please address questions or suggestions related to this workflow to GeoNet or to ImageManagementWorkflows@esri.com

Contents

Overview

Introduction

This document provides guidance on managing imagery collections for use with Oriented Imagery.

Oriented Imagery is an image management and visualization solution for non-nadir optical imagery, such as oblique, street-view, or inspection images. This imagery can be gathered aerially, by drones, or by terrestrial sensors such as mobile cameras.

Oriented Imagery users can interact with such imagery using the Oriented Imagery ArcGIS Pro add-in, or using an app built with the Oriented Imagery Web AppBuilder widget. This example Oriented Imagery app demonstrates how an end user would experience Oriented Imagery.

Before imagery can be viewed using ArcGIS Pro or an Oriented Imagery app, the imagery collection must be managed using a data model called an oriented imagery catalog (OIC). An oriented imagery catalog is defined as an OIC item, published in ArcGIS Online or your ArcGIS Enterprise portal, that references a camera exposure point feature service and optionally a coverage feature service. The OIC item defines a set of attributes, defaults, and filters used to visualize web-accessible oriented images, usually from cloud storage.

To create OICs for use with Oriented Imagery, you'll use the geoprocessing tools to do the following:

1. Create an Oriented Imagery Catalog (OIC) from collections of imagery
2. Publish an OIC to ArcGIS Online so that it can be used by the Oriented Imagery Add-In for ArcGIS Pro or Web AppBuilder applications using the Oriented Imagery Widget

Once published, an OIC can be consumed by the Oriented Imagery Add-In for ArcGIS Pro or by Web AppBuilder applications using the Oriented Imagery Widget.

Oriented Imagery can represent many different sources of imagery from many different optical cameras. Typical users have collections of imagery captured using mobile cameras or drones. Common imagery sources are supported out-of-the-box, or users can create custom oriented imagery types to support a wide range of cameras.

For commercial companies who generate such content at scale and wish to create custom or automated publication workflows, information about the database schemas used by Oriented Imagery are available at https://github.com/esri/oriented-imagery.

New in Version 2.3

- Bug fixes for the **Frame Table** and **Image Service** oriented imagery types
- Better error handling during publication
- Bug fixes to **Calculate Heading** and **Create Coverage Features** tools

Requirements

- Oriented Imagery add-in for ArcGIS Pro must be installed.
- Oriented Imagery and associated management tools are compatible with ArcGIS Pro 2.2+.
- Imagery must be in web-accessible storage, such as Amazon S3. If it's not already, the workflow includes tools to copy your imagery from local storage to web-accessible storage. Alternatively, you can publish your imagery to ArcGIS Online or your ArcGIS Enterprise portal as a feature service with attachments and use that to create an OIC.

- Metadata about the location and orientation of the imagery must be included in the EXIF headers of the JPEG files or in an external table.

Support

Oriented Imagery and these tools are currently released as Early Adopter. The tools work well with many types of imagery, but there may be issues with some types of imagery. More development and testing are required to fully stabilize the application and ensure aspects such as accuracy are clearly defined.

Feedback about the oriented imagery viewers and oriented imagery management tools are requested. For providing feedback as well as obtaining support please use the following two GeoNet sites:

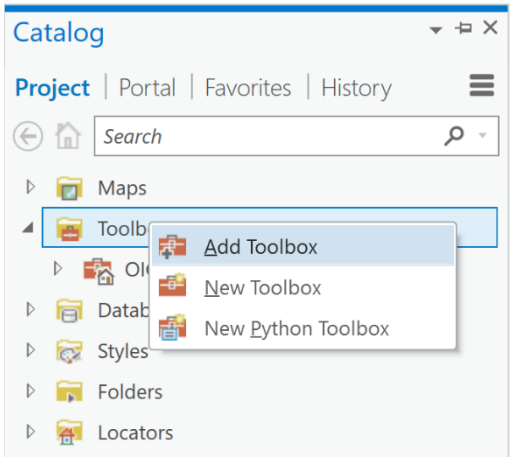
- Using Oriented Imagery (Add-In or Widget): http://esriurl.com/UsingOrientedImagerySupport
- Oriented Imagery Management Tools: http://esriurl.com/ManagingOrientedImagerySupport

Installation

To get started with the oriented imagery management tools, download and install the toolbox and add the toolbox to a project in ArcGIS Pro 2.2+.

Note: Oriented Imagery add-in for ArcGIS Pro must be installed.

1. If you haven't already, download and unzip the tools.
2. Double click on the setup file and click **Install**. This will create the folder C:\image_mgmt_workflows\OrientedImagery, which contains the geoprocessing tools required to create an oriented imagery catalog.
3. To start creating an OIC, add the geoprocessing tools to your ArcGIS Pro project:
    a. Create a new project or open an existing project.
    b. Right click on **Toolbox** and select **Add Toolbox**.



    c. Navigate to and select C:\Image_Mgmt_Workflows\OrientedImagery\GPTool.
    d. Select ManageOrientedImagery.pyt and click **OK**.

Manage Oriented Imagery Toolbox

The toolbox consists of selection of tools create and manage oriented imagery catalogs (OIC).

Below is the sequence of the most common tools used to create and manage your oriented imagery, as well as a reference for all tools.

Following are details about how to use each Oriented Imagery management tool to create or manage an oriented imagery catalog (OIC). This document will guide you in the sequence of tools to be used to create and manage your oriented imagery.

The tools in the standard workflow for creating an OIC include:

1. Create Oriented Imagery Catalog
2. Add Images to Oriented Imagery Catalog
3. Create Coverage Features
4. Create Coverage Map
5. Copy Images to Web
6. Analyze Oriented Imagery Catalog
7. Publish Oriented Imagery Catalog

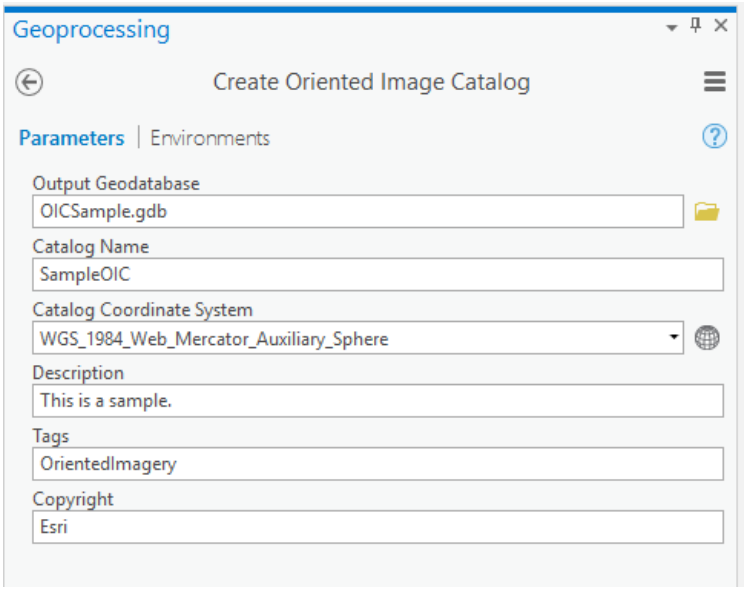
Additional tools to manage OICs include:

- Create Reference Oriented Imagery Catalog
- OIC Properties
- Repair Paths
- Select Broken Paths
- Select Local Images
- Set Exposure Station ID
- Calculate Heading

Below find a complete description of each tool and the parameters required.

Create Oriented Imagery Catalog (Required)

This tool will create a new OIC in a specified location.



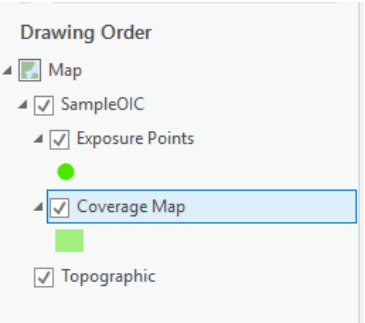
You will be prompted for:

- The name of the output geodatabase that will store the feature exposure points (required).

- The name for the oriented image catalog (required). The name given here will be used for the resulting OIC file and the exposure point feature class in the geodatabase.
- The catalog coordinate system. This will be used to define the exposure locations. In many cases, Web Mercator Auxiliary sphere can be selected, but if all the data is for a local area and you standardize using a specified coordinate system then this can be used.

  Note that the location information from the sensor may be in a different coordinate system and will be converted to the one you select, but the orientation angles will not be converted. For most coordinate systems, this is not an issue.
- A brief description of the imagery (optional).
- Tags (required).
- Copyright notices (optional). The copyright, if defined, will be used in the oriented imagery catalog item and will also be displayed on the imagery.

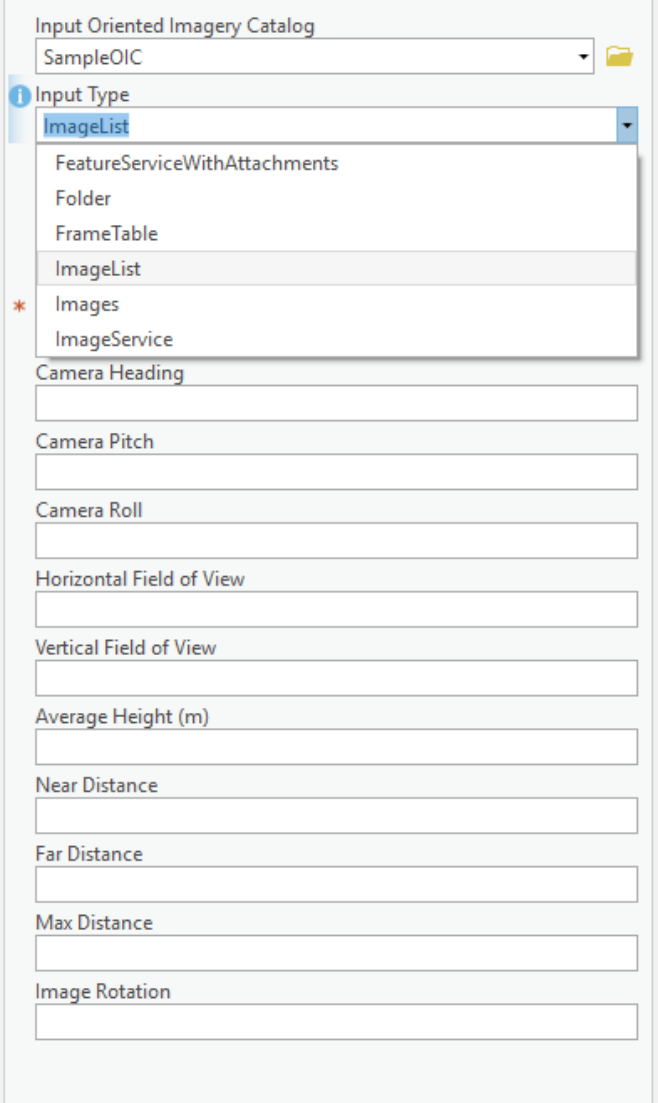
Once the dialog is filled in, click **Run** to begin the process of creating the OIC. Once complete, a group layer with the catalog name is added to your Contents pane.



Initially, these layers will be blank; use the **Add Images to Oriented Imagery Catalog** and **Create Coverage Features** tools to populate them.

Add Images to Oriented Imagery Catalog (Required)

This tool offers the user different methods of adding source images to the oriented image catalog, depending on the data source. Once images are added, points will be added to the Exposure Points feature class to indicate the locations the images were taken and to manage metadata.



Coverage of the images is determined based on metadata associated with each image. To determine the coverage of the images accurately requires:

- The coordinates of the exposure points
- The orientation and focal length of the camera
- The relative location of the ground
- Some additional auxiliary information

Oriented Imagery uses a schema that defines such information in two different ways.

1. Method One defines the camera location and angles such that the system can determine a frustum that defines what can be seen. This method is very generic but may not be accurate. It is used to aid identification of appropriate imagery for display and navigation between the images.
2. Method Two is optional and much more complex. It defines the location and angles of the camera as well as parameters such as the camera distortions. This method is used for imagery that has been captured with accurate metadata which, if available, will be stored as part of the

oriented imagery catalog. This enables more advanced capabilities, such as mensuration to allow the measurement of height and distances. Details of this second method are not defined here, but are available in the developer documentation.

The key oriented imagery parameters define a frustum that represents the coverage of the image in 3D. If the suitable values to determine these parameters are part of the input then they will be read, else they can be defined as default values and will be applied to the imagery. Later, the coverage of the imagery can be reviewed and the parameters changed if required.

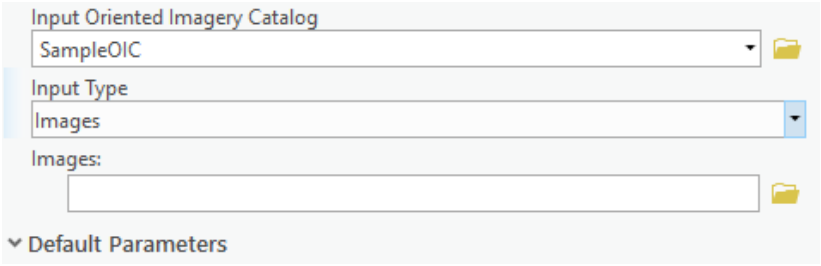
The Add Image to Oriented Imagery Catalog Tool will attempt to determine the key parameters from the source, but if the key parameters are not available, the tool will revert to using the default values.

Parameters

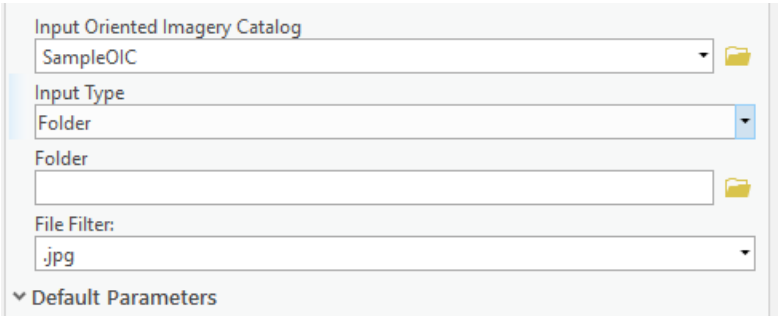
Following is a brief description of the parameters and their use. For more details see the developer documentation.

- **Input Type.** There are many different ways that the imagery may be stored and how the required attributes are defined. The Add Images to Oriented Imagery Catalog tool provides six different methods for adding images to the OIC—images, folder, frame table CSV, image list, image service, feature service with attachments, and custom. These are defined below:

    1. **Images.** This option should be used when adding only a few or selected set of images stored locally. Use the folder icon to navigate to a folder and select one or more images.



    2. **Folder.** This option should be used when adding a folder containing imagery. The folder may contain a large number of images. Use the folder icon to navigate to and select a folder. A file filter can be defined to filter the images. Currently, JPG and TIF formats are supported. Images stored in sub-folders are also added.



    3. **FrameTable.** This option should be used if the input is defined as a comma delimited (CSV) file that defines properties of the imagery. It is typically used when imagery is collected using a camera and separate GPS and Inertial Measurement Unit (IMU) that determines accurate location and orientation of the imagery. The location of the table for input needs to be defined.

Note: When you use this tool, the EXIF data from the images is not read. If required, you can create a separate OIC using the Image or Folder options, then copy the attributes to the OIC created using Frame Table.

GPS and IMU units will not generate data in the required format. Prior to using this option, it is necessary to use or develop additional programs to format the data appropriately.

The table needs to be comma delimited and have the following fields:

- **X.** X coordinate in same projection as oriented image catalog
- **Y.** Y coordinate in same projection as oriented image catalog
- **SRS.** WKID (or WKT string) of the points in the table.
- **Image**. Location of the image. May be a local file or web accessible URL
- **Name**. (Optional) Name given to the raster; may be displayed for identification purposes.
- **AcquisitionDate**. (Optional) Date field. Defines date and optionally time that the image was collected.
- **CamHeading**. (Optional) Camera Heading in degrees. 0 is north measured clockwise.
- **CamPitch**. (Optional) Camera Pitch in degrees. 90 is horizonal and 0 is vertical
- **CamRoll**. (Optional) Camera Roll in degrees. 0 is not rotated. Angles are clockwise.
- **HFOV**. (Optional) Horizontal field of view of the camera in degrees
- **VFOV**. (Optional) Vertical field of view of the camera in degrees
- **AvgHtAG**. (Optional) Average Height (m) above ground of the camera
- **NearDist**. (Optional) Nearest usable distance of the imagery in meters. What is the closest point that can be seen?
- **FarDist**. (Optional) Furthest usable distance of the imagery. Locations beyond this distance are of no interest.
- **Omega**. (Optional) Rotation around the X axis. If used, there must be corresponding Phi and Kappa fields.
- **Phi**. (Optional) Rotation around the Y axis. If used, there must be corresponding Omega and Kappa fields.

- **Kappa**. (Optional) Rotation around the Z axis. If used, there must be corresponding Omega and Phi fields.
- **OIType**. (Optional) A pipe delimited string that can be used to defined accurate image location and oriented details. See developer documentation for more details.

NOTE: Optional fields are not required in this table, but many are required values for the OIC. If you specify an OIType, it will define appropriate default values for many of the OIC fields. See OIC Properties for more info.

Also note that if you include Omega, Phi, Kappa, it will override any values for CamHeading, CamPitch, CamRoll.

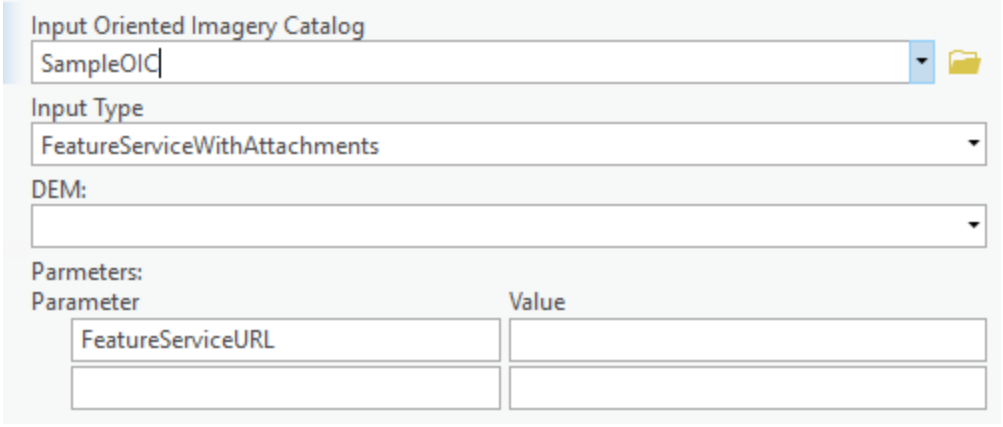
Additional fields not listed here can be defined. These are documented in detail as part of the developer documentation.

4. **Image list.** This option should be used when adding images that are stored on web-accessible storage. The input will be an image list, which is a simple text file with no header that contains a list of URLs/paths to the images to be added. If you're using URLs, this should be accessible without credentials. For example:

   https://sample.s3.amazonaws.com/redlandcampus/IMG_1234.JPG
   https://sample.s3.amazonaws.com/redlandcampus/IMG_5678.JPG
   https://sample.s3.amazonaws.com/redlandcampus/IMG_9012.JPG
   https://sample.s3.amazonaws.com/redlandcampus/IMG_3456.JPG

   If you're using simple authentication then the URL may contain additional keys.

5. **FeatureServiceWithAttachments.** Uses a URL to an existing feature service with attachments as input. It reads a feature service layer and extracts the URLs of the stored attachments to make them accessible to the OIC.



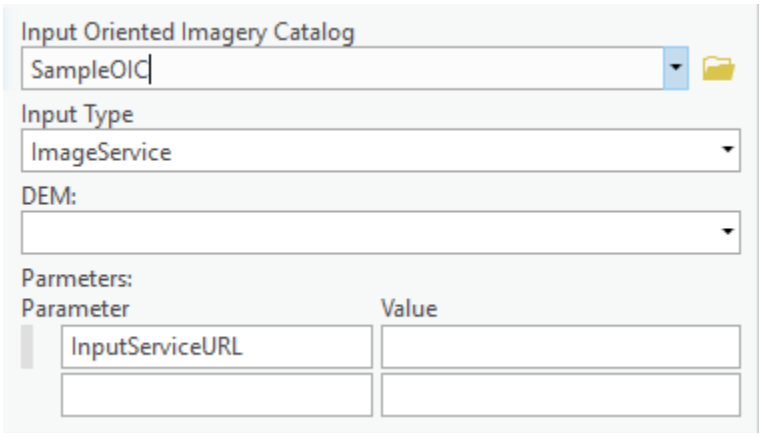
It requires the following parameters:

- **FeatureServiceURL**: The feature service URL. A typical acceptable format is: https://sample.arcgis.com/<itemID>/arcgis/rest/services/<serviceName>/FeatureServer/<layerID>

  NOTE: pointing to just the hosted feature service would result in an error. The individual layer must be specified using the layerID.

6. **Image service.** This option should be used to add exposure points from an image service. Image services created using aerial imagery raster types are well-suited. A typical image service URL should be in this format:

   https://<server_name> /arcgis/rest/services/<folder>/<service_name>/ImageServer



7. **Custom Input Type.** The Add Images to Oriented Imagery Catalog tool is designed to be extensible, providing a way to add imagery from different sensors. Each input method is contained within a Python script that defines input parameters. These are then presented to the user as options. Users proficient in Python can write their own input type.

   Additional custom types can be created for different sensors by Esri, by vendors, or by developers. Developer Brief: Creating Custom User Types is a quick introduction to

creating custom user types. Custom input types provided by Esri can be downloaded from GitHub.

- **CamHeading.** Camera Heading in degrees. 0 is north measured clockwise. Valid values are from 0 to 360 as well as -1. Many cameras including iphones capture this information. If undefined the value of -1 should be used. This indicates that no heading information is available and the system will handle the image as a geotagged image. IE the position of the camera is known, but no orientation information.
- **CamPitch.** Camera Pitch in degrees. 90 is horizonal and 0 is vertical. For most terrestrial image this value is likely to be around 90 and is a suitable default. For oblique imagery a value of around 45 degree is common. For nadir imagery (looking straight down) use 0. The valid range is 0 to 180. For values <0 it is better to rotate the heading by 180 and camera roll by 180, which has a similar effect
- **CamRoll.** Camera Roll in degrees. 0 is not rotated. Angles are clockwise. Most image are taken with a Roll of 0. IE the top of the image is parallel with the horizon.
- **HFOV.** Horizontal field of view of the camera n degrees. Defines the angle that the camera can see in the horizontal direction. 60 degrees is typical for many camera. Cameras with longer focal lengths may have smaller angles. Fisheye lenses can reach to about 180degrees. Panoramic image often have HFOV value >180. Bubble or 360 imagery would have a HOV value of 360.
- **VFOV.** Vertical field of view of the camera in degrees. Similar to HOV but in the vertical direction. This angle is typically smaller. Many cameras have a value of around 40degrees. The factor of HFOV/VFOV is often referred to as the aspect ratio of the camera. An traditional 35mm film camera would have had an aspect ratio of 3:2.
- **Average Height.** Average Height (m) above ground of the camera. This is required as an estimate of how high the camera is above the ground. It is used to estimate how the image would interest with the ground if no other height information is available.
- **Near Distance.** Nearest usable distance of the imagery in meters. This is used to define how close object of interest can be to the camera location. For a terrestrial image this may be very close. For oblique aerial image this may be much further away.
- **Far Distance.** Furthest usable distance of the imagery. Locations beyond this distance are of no interest. Again this is used to determine the suitable search distance for the image. A terrestrial image may include feature such as hills in the distance, but they are not considered features of interest. This parameters can be set so that only feature that can suitably identified or measured are included in search results when looking for imagery of a specified location.

  Imagery Type. Sets the default parameters used to load your imagery into the OIC. Each Imagery Type identifies the platform category and the camera category—descriptions of these categories are below. If you've defined a custom OI Type, it will also appear in this list.

  Based on your choice, approximate parameters will be set by default.

  Platform categories:
  - **Aerial**: Images taken from an aircraft or plane.
  - **Drone**: Images from small, remote-controlled quadcopter aircrafts. They differ from aerial imagery by flight height and resolution.
  - **Inspection**: Close-up (macro) imagery of assets.

- **Terrestrial**: Typically, images taken outdoors that assumes objects of interest are farther away than Inspection.
- **Indoor**: Typically used for images taken indoors. Heading and field of view values may have to be filled in.
- **GeoTagged (No Orientation)**: Images that only have a GPS position with no heading information.

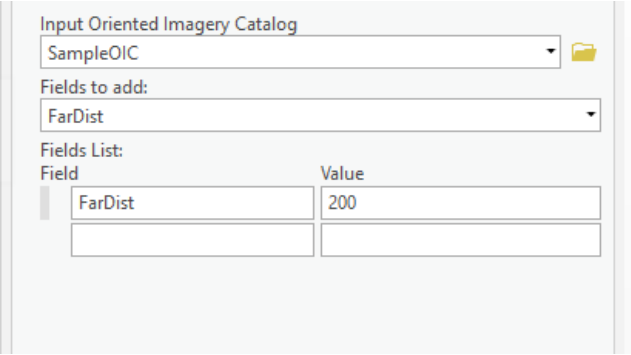
Camera categories:

- **360 Camera**: Images taken using specialized cameras that provide 360 spherical surround views or have been stitched together as 360 view from multiple cameras.
- **Nadir Camera**: Images where the exposure is perpendicular to the ground and looking straight down and only the tops of objects are seen.
- **Oblique Camera**: Images where the exposure is at an angle to the ground, typically at about 45 degrees so sides of objects can be seen.
- **Video**: Georeferenced videos. These are videos taken by cameras typically mounted on drones, cars or other vehicles and have a field of view typically of about 90 to 120 degrees. A table of camera location and orientation for different points in time is required.
- **360 Video**: Georeferenced videos with a 360 degree / spherical view. Such videos may be collected from vehicles or indoor applications.
- **Frame Camera**: Single frame images like those commonly taken from drones or cell phone cameras.
- **Panorama Camera**: Images that have field of view greater than 180 but less than 360. Typically made by stitching a few images together to have a large horizontal field of view.

Add Oriented Imagery Fields (Optional)

This tool is used to add key OIC fields to the exposure points attribute table. After selecting an OIC, the user is presented with a list of missing fields. The user can select a field and enter a value, which will then be added to the attribute table. The user can also use the calculate tool to change attribute values.

If a value is undefined because there is no attribute field or the value of an attribute is null, the default value in the OIC properties will be used.



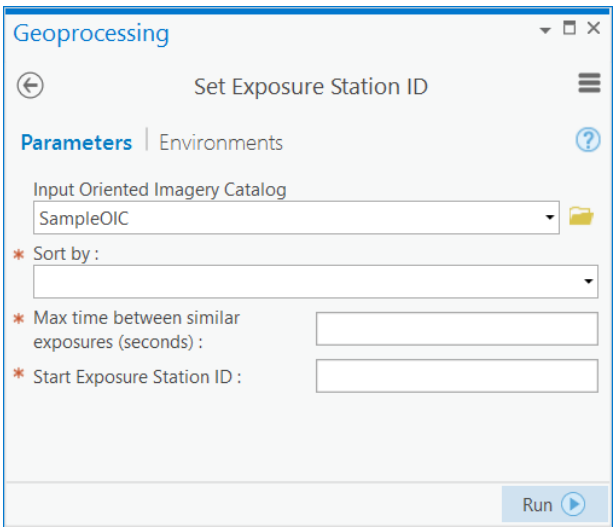
Set Exposure Station ID (Optional)

This tool is used to group images based on the time between exposures. Images with the same Exposure ID are virtually mosaicked in the oriented imagery viewer. As you pan around the image, the system will transition to another image with the same Exposure Station ID.

To use the tool, you need to specify:

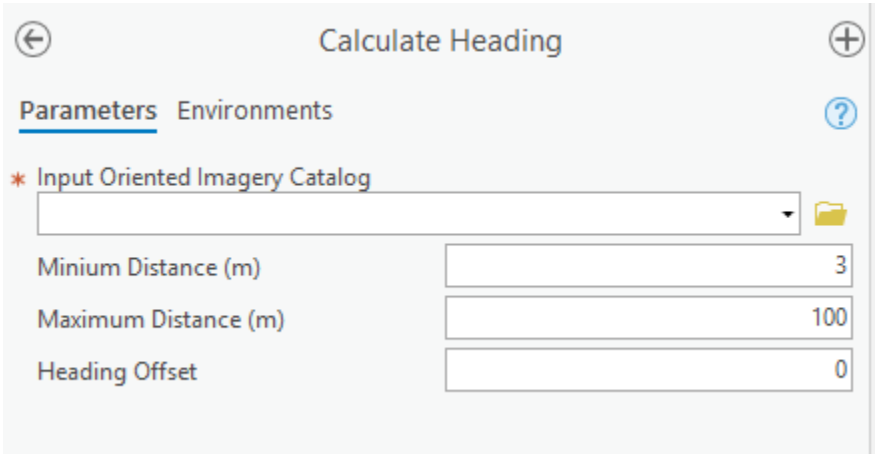
- A field to sort the images
- Maximum exposure time
- A starting Station ID

The program will then search the records and group images based on images that were taken within the maximum exposure time specified. For each group the station ID is incremented by one (1). The tool also averages the coordinates of the points based on the grouping.



Calculate Heading (Optional)

The Calculate Heading tool calculates the camera heading for each exposure point in relation to its preceding and subsequent points, read in order of ObjectID in the attribute table. Typically, this is used for street-level imagery where images are taken in sequence.



There are three parameters for this tool:

- **Minimum Distance**. Minimum valid distance between two locations. If less than this value, it is assumed the vehicle has stopped and heading will be same as previous value. The default is 3 meters.

- **Maximum Distance**. Maximum valid distance between two locations. If greater than this value, it is assumed that there has been a sudden change in location and heading will be computed from next valid value. The default is 100 meters.
- **Heading Offset**. Offset in degrees between vehicle heading and camera heading. The default is 0.

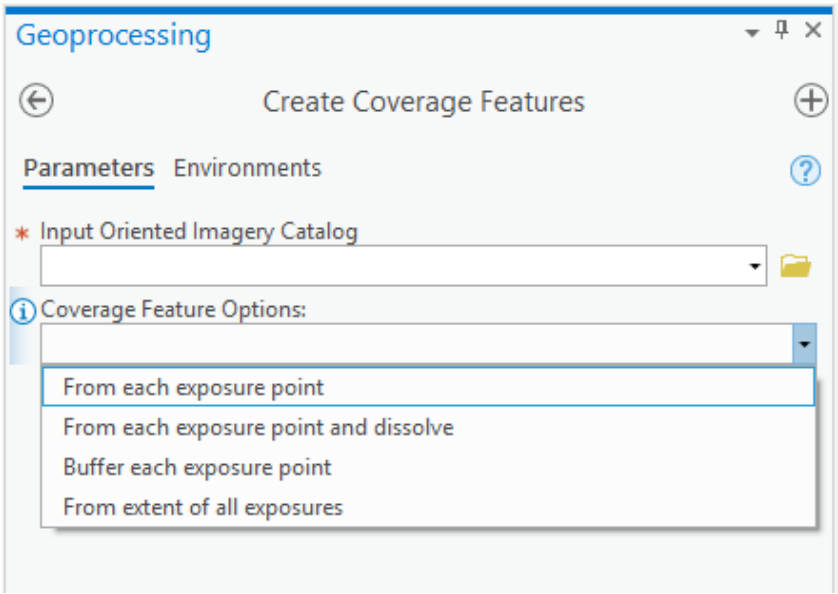
Create Coverage Features (Recommended)

After adding images to an oriented imagery catalog, the location of each image is included in the Camera Exposure Points feature class.

Creating a coverage feature class is not a requirement if you are confident in the attributes, and you don't intend for users to review the coverage of the imagery.

Depending on the sensor, the level and accuracy of the metadata may be limited. The Create Coverage Features tool computes the footprint for each image—you can view these footprints to check some of these parameters. The *location* of these points should be reviewed to ensure that are correct. If not in the correct position, then something went wrong when adding the data. It is also important to check the *orientation* of the imagery.

In some cases, you may find you want to edit the attributes in the exposure feature class and then recreate the coverage features. There are various tools in ArcGIS Pro for editing attributes, including the Attribute Calculator.



The tool prompts for the name of the oriented imagery catalog and the coverage feature options.

There are four methods to calculate coverage features:

1. **From each exposure point.** Creates coverage features based on all the required parameters.

   Use this option to check the orientation of the imagery that has been added. To tweak the coverage features, you can change the values for Camera Heading, Camera Roll, Horizontal and Vertical Field of view, Average Height, Far distance and Near distance.

   For bubble imagery, only the far distance is used to create a coverage map. This option is best used if the exposure points are few and scattered over a large project area.

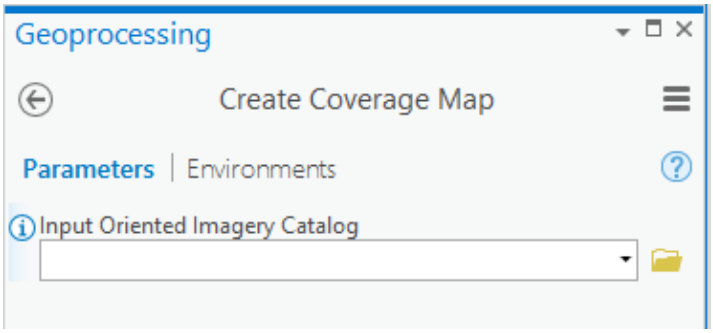
2. **From each exposure point and dissolve.** Same as the option above, but the features are dissolved together to create a more optimized coverage feature map.
3. **Buffer each exposure point.** Buffers each exposure point using the Far Distance value. This option works best works for street view imagery.
4. **From extent of all exposures.** Creates coverage features based on the extent of the exposures point feature class. This option is best used if there are many points in a small location.

Create Coverage Map (Recommended)

Note: Coverage features should be finalized before creating the coverage map. If coverage features have not been created, it will create them.

The coverage map, once published to ArcGIS Online, provides users with an indication of the available imagery coverage.

The Create Coverage Map tool creates a vector tile cache of the coverage features. A vector tile cache is more scalable than a feature service for such display and can handle many millions of records. It is optional, but aids in finding areas with imagery in cases were the coverage is not uniform.

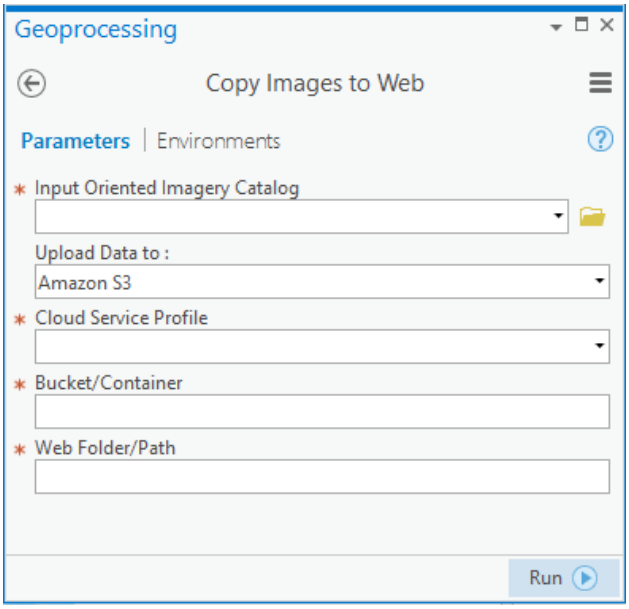


Copy Images to Web (For local images only)

Note: Oriented imagery managed using an OIC must be web-accessible. If the data you're using is already web accessible this tool is not required.

If the images you've added to your OIC are stored locally, use this tool to copy the images to web-accessible storage. For most users, the simplest way to make imagery web-accessible is to upload the imagery to Amazon, Google or Azure cloud storage.

This tool will update the paths in the OIC from the local image location to the web-accessible location. You can use the Select Local Images tool to double-check that all your paths are pointing to cloud storage. If you use another tool to copy your imagery to cloud storage, you can use the Repair Paths tool to update the path to the cloud storage location.



The following parameters are required:

- **Upload Data to**. Defines the cloud storage provider. Amazon S3, Microsoft Azure, and Google Cloud Storage are supported.
- **Cloud Service Profile.** Defines a profile that stores the credentials required to write to cloud storage. If set up correctly on your machine you should see a list of available profiles. If you don't see profiles, refer to the OptimizeRasters documentation for help creating them.
- **Bucket/Container.** Defines the name of the destination bucket to store your images. The bucket must already exist.
- **Web Folder/Path.** Defines the name of the web folder or path to which the data should be written. If the folder does not exist, the program will attempt to create it and then copy all the images to that location.

Note about OptimizeRasters

This tool uses OptimizeRasters to upload imagery into cloud storage. It may require the installation of some additional Python packages, depending on the platform:

Cloud platform	Python package needed
Amazon S3	boto3 (boto is no longer supported)
Microsoft Azure	azure
Google Cloud Storage	google-cloud-storage (google-cloud is no longer supported)

Note that cloud storage security can be defined in multiple ways. Defining detailed security policies with detailed roles is possible, but often not necessary. Often a simpler alternative is to use a public bucket that is not listable, then define the names of the images so they contain random characters and are not accessible or listable unless the program knows the URL of the file. In this way, the URLs can be included in the Exposure Locations feature service, but only users with access to the feature service can access the files.

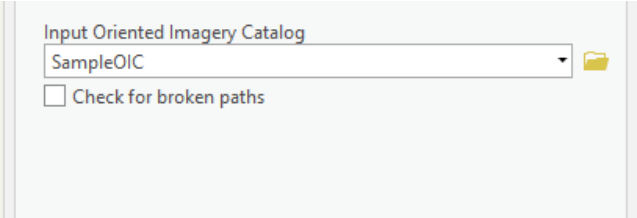
For more details on how OptimizeRasters works and for help setting up the appropriate accounts or security, please see OptimizeRasters documentation included as part of this install or in the [OptimizeRasters GitHub repo](#).

Analyze Oriented Imagery Catalog (Optional)

This tool should be run before publishing an OIC. It will analyze the selected oriented imagery catalog for errors and any optimization errors. Any errors found are displayed as warnings in the GP tool window, along with recommendations for the users to address the issue.

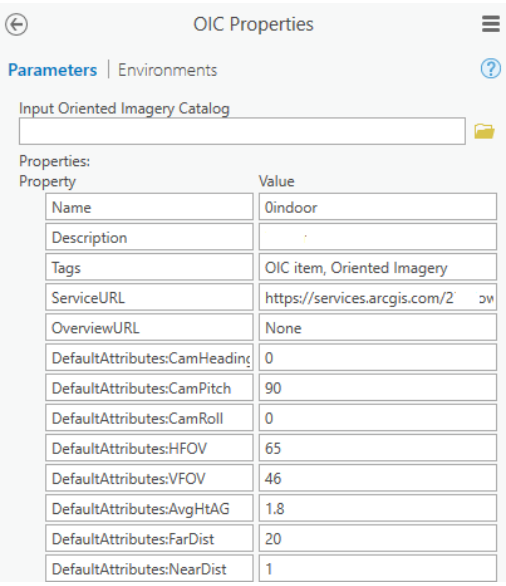
The tool performs the following tests:

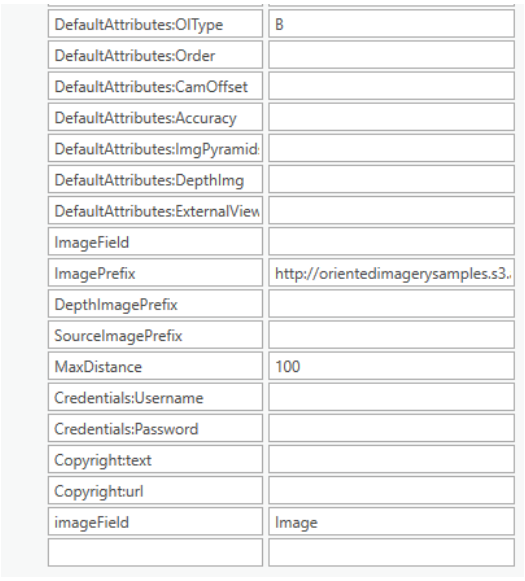
- Checks for broken paths (User option. Should only be used if a user thinks paths might be broken as this is an expensive operation.)
- Checks if there are any local images in the OIC
- Checks if there are any empty fields that should be removed before publishing
- Checks for missing values in the parameter file
- Checks if the coverage map has been built or not



OIC Properties (Optional)

This tool enables all the properties of an oriented image catalog to be set. Some of these parameters are defined when creating the OIC or adding data, but can be edited here.





Following is an explanation of the properties:

- **Name**. Name to be displayed

- **Description**. Short description often used in the titles
- **Tags**. Tags used to aid identification and help search
- **ServiceURL**. URL to the Exposure point service or item.
- **OverviewURL**. URL to the service to be used to provide overview coverage
- **DefaultAttributes**. Following are default for each field:
    - **CamHeading**. Camera heading, defined in degrees. Often set to 0 for processed spherical imagery. CamHeading = 0 means the top of the camera is oriented toward the North, or CamHeading = 90 means the top of the camera is oriented toward the East. Value should be between 0 and 360. A negative value or NULL indicates that orientation is not known.
    - **CamPitch**. Defined in degrees. CamPitch = 90 means camera is viewing toward the horizon. CamPitch=0 means camera is pointing straight down
    - **CamRoll**. Measured in degrees. CamRoll will typically be near 0. CamRoll is used to account for +/- 90 degree rotations in the camera housing
    - **HFOV**. Horizontal Field of View in degrees. This typically refers to the FOV for the rows direction of the camera, but may be column direction if the CamRoll is +- 90. The value can be estimated based on the focal length of the camera and CCD width.
    - **VFOV**. Vertical Field of View in degrees. Typically refers to the VOV for the column direction of the image, but may not be if CamRoll is +/- 90. The value can be estimated based on the focal length of the camera and CCD height.
    - **AvgHtAG**. Average Height (m) of the camera above ground.
    - **FarDist**. Furthest usable distance from the camera position of the imagery in meters.
    - **NearDist**. Nearest usable distance of the imagery in meters. Can be negative if the camera looks partially backwards
    - **OIType**. Oriented Imagery type. Controls some default behavior of the viewers
        - **T**. Terrestrial (Default). Typical frame imagery taken at street level
        - **O**. Oblique. Frame image from aerial or drone sensor
        - **I.** Inspection. Frame image with feature of interest are near camera
        - **B**. Bubble. A single stitched panorama image with 360view
        - **S**. Spherical. 6 separate images stitched together to form a cube.
        - **D**. Multiple frame images taken from the same location pointing in different directions, but considered a group.
        - **P**. Panorama. HFOV >4x VFOV
        - **V**. Video. Image may or may not exist. Video field should define location of video
        - **ArcGIS Image Service**. Defines that mensuration should be achieved using ArcGIS Image Server.
    - **Order**. Field used to define the order of the imagery in image galleries or workflows that inspect one image after the other. There is little use in having a single default order.

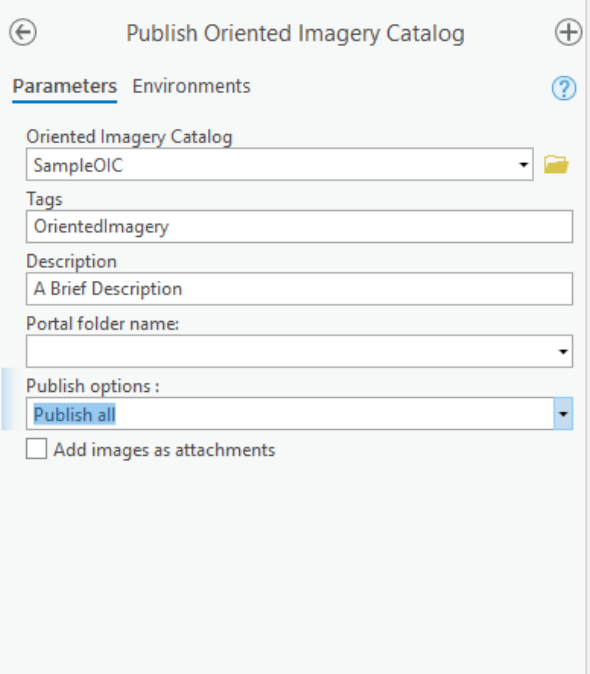
    - **CamOffset**. Offset between the recorded camera location and its actual location. There is little use in having a single default order.
    - **Accuracy**. Accuracy of the measurements. Comma delimited string with the following order StdDevX(m), StdDevY(m), StdDevZ(m), StdDevHeading(deg), StdDevPitch(deg), StdDevRoll(deg),StdDevDistNear(m), StdDevDistFar(m). Based on these value an indication of the measurement accuracy will be computed.
    - **ImgPyramids**. Naming used for external pyramids. See developer doc.
    - **DepthImg**. Definition of a depth image. Use for some sensors that that can provide depth to help determine coordinates from an image.
    - **ExternalViewer**. Link to an external viewer. See developer doc.
- **ImageField**. If defined, then the name of the field to be used as the Image Field
- **ImagePrefix**. Prefix added to Image attribute. Used for example to define first part of a URL
- **VideoPrefix**. Prefix added to Video attribute.
- **DepthImagePrefix**. Prefix added to DepthImage attribute
- **SourceImagePrefix**. Prefix added to Source image
- **MaxDistance**. Max search distance to be used
- **Editor**. Defines if Editor mode should be enabled True/False
- **Credentials**. Credentials required to access imagery
- **Username**. For some authentication systems
- **Password**. For some authentication systems (Do not enter any secure passwords here)
- **Copyright**. Define copyright/attribute text to display on image

Publish Oriented Imagery Catalog (Required)

This tool is used to publish oriented imagery catalogs to ArcGIS Online or an ArcGIS Enterprise portal.

Note: Publishing requires that the user has administrator- or publisher-level access to ArcGIS online or the ArcGIS Enterprise portal.

The tool creates or updates the OIC item on ArcGIS Online and, if required, publishes the associated Camera Event feature class and Coverage Map as services.

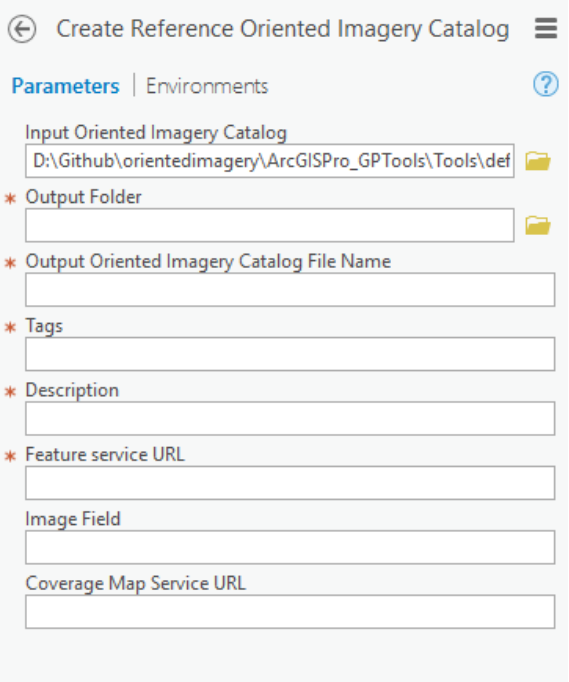


- **Tags**. Defines a list of semicolon-delimited tags to be associated with the OIC in ArcGIS Online
- **Description**. Defines the associated ArcGIS Online description
- **Portal folder name**. Can be used to define a folder in your Content on the portal, so that the items are not added to the root folder.
- **Publish options**:
    - **Publish all**. Publishes the Camera Exposure feature service, the Map Coverage feature service (if available), and the oriented imagery catalog item. If any of the items already exists, the tool will fail. This is typically the default when publishing. Note that the inclusion of the Coverage Map is optional; if it does not exist it will not be published and will not cause the tool to fail.
    - **Publish all - overwrite**. This option will overwrite the items if they already exist (if they don't exist, this option will not be available). This is used if you need to update all the items.
    - **Publish Oriented Imagery Catalog item only**. Should be used when you want to update an existing OIC with new properties, but don't want to change the existing camera event feature service. It can also be used to publish a Reference Oriented Imagery Catalog. It will create a new item, but will not attempt to overwrite the existing feature services. It will fail if the OIC item already exists.
    - **Publish Oriented Imagery Catalog item only - overwrite**. This option will overwrite an existing OIC item. This is used to update properties for an OIC item without making any other changes.

Create Reference Oriented Imagery Catalog (Optional)

Use this tool to create a new oriented imagery catalog item that references the services of another OIC item, but enables properties to be modified.

Typically, after using this tool the properties of the new OIC would be edited using the OIC Properties tool.

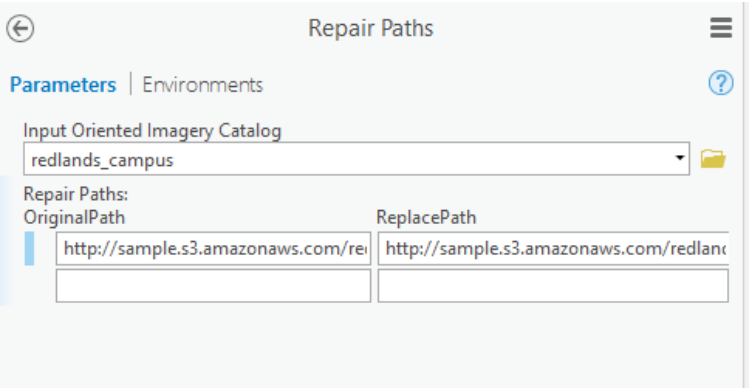


- **Input Oriented Imagery Catalog**. An existing OIC. This can be a path to a local OIC file, or the URL to an OIC item in ArcGIS Online.
- **Output Folder** and **Output Oriented Imagery Catalog File Name**. the local folder location and name of the new oriented imagery catalog should be defined. The program will copy the OIC to new the new file. Properties can then be edited.
- **Tags** and **Description**. Define new tags and description for the reference OIC.
- **Feature service URL** and **Coverage Map Service URL**. Optionally, the reference to the Camera Exposure point feature service or the Coverage map service can be redefined.
- **Image Field**. The tool enables the definition of an Image Field, which enables an OIC to be created from a point feature class that references images using a different field name. The Image Field can be used to create an OIC that references a feature service that already includes a reference to imagery. A typical example may be a feature class/service of geotagged images that have been loaded using some other tools.

Note that this tool will not attempt to read the EXIF headers of the imagery nor modify the feature services. It will assume that default parameters are suitable or that the suitable parameters have been defined independently in the feature class/service.

Repair Paths (Optional)

This tool is used to repair paths in an OIC. It can be used with either file paths or URLs, so it's especially useful to update the path if your images have been copied to cloud storage using a tool other than the Copy Images to Web tool.



You'll be prompted to add the following paths:

- **OriginalPath**. The path you want to replace. You can use the original path value as-is, or edit it to replace a partial path.
- **ReplacePath**. Enter the path you want to replace OriginalPath.

Example:

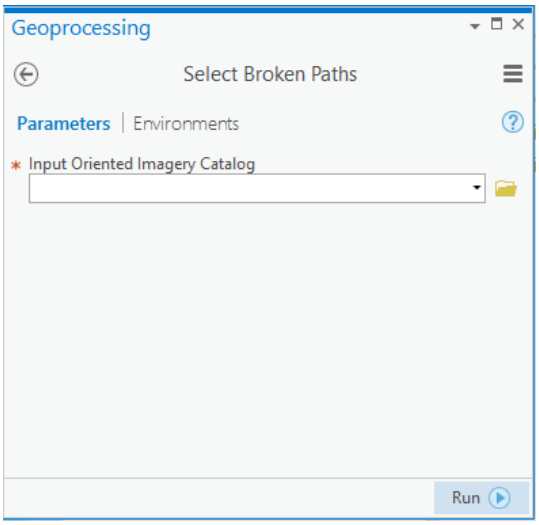
OriginalPath: C:/Imagery/redlandsCampus
ReplacePath: http://redlandssample.s3.amazonaws.com/campus

Select Broken Paths (Optional)

This tool is used to determine which paths are broken. It will select any record with a path that doesn't point to a valid local data source or URL.

For example, if images in your OIC were originally located in C:\images, but those images were deleted or moved, this tool will select all those records.

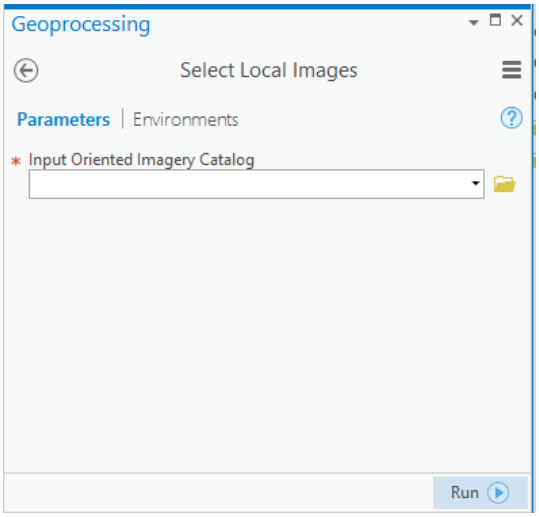
Once the images are selected, you can use the Repair Paths tool to point to the new location.



Select Local Images (Optional)

This tool selects any records that are pointing to local file locations. Since the images must be web-accessible, this tool should be used before publishing an OIC to determine which records have not been updated with a URL or still point to local images.

After local paths are selected, you can either upload those images to the web using the Copy Images to Web tool, or use the Repair Paths tool to edit the path to point to the correct cloud storage location.



Appendix A: Custom User Types (Developer Brief)

Oriented imagery catalogs can be created using a number of imagery sources. Most sources are standard JPG imagery with EXIF (Exchangeable Image File Format) headers. However, there are other sources which may have image info stored as a table or as a custom header file. You can write your own Python script to read this information and use it to create the OIC table.

OIC user types are composed of a Python script with an accompanying JSON parameters file that has various input parameters that can be accessed using the Oriented Imagery Management GP tools.

Refer to the inbuilt types in C:\Image_Mgmt_Workflows\OrientedImagery\Types for examples.

Creating an OIC type

A new OIC Type can be created using the following steps:

1. Locate the OIC Types folder (by default located in the C drive).
   E.g., C:\Image_Mgmt_Workflows\OrientedImagery\Types
2. Create a new Python file in the OIC Types folder.
   E.g., <xyz>.py
3. Create a Parameters file in the OIC Types folder.
4. Eg: <xyz>.oictype

Note: Make sure that the names of the Parameters file and the Python file are the same.

Once these files are created, in the Add Oriented Imagery Catalog GP tool, you will be able to see your new Oriented Imagery Type in the Input Type drop down. (A refresh of the tool or restart of Pro might be required.)

OIC Parameters file

This file is supposed to be in the JSON format. Two main keys are expected.

1. **Input**. The value of this key is a nested dictionary(key/value pair) with the following keys.
    a. **Name**: This is the input name
    b. **Type**: This is the Input Type. Currently we support types File/Folder
    c. **Filter**: In case of Folder input type, you can provide optional comma-separated file filters.
2. **Parameters**. (Optional) Add any additional parameters you require in your Python file.

Parameter file, e.g.:

```
{
        "Input": {
                "Name": "Input Images Folder",
                "Type": "Folder",
                "Filters": ".jpg,.tif"
        },
        "Parameters": {
        }
}
```

OIC Type Python file

The OIC Type Python file has a minimum requirement of two methods. These are described in detail below.

- **Init method**: This method initializes the OI Type name.
- **Main method**: This method will be internally called on running the Add Images to Oriented Imagery Catalog GP Tool with the Input Type selected as your custom input type. The return type for the Main method is a string that contains “success” or “error” based on whether the method returned successfully or not.
  The following arguments (in the corresponding order) are passed to this method internally.
    - **Exposure point feature class path**: This is full path to the exposure points feature class
    - **OIC Parameters**: This is a dictionary containing the additional parameters you have defined in the OIC Type file under “Parameters”. Each of these parameters can be referenced using the same key defined in the oictype file.
    - **Input Parameters**: This is a dictionary containing the input folder/file path and the file filters if any. The file path can be referenced using the key ‘file’, folder path using the key ‘folder’ and list of filters using the key ‘filter’
    - **Default values**: This is a list of dictionaries containing the default parameter values. The default values are used when no other values can be found. An example of the dictionary is below.
      {“name”:”CameraRoll”, “value”:”10.4”,”isDefault”:False}

For each parameter, you need to check if there is a value for it that can be derived from the image metadata. Ideally, the value will be derived from the image metadata. Default values should only be used in the event of no metadata values.

If there is a value, and if “isDefault” is True, then the value needs to be written to the OIC File. This is done so that the same value need not be written to the table for each row, thereby reducing the size of the table.

- If there is a value and if “isDefault” is False, then the value is written to the table.
    - **Log**: This is the logger object for the OIType.
        - For logging a message, use log. Message(“message”, log.const_general_text)
        - For logging a warning, use log. Message(“message”, log.const_warning_text)
        - For logging an error, use log. Message(“message”, log.const_critical_text)

A sample skeletal structure for the Python file is given below:

```
def __init__(self, nametype):
  self._name = nametype

def main(exposurePointFeatureClassPath, oicParameters, inputParameters, defaultValueList,
log):
  try:
    input_folder = inputParameters[‘folder’]
    file_filters = inputParameters[‘filter’]
    param1 = oicParameters[‘param1’]
    #add code to add images
    return (“Successfully added images”)
  except Exception as e:
    log.Message(“Error in adding images”, log.const_critical_text)
    return(“Error in adding images”)
```