# Wall-to-Wall Python library

## Introduction

The wall-to-wall library helps simplify and automate the workflow of GCBM simulations for a wall-to-wall Canadian framework. It is less general-purpose than the typical "Standalone Template" workflow found in the training materials and is not meant to replace those steps for more complicated projects, or projects with data that requires the full set of transformation and filtering options found in the tiler library.

## User-facing tools

Users interact with the wall-to-wall library through a Python command-line application. 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, configuring and running GCBM, and queueing a GCBM run on the cluster.

### Commands

walltowall build <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; overwrites the existing json config file unless output config file path is specified.

walltowall prepare <config file path>

* Using the project configuration in the config file, tile the spatial layers, generate the input database, run the spatial rollback, and configure the GCBM run.

walltowall run (local|cluster) <config file path>

* Run the specified project either locally or on the cluster.

## Configuration format

The configuration format for projects in 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 optional and is a shortcut for creating projects that conform to the general specifications of the selected builder, instead of having to manually fill out the 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/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

}

}

}

### 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] "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 to attach. 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 for <layer filename>.csv in the same directory as this configuration file, and then in the layer directory.

#### Required

"aidb": <path to AIDB>,

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

"yield\_interval": <yield table increment interval, in years>,

"classifiers": {

<classifier name>: {

<layer definition 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: classifier name>,

[optional] "yield\_col": <column name/# in yield table; default: classifier name,

or use values\_col if values\_path is the same as yield table path>

}

}

#### Optional

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

##### 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>

or

"bounding\_box": {

"layer": <path to layer>,

[optional] "attribute": <attribute to read; default: use first found>,

[optional] "filter": <attribute value to limit bounding box to; default: all non-null>

}

##### 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>

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 parse into years.

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

containing disturbance type>

If disturbance type is not specified, wall-to-wall will first search for an attribute named "disturbance\_type", then 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.

}

}

##### Rollback

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

"rollback": {

"age\_distribution": <path to age distribution JSON 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>

}

## 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 yt\_casfri.json

walltowall run cluster 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 the root directory, 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",

"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"

},

"Classifier2": {

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

}

},

"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": {

"disturbance\_type": "dist\_type"

}

}

}

After the fully-specified project configuration is authored, 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 yt\_casfri.json

walltowall run cluster yt\_casfri.json