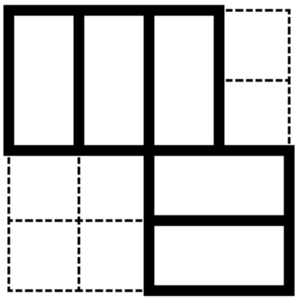
MDLR_trsrch / Users Manual

For App Version: 1.0.0.0 / 20221008

App Developers: Woojae Sung @ SA:PN:DA (www.sapnda.com/da)

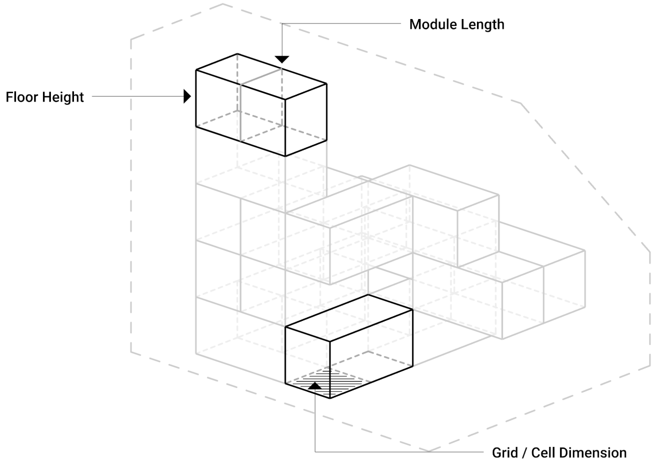
MDLR_trsrch [Modular TreeSearch]

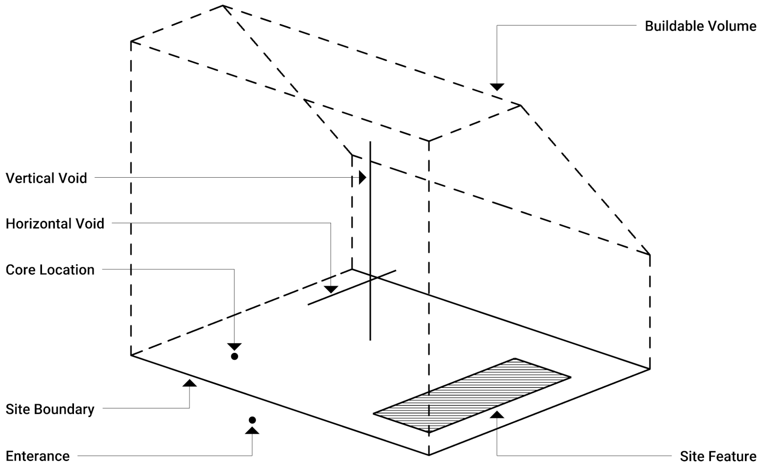


Welcome, and thank you for choosing **MDLR** plugin for Grasshopper!

MDLR_trsrch [Modular TreeSearch] calculates the 3d arrangement of modular buildings in real-time on any given site with design constraints and variables utilizing a tree search algorithm over the square grid system.

The plugin divides the site boundary into a grid system with a series of square cells. A module can be set with any number of consecutive square cells in one direction (the grid and the cell are of the same dimension). One can set the placement of the initial module on the site along with the design constraints such as site features(yard, parking, etc), voids, core location, and a buildable volume. Then the plugin calculates a feasible arrangement of the modules by recursively searching for the following modules' positions on the grid. One can also set design parameters such as the grid dimension, angle, module size, floor-to-floor height, FAR, BCR, max floor count, building height, etc.





The plugin calculates the modular arrangement on the ground floor until no available space is left or BCR is maxed out whichever comes first. Then it continues on to the floor above starting from the core location provided by the user constantly monitoring the BCR, FAR, max building height, max floor count, and buildable volume.

3d modular arrangement calculation is easy and intuitive: users can set the lot boundary, entrance, and core locations, and feed some optional inputs, then the plugin calculates the layout in real time. Users can also change design constraints and variables per their design intents.

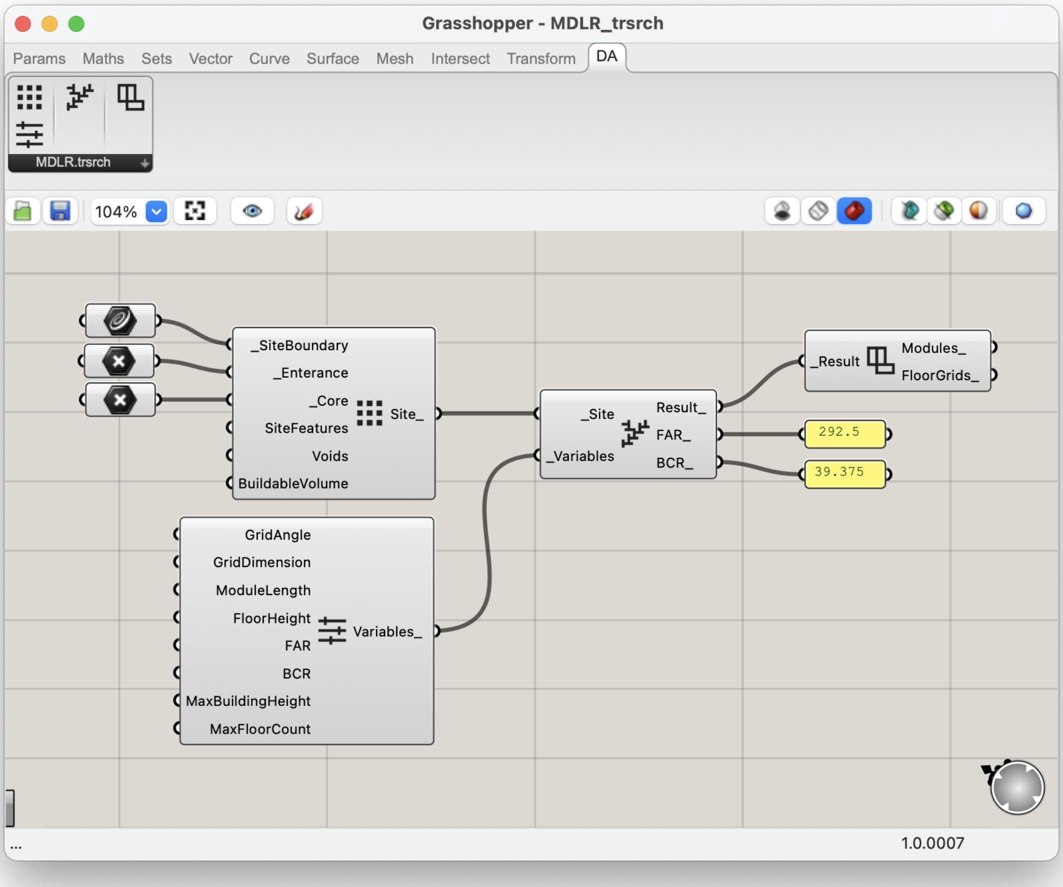
Installation

MDLR_trsrch support both Mac and PC versions of Rhino3d (v7 and above).

There are three ways to install the plugin: through the package manager built in Rhino3d, Food4Rhino/Grasshopper Apps, or manual installation as we used to.

- **Rhino3d built-in package manager**
    - Open Rhino3d, and type in “PackageManager” in the command line.
    - In the search box, look for modular and choose **MDLR_trsrch** from the search result.
    - Then hit the install button, and follow the on-screen instructions.
    - Close and reopen Rhino3d and Grasshopper.
- **Food4Rhino**
    - Go to https://www.food4rhino.com.
    - Go to Grasshopper Apps.
    - Search for modular treesearch.
    - Click download and follow the on-screen instruction.
    - For Mac users, this sometimes doesn’t work. Please stick with other installation methods in that case.
- **Manual installation**
    - Although we encourage doing it through the package manager, you still can stick with the old-fashioned.
    - Open Rhino3d and Grasshopper.
    - In the Grasshopper menu, go to File > Special Folders > Components Folder
    - Then copy and paste the downloaded gha file to the folder.
    - Make sure the file is “unblocked” in the file property.
    - Close and reopen Rhino3d and Grasshopper.

Overview



MDLR_trsrch lives under the **DA**[Design-Autonomy] shelf if successfully installed. The standard version has four components: Get Site, Get Variables, Compute Modular, and Visualize Modular. With just four simple steps, one can get the 3d modular building layout in less than a second.

Components

The Convention

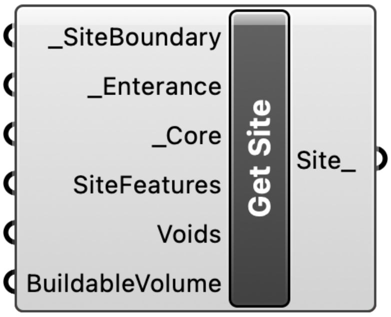
Component variable naming convention, inspired by LB/HB, has been adopted for intuitive usage. Any variable name starting with an underscore is an essential input for the component. One without an underscore means the variable is optional or has a pre-defined default value. Output variable names always end with an underscore. Connecting components is easy enough: match the same variable names (excluding the underscores).

The Unit

The default unit is set to meters. If you are working with other units, you can change variables accordingly at the “Get Variables” component below.

Get Site Information

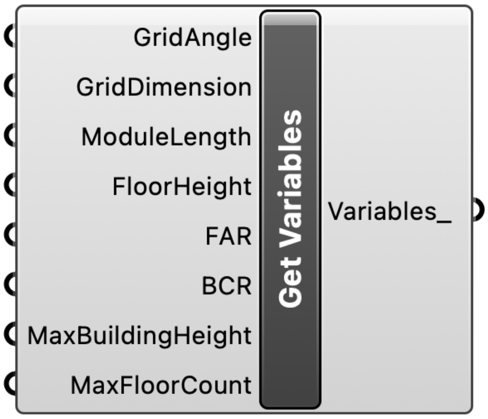
The component collects and sorts the site-related information for the modular calculation.



- **_SiteBoundary**: A closed polyline representing a lot boundary. The polyline must be planar.
- **_Enterance**: A point input to locate the first module at the ground level. The point can be either along or inside the site boundary.
- **_Core**: A point input to locate the core position. The point will be the start points of the floors above.
- **SiteFeatures**: Optional. A list of closed polylines representing site features such as parking stalls, yards, pools, decks, or some other area that cannot be occupied by the modules.
- **Voids**: Optional. A list of open/closed polylines representing voids in 3d space. This is useful to carve out certain modules to let air/light in through the modules.
- **BuildableVolume**: Optional. A closed Brep representing the buildable volume in 3d space. This is useful to constrain a building envelope set by zoning regulations.
- **Site_**: The result of the component.

Get Variables

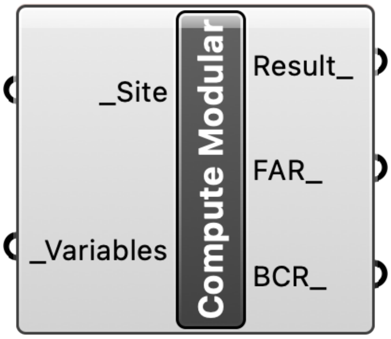
The component defines dimensional variables of the grid, module, floor-to-floor height, FAR(floor area ratio), BCR(building coverage ratio), max building height, and floor count. All the variables of the components are optional, meaning they are all set to the default values unless you want to change something.



- **GridAngle**: Optional. When working with a non-orthogonal site shape, you might want to align your grid to a specific angle. The default value is 0 degrees.
- **GridDimnsion**: Optional. The input determines the dimension of both the grid and the cell. The default value is set to 3 meters.
- **ModuleLength**: Optional. The integer input sets the longitudinal length(count) of cells of the module. The default value is 2.
- **FloorHeight**: Optional. The default value is 3.0 meters.
- **FAR**: Optional. The default value is 300%.
- **BCR**: Optional. The default value is 50%.
- **MaxBuildingHeight**: Optional. The input sets the max building height in meters. 0 means no limit. The default value is 0.
- **MaxFloorCount**: Optional. The input sets the max floor count. 0 means no limit. The default value is 0.
- **Variables_**: The result of the component.

Compute Modular

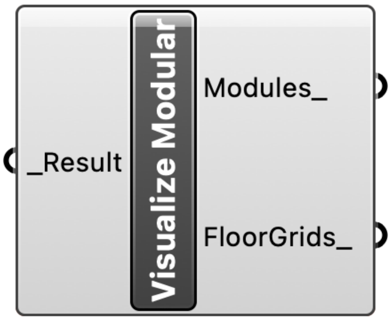
The component computes the 3d modular arrangement.



- **_Site**: Gets the result of the “Get Site” component.
- **_Variables**: Gets the result of the “Get Variables” component.
- **Result_**: The result of the component.
- **FAR_**: The resulting FAR. It differs from the FAR input (smaller or equal to the FAR input).
- **BCR_**: The resulting BCR. It differs from the BCR input (smaller or equal to the BCR input).

Visualize Modular

The component interprets the “Compute Modular” calculation result and produces 3D geometries, including floor grids and 3d modules in DataTree format.



- **_Result**: Gets the result of the “Compute Modular” component.
- **Modules_**: Returns a data tree of 3d modules(Brep).
- **FloorGrids_**: Returns a data tree of floor grids(Rectangle3d).

Feedback / Bug Report

We will be happy to get your feedback! Please let us know if you have any suggestions or any bugs to report. When reporting a bug, please include all the required files to reproduce the issue and send it over to ws@sapnda.com.