# Wall-to-Wall Python library

Contents

[Wall-to-Wall Python library 1](#_Toc178945831)

[Introduction 1](#_Toc178945832)

[Command-line Tools 2](#_Toc178945833)

[Commands 2](#_Toc178945834)

[Configuration Format 3](#_Toc178945835)

[Requirements 3](#_Toc178945836)

[Builder configuration 3](#_Toc178945837)

[Fully-specified configuration 4](#_Toc178945838)

[Examples 6](#_Toc178945839)

[CASFRI - YT 6](#_Toc178945840)

[Standalone Template 7](#_Toc178945841)

[Appendix: Configuration File Format 8](#_Toc178945842)

## Introduction

The wall-to-wall tool helps simplify preparing and running GCBM simulations by consolidating the manual workflow into a standardized configuration file format. Most projects can be run exclusively with this tool, with only advanced or complex simulations requiring the use of the tiler library and custom scripting.

The tool performs these steps for a GCBM project

1. tiles pre-rollback layers
2. creates input database
3. optionally runs spatial inventory rollback:
   1. tiles post-rollback layers
   2. creates post-rollback input database
4. configures GCBM to run

## Command-line Tools

Users interact with the wall-to-wall library through a Python command-line application installed into the ‘Scripts’ directory of the local Python installation. The application accepts a JSON configuration file containing all the necessary information for tiling the spatial layers, creating the input database, performing a spatial inventory rollback, and configuring and running GCBM.

### Commands

walltowall build <builder config file path> [output config file path]

* Use the builder configuration contained in the config file to fill in and configure the rest of the project; creates the fully-specified configuration in a separate file in the same directory as the builder config file unless an output config file path is specified.

walltowall prepare <fully-specified config file path> [output root path]

* Using the fully-specified project configuration in the config file, tile the spatial layers, generate the input database, run the spatial rollback, and configure the GCBM run. Project files are generated in various subdirectories off the specified output root path, or the directory containing the config file by default.

walltowall merge <fully-specified config file path> <prepared project root 1> <prepared project root 2> [prepared project root n …] --output\_path <output root path> [--include\_index\_layer]

* Using the fully-specified project configuration from the first argument to get the GCBM config file template path and disturbance order, merge two or more projects together in descending priority order into a single project in the specified output path. Projects must have been prepared by the walltowall scripts and have the same spatial extent and resolution. If include\_index\_layer is specified, a layer is generated showing which dataset was used for each pixel.

walltowall run (local|cluster) <prepared project root> [--config\_path <fully-specified config file path>] [--end\_year <yyyy>] [--title <title> (cluster only)] [--compile\_results\_config <compile results config path> (cluster only)]

* Run the specified project either locally or on a gcbm\_celery cluster, using the optionally specified walltowall config file for the project title and overridden executable or distributed run client paths. If end\_year is specified, the current simulation will run to that year without permanently altering the configured value. For cluster runs only, a title and a custom compile results configuration can also be specified.

## Configuration Format

The configuration format for a project using the wall-to-wall library is designed to be more streamlined and user-friendly than writing a tiler script and running the Recliner2GCBM GUI, as well as allowing known standardized collections of input data (i.e. CASFRI) to be configured automatically through Python code in the library instead of through explicit configuration by the user.

There are two parts to the configuration: the optional "builder" configuration used with the "walltowall build" command that acts as a shorthand format for automatically configuring a project based on standardized sets of input data, and the fully-specified configuration read by the "walltowall prepare" command that contains all of the specific details needed to prepare and run the simulation.

### Requirements

The configuration file starts with the project name, followed by the optional builder configuration, and then the fully-specified configuration which can either be initially blank and generated by the builder using "walltowall build", or written manually if not using a builder.

{

"project\_name": <project name>,

[optional builder configuration]

(fully-specified configuration)

}

### Builder configuration

The builder configuration is an optional shortcut for creating projects using input data that conforms to the requirements of a builder module, usually allowing a much shorter configuration to generate the more detailed fully-specified configuration. It consists of the type of builder to use, followed by any configuration items recognized by the selected builder, and optionally any fully-specified configuration items that should take precedence over configuration generated by the builder.

"builder": {

"type": <builder type name>,

(any configuration specific to the selected builder)

[any main configuration items that take priority over builder-generated configuration]

}

Example:

{

"builder": {

"type": "casfri",

"casfri\_data": "../00\_preprocessing/casfri\_data/processed/YT03",

"other\_data": "../00\_preprocessing",

"yield\_table": "../00\_preprocessing/aspatial/yields/afforestation\_national/YT\_yield\_table.csv",

"dm\_xls": "../00\_preprocessing/aspatial/archive\_index/casfri\_dms.xlsx",

"aidb": "../00\_preprocessing/aspatial/archive\_index/casfri\_archive\_index.mdb",

"age\_distribution": "../../00\_preprocessing/aspatial/BGI\_age\_class\_dist.xlsx"

"rollback": {

"age\_distribution": "rollback/age\_distribution.json",

"inventory\_year": 2022

}

}

}

### Fully-specified configuration

The fully-specified configuration contains all of the details required to prepare and run a GCBM simulation. Because the objectives for this tool are to increase user-friendliness and efficiency, many of the configuration items are optional, and the tool will attempt to detect or use defaults for omitted configuration.

#### Layer definition

There are several places in the configuration that accept either a layer path for simple layers that comply with all the default settings, or a layer definition where the details of the layer can be specified more explicitly. Layer definitions can take these settings:

"layer": <path to layer>

[optional] "attribute": <attribute to read; default: search by layer name, or use first attribute found>

[optional] "attribute": {

<attribute to read>: <attribute value to filter for; only matching polygons are rasterized>

}

[optional] "lookup\_table": <path to csv file>

The lookup table replaces original pixel or attribute values with new ones. For rasters, the first column is the pixel value, and subsequent columns are the attributes. For vectors, the columns are paired up; the first column in each pair is named after the attribute to replace and contains the original values, and the second column’s name doesn’t matter and contains the values to substitute.

If a substitution does not appear in the lookup table, the original value is used.

If no lookup table is specified, the wall-to-wall tool searches first for <layer filename>.csv in the same directory as this configuration file, and then in the layer directory.

[optional] "strict\_lookup\_table": <true/false>

If strict lookup table is enabled and the layer is a vector layer, polygons with attribute values that do not have a substitution in the lookup table are excluded from rasterization.

#### Required configuration items

"aidb": <path to ArchiveIndex database (AIDB)>,

"yield\_table": <path to yield table>,

"yield\_interval": <yield table age/volume interval, in years>,

##### Classifiers

Classifiers link the yield table to the spatial landscape. They are configured with at least a layer definition for the spatial component, and by default the walltowall script will attempt to match the spatial layer to the yield table, or the corresponding yield column and even a separate file containing all possible classifier values can be configured explicitly.

At least one classifier must be included in the "classifiers" section, with the configuration format:

"classifiers": {

<classifier name>: {

<layer definition items for spatial layer linked to classifier>,

[optional] "values\_path": <file with classifier values; default: use yield table>,

[optional] "values\_col": <column name/# in values\_path; default: use only column in single-column

file, or classifier name or spatial attribute, or column where values intersect with

spatial values>,

[optional] "yield\_col": <column name/# in yield table; default: use values\_col if values\_path is

the same as yield table path, otherwise search by classifier name, or search for column where

values are a subset of values from values\_path>

}

}

#### Optional configuration items

"resolution": <pixel resolution in degrees lat/lon>,

"gcbm\_config\_templates": <path to directory containing GCBM simulation config file templates>,

"disturbance\_order": <path to disturbance order text file>,

##### Bounding box

The bounding box defines the simulation area; nodata pixels in the bounding box are propagated to all other spatial layers. If the bounding box is not configured, the "initial\_age" layer is used.

"bounding\_box": <path to layer or full layer definition>

##### Layers

This section holds all the "miscellaneous" spatial layers that are not the bounding box, classifiers, or disturbances.

"layers": {

<layer name>: <path to layer or full layer definition>

}

##### Disturbances

This section holds any disturbance layers to be included with the simulation. The keys are the filenames or file patterns to search for, followed by the disturbance details. If the disturbance layer is a shapefile, the wall-to-wall tool will automatically split it into tiled layers by year.

"disturbances": {

<layer path or glob file pattern>: {

[optional] "year": <specific year, or name of attribute containing year, or "filename">

If specific year is given or the special keyword "filename" is used, use that exact year or try to parse it from the filename, respectively. If year is not specified, wall-to-wall will first search for an attribute named "year", then for the first column where all the values are 4-digit integers, and finally will check if the disturbance year can be parsed from the filename.

[optional] "disturbance\_type": <specific disturbance type, or name of attribute

containing disturbance type>

If disturbance type is not specified, wall-to-wall will search for the first attribute where all the values appear in tblDisturbanceTypeDefault in the AIDB.

[optional] "age\_after": <specific age after, or name of attribute containing reset age>

If age\_after is not specified, wall-to-wall will search for an attribute named "age\_after" or fall back to no transition rule if not found. If age\_after or regen\_delay are present, this causes a transition rule to be attached directly to the disturbance layer. regen\_delay can be omitted even if age\_after is present.

[optional] "regen\_delay": <specific regen delay, or name of attribute containing regen delay;

default: 0>

If regen\_delay is not present, wall-to-wall will search for an attribute named "regen\_delay" or fall back to no transition rule if not found and age\_after is not present. If age\_after or regen\_delay are present, this causes a transition rule to be attached directly to the disturbance layer. age\_after must be configured if regen\_delay is present, as there is no sensible default.

[optional] <other Layer definition items>

}

}

##### Rollback

If present, this section causes a spatial inventory rollback to be run.

"rollback": {

"age\_distribution": <path to age distribution JSON or Excel file>,

"inventory\_year": <path to inventory vintage layer or full layer definition, or global

inventory vintage year>,

[optional] "rollback\_year": <year to roll back to; default: 1990>,

[optional] "prioritize\_disturbances": <true/false, default: false>,

[optional] "single\_draw": <true/false, default: false>,

[optional] "establishment\_disturbance\_type": <default establishment disturbance type name>,

[optional] "disturbance\_order": <path to disturbance order text file>

}

## Examples

### CASFRI - YT

This configuration uses the CASFRI builder in the wall-to-wall tool to set up the Yukon simulation – for this example, assume that it's stored in a file called yt\_casfri.json:

{

"project\_name": "casfri\_yt",

"builder": {

"type": "casfri",

"casfri\_data": "../00\_preprocessing/casfri\_data/processed/YT03",

"other\_data": "../00\_preprocessing/other\_data",

"yield\_table": "../00\_preprocessing/yields/afforestation\_national/YT\_yield\_table.csv",

"dm\_xls": "../00\_preprocessing/archive\_index/casfri\_dms.xlsx",

"aidb": "../00\_preprocessing/archive\_index/casfri\_archive\_index.mdb",

"rollback": {

"age\_distribution": "rollback/age\_distribution.json",

"inventory\_year": 2022

}

}

}

After running walltowall build yt\_casfri.json, the builder fills in the rest of the project details based on some assumptions about the way CASFRI projects are structured:

{

"project\_name": "casfri\_yt",

"builder": {

"type": "casfri",

"casfri\_data": "../00\_preprocessing/casfri\_data/processed/YT03",

"other\_data": "../00\_preprocessing/other\_data",

"yield\_table": "../00\_preprocessing/yields/afforestation\_national/YT\_yield\_table.csv",

"dm\_xls": "../00\_preprocessing/archive\_index/casfri\_dms.xlsx",

"aidb": "../00\_preprocessing/archive\_index/casfri\_archive\_index.mdb",

"rollback": {

"age\_distribution": "rollback/age\_distribution.json",

"inventory\_year": 2022

}

},

"resolution": 0.001,

"aidb": "../00\_preprocessing/archive\_index/casfri\_archive\_index.mdb",

"yield\_table": "../00\_preprocessing/yields/afforestation\_national/YT\_yield\_table.csv",

"yield\_interval": 10,

"classifiers": {

"RU": {

"layer": "../00\_preprocessing/pspu/pspus\_2016.shp",

"attribute": "Reconcilia"

},

"LeadingSpecies": {

"layer": "../00\_preprocessing/casfri\_data/processed/YT03/layer\_1/leading\_species.tiff",

"values\_col": "casfri\_species\_name"

}

},

"layers": {

"initial\_age": "../00\_preprocessing/casfri\_data/processed/YT03/layer\_1/age\_2022.tiff",

"mean\_annual\_temperature": "../00\_preprocessing/other\_data/NAmerica\_MAT\_1971\_2000.tif",

"admin\_boundary": {

"layer": "../00\_preprocessing/other\_data/pspu/pspus\_2016.shp",

"attribute": "ProvinceNa"

},

"eco\_boundary": {

"layer": "../00\_preprocessing/other\_data/pspu/pspus\_2016.shp",

"attribute": "EcoBound\_1"

}

},

"disturbances": {

"../00\_preprocessing/casfri\_data/processed/YT03/layer\_1/disturbances\_\*.tiff": {}

},

"rollback": {

"age\_distribution": "rollback/age\_distribution.json",

"inventory\_year": 2022

}

}

After the fully-specified project configuration is generated, the project can be prepared by running:

walltowall prepare yt\_casfri.json

This tiles the spatial layers, generates the input database, runs the spatial inventory rollback, generates the rollback input database, and finally configures the GCBM simulation for running.

Finally, the simulation can be run either locally or on the cluster (with wall-to-wall correctly configured and tunnels already connected):

walltowall run local . --config\_path yt\_casfri.json

walltowall run cluster . --config\_path yt\_casfri.json

### Standalone Template

This assumes the same directory structure as the standalone template training project, with the wall-to-wall configuration file located in an extra ‘config’ directory (walltowall\_config.json), and the input files in the usual locations.

There is no builder shortcut for this type of custom project, so we skip the builder step and go straight to hand-writing the fully-specified configuration, relying on the defaults to keep everything concise:

{

"project\_name": "standalone\_template",

"start\_year": 2010,

"end\_year": 2020,

"resolution": 0.00025,

"aidb": "../input\_database/ArchiveIndex\_Beta\_Install.mdb",

"yield\_table": "../input\_database/yield.csv",

"yield\_interval": 10,

"classifiers": {

"Classifier1": {

"layer": "../layers/raw/inventory/inventory.shp",

"attribute": "Classifer1"

},

"Classifier2": {

"layer": "../layers/raw/inventory/inventory.shp",

"attribute": "Classifer2"

}

},

"layers": {

"initial\_age": {

"layer": "../layers/raw/inventory/inventory.shp",

"attribute": "AGE\_2010"

},

"mean\_annual\_temperature": {

"layer": "../layers/raw/inventory/inventory.shp",

"attribute": "AnnualTemp"

}

},

"disturbances": {

"../layers/raw/disturbances/disturbances.shp": {

"age\_after": 0,

"regen\_delay": 0

}

}

}

After the fully-specified project configuration is written, the project can be prepared by running this command from the project root directory, i.e. Standalone\_Template\:

walltowall prepare walltowall\_config.json .

This tiles the spatial layers, generates the input database, runs the spatial inventory rollback, generates the rollback input database, and finally configures the GCBM simulation for running.

Finally, the simulation can be run either locally or on the cluster (with wall-to-wall correctly configured and tunnels already connected):

walltowall run local . --config\_path config\walltowall\_config.json

walltowall run cluster . --config\_path config\walltowall\_config.json

## Appendix: Configuration File Format

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Section** | **Item** | **Required?** | **Value** | **Behaviour** | **Validation** |
| <root> | project\_name | Y | string | Use as project name. |  |
| <root> | project\_name | Y | null/empty | Required field, throw exception. |  |
| <root> | builder | N | dict | Contains optional builder configuration. | Builder section must include at least "type" item. |
| builder | type | Y\* | string | Use as builder name. | Ensure builder with specified name exists. |
| builder | type | Y\* | null/empty | Required field if using optional builder section - throw exception if so. |  |
| <any layer definition> | layer | Y | string | Use as path to spatial layer. | Ensure file exists. |
| <any layer definition> | layer | Y | null/empty | Required field, throw exception. |  |
| <any layer definition> | attribute | N | string | Use as attribute to read from. | Ensure attribute exists in layer's attribute table: csv file with same name as raster layer in config file directory or layer directory, or attribute table natively paired with vector layer. |
| <any layer definition> | attribute | N | dict | Accept a single-item dictionary where the key is the attribute to read from, and the value is the attribute value to filter for; only polygons matching the filter value are rasterized. | Ensure attribute exists in layer's attribute table: csv file with same name as raster layer in config file directory or layer directory, or attribute table natively paired with vector layer. |
| <any layer definition> | attribute | N | null/empty | Use attribute matching layer name if exists, otherwise use first attribute in layer if layer has an attribute or lookup table, or use pixel value directly if a raster layer. |  |
| <any layer definition> | lookup\_table | N | string | Use as path to lookup table to replace original pixel or attribute values with new ones. Lookup table should be csv file where for rasters, first column is pixel value and subsequent columns are attributes to attach; for vectors, columns are paired so that the first column matches the original attribute name and values, and the second column provides substitutions. | Ensure file exists and is in csv format. |
| <any layer definition> | lookup\_table | N | null/empty | Search for <layer filename>.csv in the same directory as the config file, then in the same directory as the spatial layer. It is valid to have no lookup table. |  |
| <any layer definition> | strict\_lookup\_table | N | boolean | Set strict mode for lookup tables associated with vector layers (default: False). Strict mode means that only values that appear in the lookup table are included in the final rasterized layer, otherwise the polygons are skipped and become nodata. Non-strict mode uses the original attribute value when no substitution is provided in the lookup table. |  |
| <any layer definition> | (any mojadata library Layer args) | N | any | Any additional keyword arguments to pass directly to the underlying mojadata library Layer constructor. |  |
| <root> | resolution | N | decimal | Use as resolution for bounding box when tiling spatial layers. | resolution <= 0.01 and evenly multiplies to 0.01 |
| <root> | resolution | N | null/empty | Use 0.001 as default value. |  |
| <root> | aidb | Y | string | Use as path to AIDB. | Ensure file exists. |
| <root> | aidb | Y | null/empty | Required field, throw exception. |  |
| <root> | gcbm\_config\_templates | N | null/empty | Use default GCBM config file templates bundled with gcbmwalltowall. |  |
| <root> | gcbm\_config\_templates | N | string | Path to directory containing GCBM config file templates. |  |
| <root> | disturbance\_order | N | string | Use as path to disturbance order file, a text file with a list of disturbance type names in descending priority order; this list is used to order the sequence of disturbances if there are multiple disturbances in the same year on the same pixel. | Ensure disturbance types exist in the AIDB. |
| <root> | disturbance\_order | N | null/empty | A default disturbance order is generated based on disturbance code (default disturbance type ID), where lowest code/ID = highest priority. |  |
| <root> | yield\_table | Y | string | Use as path to yield table. | Ensure file exists and is csv/xls/xlsx. |
| <root> | yield\_table | Y | null/empty | Required field, throw exception. |  |
| <root> | yield\_interval | Y | int | Use as yield increment interval, in years. |  |
| <root> | yield\_interval | Y | null/empty | Required field, throw exception. |  |
| <root> | classifiers | Y | dict | Use top-level keys as classifier names, with details in the values. | At least one classifier defined. |
| <root> | classifiers | Y | null/empty | Required field, throw exception. |  |
| <classifier definition> | <layer definition> | Y | (layer def keys) | Use as spatial layer corresponding to classifier. | Same as layer definition validation. |
| <classifier definition> | <layer definition> | Y | null/empty | Required field, throw exception. |  |
| <classifier definition> | values\_path | N | string | Use as path to file containing all possible values for the classifier. | Ensure file exists and is csv/xls/xlsx. |
| <classifier definition> | values\_path | N | null/empty | Use yield table. |  |
| <classifier definition> | values\_col | N | string | Use as column name for reading from classifier values file. | Ensure column exists in values\_path. |
| <classifier definition> | values\_col | N | int | Use as column # for reading from classifier values files. | Ensure column exists in values\_path. |
| <classifier definition> | values\_col | N | null/empty | If values\_path only has a single column, use that; otherwise search for column in values\_path matching classifier name or spatial attribute; if not found, search for a column in values\_path where the column values intersect with this classifier's values from its spatial layer. |  |
| <classifier definition> | yield\_col | N | string | Use as column name linked to classifier in the yield table. | Ensure column exists in yield table. |
| <classifier definition> | yield\_col | N | int | Use as column # linked to classifier in the yield table. | Ensure column exists in yield table. |
| <classifier definition> | yield\_col | N | null/empty | Use values\_col if values\_path is the same as yield table path, or search yield table for column matching classifier name, or search for column where values are a subset of values from values\_path. |  |
| <root> | bounding\_box | N | string | Use as path to bounding box layer. | Ensure file exists. |
| bounding\_box | (layer def keys) | N | (layer def keys) | Use as layer definition for bounding box. | Same as layer definition validation. |
| <root> | layers | N | dict | Use as list of miscellaneous spatial layers to tile, where key = layer name (for the tiler), and values are either simple layer paths or full layer definitions. | Same as layer definition validation. |
| <root> | layers | N | null/empty | Layers section is technically optional, but will usually contain at least the initial\_age layer. |  |
| <root> | disturbances | N | dict | Use as list of disturbance layers to tile. Keys are paths to layers or glob file patterns, and values are disturbance layer definitions. |  |
| <root> | disturbances | N | null/empty | Disturbances section is optional. |  |
| <disturbance layer definition> | year | N | string | Use as name of attribute containing year of disturbance, or if special "filename" keyword is used, try to parse year from filename. | Ensure attribute exists in layer or year is parseable from filename. |
| <disturbance layer definition> | year | N | int | Use as specific year of disturbance for whole layer. | 4 digits, > 0 |
| <disturbance layer definition> | year | N | null/empty | Search for an attribute named "year"; if not found, search for the first attribute where all the unique values parse into years; or finally, try to parse the year from the filename. |  |
| <disturbance layer definition> | disturbance\_type | N | string | Use as name of attribute containing GCBM disturbance type. | Ensure attribute exists in layer. |
| <disturbance layer definition> | disturbance\_type | N | string | Use as specific GCBM disturbance type for whole layer. | Ensure disturbance type exists in AIDB. |
| <disturbance layer definition> | disturbance\_type | N | null/empty | Search for the first attribute where all the unique values appear in tblDisturbanceTypeDefault in the AIDB. |  |
| <disturbance layer definition> | age\_after | N | string | Use as name of attribute containing reset age. | Ensure attribute exists in layer. |
| <disturbance layer definition> | age\_after | N | int | Use as specific reset age for whole layer. | -1 (no change) or >= 0 |
| <disturbance layer definition> | age\_after | N | null/empty | Search for an attribute named "age\_after"; if not found and regen\_delay is not present, do not create a transition rule; if regen\_delay is present but age\_after is not, throw an exception. |  |
| <disturbance layer definition> | regen\_delay | N | string | Use as name of attribute containing regen delay. | Ensure attribute exists in layer. |
| <disturbance layer definition> | regen\_delay | N | int | Use as specific regen delay for whole layer. | >= 0 |
| <disturbance layer definition> | regen\_delay | N | null/empty | Search for an attribute named "regen\_delay"; if not found and age\_after is not present, do not create a transition rule; otherwise default to 0. |  |
| <root> | rollback | N | dict | Use as rollback configuration and enable spatial rollback. |  |
| <root> | rollback | N | null/empty | Rollback section is optional; if not present, no rollback is performed. |  |
| rollback | age\_distribution | Y | string | Use as path to age distribution JSON or Excel file for rollback tool. | Ensure file exists. |
| rollback | age\_distribution | Y | null/empty | Required field if rollback section is present; throw exception. |  |
| rollback | inventory\_year | Y | int | Use as global inventory vintage year for the rollback tool. | 4 digits, > 0 |
| rollback | inventory\_year | Y | string | Use as path to inventory vintage layer for the rollback tool. This layer should be tiled with the other layers, with the resulting layer passed to the rollback. | Ensure file exists. |
| rollback | inventory\_year | Y | <layer definition> | Use as full layer definition for the inventory vintage layer. | Same as layer definition validation. |
| rollback | inventory\_year | Y | null/empty | Required field if rollback section is present; throw exception. |  |
| rollback | rollback\_year | N | int | Use as year to roll back to. | 4 digits, > 0 |
| rollback | rollback\_year | N | null/empty | Use 1990 as default value. |  |
| rollback | prioritize\_disturbances | N | boolean | Set prioritize disturbances mode in the rollback tool (default: False). |  |
| rollback | prioritize\_disturbances | N | null/empty | Use False as default value (prioritize inventory). |  |
| rollback | single\_draw | N | boolean | Set single-draw mode in the rollback tool (default: False) - draws a single age per work unit requiring a draw from the age distribution, instead of a draw per pixel. |  |
| rollback | single\_draw | N | null/empty | Use False as default value (per-pixel age draw). |  |
| rollback | establishment\_disturbance\_type | N | string | Use as default establishment disturbance type for pixels which require one to be created due to a lack of historic disturbance information (default: Wildfire). | Ensure disturbance type exists in AIDB. |
| rollback | establishment\_disturbance\_type | N | null/empty | Use Wildfire as default value. |  |