GRID CARTOGRAPHER 4

MAPPING MADE EASY

DATA EXPORT REFERENCE

Version 4.3.0

CHAPTER 1 – INTRODUCTION 3
Export Overview 4
XML Export 4
Mesh Export 4
Tile Data Models 5
Common Elements 5
Square Grid 5
Hex 'H' Grid 6
Hex 'V' Grid 6
CHAPTER 2 – XML FORMAT SPECIFICATION 7
Element Hierarchy 8
Element Reference 9
CHAPTER 3 – MESH EXPORT 20
Export Options 21
Compatibility 21
Geometry Adjustment 21
Materials / Sub-meshes 22
Edge Style Material Suffixes 22
Terrain Type Material Suffixes 24
APPENDIX A – TABLES 26
Edge Table 27
Marker Table 28
Terrain Table 34

CHAPTER 1 – INTRODUCTION

Table of Contents

Export Overview **4**
XML Export 4
Mesh Export 4
Tile Data Models **5**
Common Elements 5
Square Grid 5
Hex ‘H’ Grid 6
Hex ‘V’ Grid 6

Export Overview

Grid Cartographer can export maps for use in game development and visualization applications. The export commands can be found in the *File* menu on the *Export Mesh* and *Export Data* pages.

Export features are only available in the Professional Edition of the software.

XML Export

Use the *Export Map As XML* button on the *Export Data* page to begin the export process. A file selector will open, choose the destination and file name and click OK. The export process will then begin and once completed you will be notified of success or failure with a message box.

The specification of the XML is described in Chapter 2.

Mesh Export

The floors of the map can be exported as simple flat-shaded 3D meshes. This can be useful, for example, when initially blocking-out environments for game development. Meshes are exported to standard Wavefront .obj format which is compatible with most 3D editors as well as the Unity and Unreal Engine game engines.

To export a mesh, switch to the main editor and navigate to the floor or region you wish to export. Then select the *File* tab, *Export Mesh* menu, select your options and then click the *Export Mesh* button. This will open a standard file selector and begin the export process. When complete you will be prompted with a confirmation message box.

For more information see Chapter 3

Tile Data Models

Map grids are represented internally using a data model dependent on the shape of the grid. Below are diagrams of how data fields of standard region types in exported data relate to the map displayed in the editor.

Common Elements

Each tile holds the following common fields regardless of the shape of the grid:

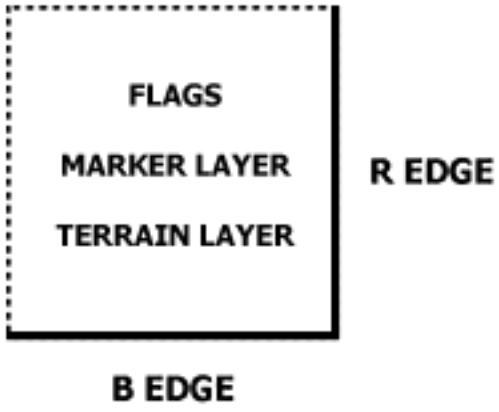
Flags: which specify which of the field effect colors (red, green or blue), darkness and/or ceiling attributes have been applied to the tile.

Marker Layer: Specifies the marker used in this tile. See the marker table in Appendix A for a complete list.

Terrain Layer: Specifies the terrain style used in the tile. See the terrain table in Appendix A for a complete list.

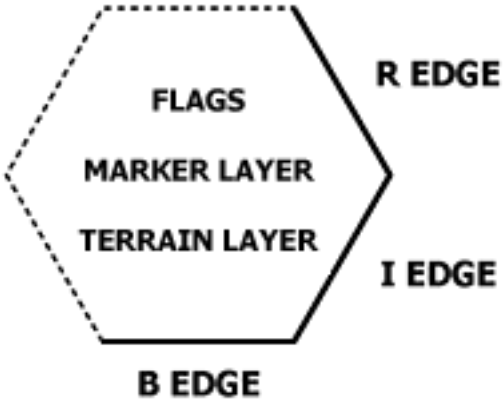
Square Grid

The square grid model specifies a right and a bottom edge for the tile. To make a full square requires additional tiles to the left and above this one.



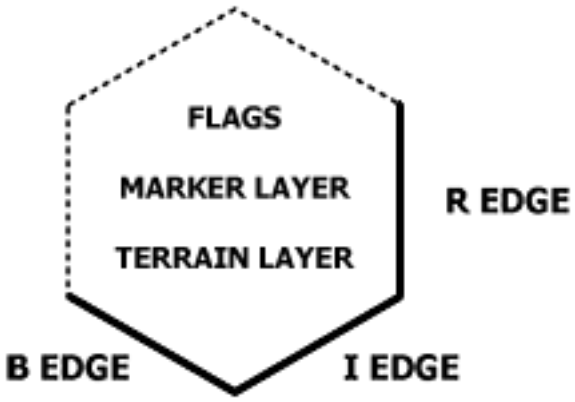
Hex ‘H’ Grid

Hexagon grids introduce the ‘I’ (intermediate) edge. For the horizontal hexagon grid type this specifies an additional edge on the right side.



Hex ‘V’ Grid

In vertical hexagon grids the intermediate edge specifies an additional bottom edge.



.

CHAPTER 2 – XML FORMAT SPECIFICATION

Table of Contents

Element Hierarchy **8**
Element Reference **9**

Element Hierarchy

The XML document uses the following elements arranged in the hierarchy below. Depending on the content of the map you're exporting, some elements may not be present.

Each element is described in detail in the next section.

```
<map>
|---  <export>
|---  <region>
|     |---  <name>
|     |---  <setup>
|     \---  <floor>
|           |---  <name>
|           \---  <tiles>
|                 |---  <bounds>
|                 \---  <row>
|                       \---  <t>
|                 <notes>
|                 |---  <note>
|                 \---  <label>
|---  <tilemap>
|     |---  <name>
|     |---  <setup>
|     |---  <mapsize>
|     \---  <plane>
|           |---  <name>
|           |---  <tiles>
|           |     \---  <row>
|           |           \---  <c>
|           \---  <notes>
|                 |---  <note>
|                 \---  <label>
|---  <custom>
|     |---  <monochrome>
|     |     \---  <tile>
|     |           \---  <name>
|     \---  <color>
```

```
|           \---  <tile>
|                 \---  <name>
\---  <palette>
      \---  <entry>
```

Element Reference

<map>

This is the root container element of the document.

<export>

Records the date, time and version number of Grid Cartographer used when the document was exported. It has the following attributes:

Attribute	Meaning
`from`	The friendly version number of Grid Cartographer used to export the document. This takes the form `v#.#.#`.
`build`	The build number of Grid Cartographer used to export the document. This is guaranteed to increase with each version of the software and can be used for comparisons.
`date`	The date of export in `YYYY-MM-DD` format.
`time`	The (local) time of export in `HH:MM:SS` format.

<region>

This is the container element for a standard map region. Multiple regions are supported by the editor within one map file. Each region has the following attributes:

Attribute	Meaning
`floors`	The floor count of this region. Excluding ground floor if disabled.
`lowest_floor`	The index of the lowest floor. Negative values are basements.
`grid_shape`	The shape of the grid used for all floors in this region. The value can be one of: `square`, `hexh` or `hexv`.

<tilemap>

This is the container element for a tilemap. Multiple tilemaps are supported by the editor within one map file. Each tilemap element has the following attributes:

Attribute	Meaning
`planes`	The number of planes in this tilemap.
`lowest_plane`	The index of the lowest plane in the stack. Plane indices start at 1.
`grid_shape`	The shape of the tilemap grid. Reserved for future expansion – as of this version, the value will always be: `square`.

<name>

The name of the region or tilemap (UTF-8 encoded) is stored in a child CDATA element.

<setup>

Provides information about the map setup. It has the following attributes:

Attribute	Meaning
`origin`	Either `tl` or `bl` which, respectively, specify either a top-left or bottom-left grid origin.

<mapsize>

Defines the size of all planes in a tilemap.

Attribute	Meaning
`width`	The number of tiles in each row.
`height`	The number of rows.

Note: This element is only present on a tilemap, see `<bounds>` below to determine the size of a floor in a region.

<floor>

A floor in the region. Contains the tiles and notes you've created.

Attribute	Meaning
`index`	The number of the floor. Negative values are basements, zero is the ground floor and positive values are the floors above.

Note: A completely empty floor will not be exported. Take care to use the `index` attribute to correctly identify a floor.

<plane>

A plane in the tiles. Contains the tiles and notes you've created.

Attribute	Meaning
`index`	The index of the plane. Plane indices start at 1.

Note: Even a completely empty plane will still be exported (unlike `<floor>` elements).

<name>

If the floor or plane name has been changed, this element stores its name (UTF-8 encoded) in a child CDATA element.

<tiles>

Container for tiles on this floor/plane. Floors, if no tiles are present, will omit this element. Planes will always include this element.

<bounds>

Defines the maximum bound of tiles on this floor. This element is not included for tile map planes, see `<mapsize>`.

Attribute	Meaning
`x0`	The x co-ordinate of the left-most occupied tile.
`y0`	The y co-ordinate of the top or bottom most occupied tile as defined by the co-ordinate space specified by the origin attribute of the `<setup>` element.
`width`	The number of tiles in each row.
`height`	The number of rows from y0 to the last occupied row on this floor.

<row>

Contains one row on this floor or tile map plane. If this row is part of a floor then it will be omitted if it contains no tiles. Tile map rows are always exported.

Attribute	Meaning
`y`	The y co-ordinate of the row in the co-ordinate space specified by the origin attribute of the `<setup>` element.

<t>

Data for a single tile on a standard map region (not a tilemap).

Attribute	Meaning
`m`	A standard marker is present. See the marker table in Appendix A for the full list of supported icons.
`msub`	Marker sub-data. This is only present for some marker types. See the corresponding entry in the marker table appendix for further details.
`mcm`	A custom monochrome marker is present. This is a value from 0 to 65000 corresponding to the index of the custom tile used in the monochrome list. See the `<custom>` element below for more information.
`mcc`	A custom color marker is present. This is a value from 0 to 65000 corresponding to the index of the custom tile used in the color list. See the `<custom>` element section below for more information.
`mc`	The color of the marker layer. This is a palette index from 0 to 255. See the `<palette>` element below for more information. Note that color custom markers are not tinted and should ignore this value.
`t`	A standard terrain type is present. See the terrain table in Appendix A for the full list of supported terrain types.
`tcm`	A custom monochrome terrain is present. This is a value from 0 to 65000 corresponding to the index of the custom tile used in the monochrome list. See the `<custom>` element below for more information.

tcc	A custom color terrain is present. This is a value from 0 to 65000 corresponding to the index of the custom tile used in the color list. See the <custom> element section below for more information.
tc	The color of the terrain layer. This is a palette index from 0 to 255. See the <palette> element section below for more information. Note that color custom terrain is not tinted and should ignore this value.
r	The style of the R edge of this tile. See the edge table in Appendix A for the full list. Also see the tile data model section for the location of the R edge for the grid shape used.
i	The style of the I edge of this tile. See the edge table in Appendix A for the full list. Also see the tile data model section for the location of the I edge for the grid shape used.
b	The style of the B edge of this tile. See the edge table in Appendix A for the full list. Also see the tile data model section for the location of the B edge for the grid shape used.
rc	The color of the R edge. This is a palette index from 0 to 255. See the <palette> element section below for more information. See the tile data models (in Chapter 1) for the location of the R edge for the grid shape used.
ic	The color of the I edge. This is a palette index from 0 to 255. See the <palette> element section below for more information. See the tile data models (in Chapter 1) for the location of the R edge for the grid shape used.
bc	The color of the B edge. This is a palette index from 0 to 255. See the <palette> element section below for more information. See the tile data models (in Chapter 1) for the location of the R edge for the grid shape used.
d	If set to 1 this signifies the tile is 'dark'.
fx	A value composed from characters r, g and/or b that specify which of the three colored FX flags have been assigned to this tile. Multiple characters can be present and assigned to the tile.
c	If set to 1 this signifies the tile has a ceiling.

`sp`	A string of characters that indicate special attribute flags assigned to this tile. Multiple characters can be present and will always appear in the order listed below. Meanings are: `h` Tile is horizontally flipped (custom tiles only). `v` Tile is vertically flipped (custom tiles only).
`snip`	If the terrain snipper tool has been used on this tile to remove some part of the ground, this element is present. It can have one of two values: `tl` or `br` which represent whether the top/left or bottom/right of the tile is still visible.

<c>

A single tilemap cell. These are present under `<plane>` elements. Note: All attributes are optional and if not present are assumed to be zero (unless otherwise specified).

Attribute	**Meaning**
`i`	The index of the custom tile used by this cell. A value from 0 to 65000. Only color custom tiles can be used with tilemaps. This attribute will be omitted for empty cells and the cell can be interpreted as having an empty or default appearance if required.
`sp`	A string of characters that indicate special attribute flags assigned to this tile. Multiple characters can be present and will always appear in the order listed below. Meanings are: `h` Tile is horizontally flipped. `v` Tile is vertically flipped.

<notes>

Container element for notes and labels on this floor. If no notes or labels were created this element will be omitted.

<note>

Multiple note entries can be added to each map floor/plane. If multiple notes are assigned to a single tile location, they will remain as separate entities. This element represents one of those notes.

Notes are not stored in any particular order, even those assigned to the same location may not be grouped together in the exported XML document.

Note elements have the following attributes:

Attribute	Meaning
`x`	The X co-ordinate of the note.
`y`	The Y co-ordinate of the note given in the co-ordinate space specified by the origin attribute of the `<setup>` element.

The text itself is stored in a child CDATA element of the `<note>` element.

<label>

Multiple label entries can be added to each map floor/plane, but only one label is permitted at each tile location. Labels are not stored in any particular order. The label text itself is stored as a child CDATA element of the `<label>` element.

Attribute	Meaning
`x`	The X co-ordinate of the label.
`y`	The Y co-ordinate of the label given in the co-ordinate space specified by the origin attribute of the `<setup>` element.
`halign`	The horizontal alignment mode of the label. One of `left`, `center` or `right`. If this attribute is omitted, centered alignment is default.
`valign`	The vertical alignment mode of the label. One of `top`, `middle` or `bottom`. If this attribute is omitted, middle alignment is default.
`c`	The text color of the label. A palette index from 0 to 255.

<custom>

Container element for custom tile information. If no custom tiles have been added to the map then this element is omitted.

<monochrome>

Additional information about monochrome custom tiles is stored within this element. If there aren't any custom tiles of this type then the element is omitted.

<color>

Additional information about color custom tiles is stored within this element. If there aren't any custom tiles of this type then the element is omitted.

<tile>

Describes a single custom tile added to the map.

Attribute	**Meaning**
`index`	The index number of the custom tile. See the `<t>` element above which references this value with its `mcm`, `mcc`, `tcm` or `tcc` attributes.
`width`	The width, in pixels, of the custom tile image.
`height`	The height, in pixels, of the custom tile image.
`unused`	If set to `1` this signifies that the custom tile is available but is not current used on any of the map floors.

The name of the custom tile is stored as a child CDATA element. The default name for a custom tile is the file name of the imported image (with extension)

<palette>

Container element for palette entries used by this map. This element is present only if the user checks the *Include Color Palette* option on the *Export Data* menu.

<entry>

Describes a color palette entry.

Attribute	Meaning
`i`	The index number of the custom tile. See the `<t>` element above which references this value with its `mcm`, `mcc`, `tcm` or `tcc` attributes.
`rgb`	The color value of the entry expressed in HTML notation `#RRGGBB`.
`edit`	If set to `1` this indicates a color that has been edited.

.

CHAPTER 3 – MESH EXPORT

Table of Contents

Export Options **21**
Compatibility 21
Geometry Adjustment 21
Materials / Sub-meshes 22
Edge Style Material Suffixes **22**
Terrain Type Material Suffixes **24**

Export Options

The exported mesh can be configured using a number of options provided on the *Export Mesh* menu.

Include All Floors: With this option checked, every floor in the current region will be exported to the mesh file. If unchecked, only the current floor will be included.

Compatibility

Select from three target modes to apply specific compatibility options to the exported mesh. These options are purely to make importing into other applications more convenient, they do not represent significant changes and can be changed further using import options on the target side.

3D Editor: Export using the standard 'right handed' co-ordinate system used by many popular 3D editing packages. In this system the x-axis points to the right, y-axis points up and the z-axis points towards the viewer. Note: Blender's default import settings for .obj files applies the correct adjustments to convert into its z-up co-ordinate system.

Unity: Export using Unity compatible co-ordinate system (x-axis to the right, y-axis is up and z-axis is forward/away). In this mode a single square tile is 1 unit in size.

Unreal Engine: Export using the Unreal Engine editor co-ordinate system (x-axis is right, y-axis towards and z-axis is up). It also applies a 100x scale to convert from centimeters to meters.

Geometry Adjustment

Offset Mesh by Floor Height: By default, the mesh is exported with the bottom of the mesh at 0 units and the top at 1 unit. If the option is enabled, the mesh is moved up or down by the number of units corresponding to the number of the floor or basement being exported. This allows multiple floors to be combined in one scene without having to adjust the individual origin points.

Double Sided Edge Polygons: When enabled the polygons used to create edge geometry are duplicated so the edge appears solid from both sides. When disabled only one polygon is created per-edge, you will need to render with face culling disabled to see the edges from both sides. This option may be helpful when exporting the map for use as collision geometry.

Triangulate: This option forces the geometry output to only contain triangles. By default, this option is not enabled and the mesh will contain both quads and hexagons.

Materials / Sub-meshes

The mesh exporter provides several options to control the naming of materials and the subsequent creation of sub-meshes in the exported model. The options are:

Create Materials / Sub-meshes: This is the master control for material naming. If enabled the other check-boxes will also become available for fine-tuning. If disabled, the mesh will be exported as a single piece with one material named default applied to every polygon.

One per- Color Tint: Append `_col#` to the material names of all edges and floor tiles that use a color tint. If disabled then all color will be ignored. The # value ranges from 1 to 255.

One per- Edge Style: If enabled, append the style of edge to the material name used (see below for the full list). For example: a locked door will use the prefix `gc_edge_locked`. If the option is disabled then all edges will use the same `gc_edge` prefix for their material name.

One per- Terrain Type: If enabled, the specific type of terrain will be used to create the material name (see below for the full list). For example: a lava tile will be assigned `gc_floor_lava`. If disabled then all terrain types will use the material name `gc_floor`.

Note that ceilings will always use the prefix `gc_ceil` (unless Create Materials / Sub-meshes is disabled). If the 'one per-color tint' option is enabled the color of the terrain (if not default) will be used to create the `_col#` suffix.

Edge Style Material Suffixes

The right-hand column gives the text appended to the material name when the 'One per- Edge Style' option is enabled.

Edge Style	Material Suffix
Plain wall	`wall`
Standard door	`door`
Locked door	`locked`
Hidden door	`hidden`
One-way door (facing left or up)	`oneway_door_lu`

One-way hidden door (left/up)	`oneway_hidden_lu`
One-way wall (left/up)	`oneway_wall_lu`
One-way door (facing right or down)	`oneway_door_rd`
One-way hidden door (right/down)	`oneway_hidden_rd`
One-way wall (right/down)	`oneway_wall_rd`
Empty frame	`frame`
Secret wall	`secret_wall`
Trapped door	`door_trapped`
Half-door (left side)	`door_half_left`
Half-door (right side)	`door_half_right`
Half-wall (left side)	`wall_half_left`
Half-wall (right side)	`wall_half_right`
Button wall (facing left or up)	`btn_lu`
Button wall (facing right or down)	`btn_rd`
Torch wall (facing left or up)	`torch_lu`
Torch wall (facing right or down)	`torch_rd`
Lever (facing left or up)	`lever_lu`
Lever (facing right or down)	`lever_rd`
Bars	`bars`
Double-sided torch wall	`torch_pair`
Gate	`gate`
Message wall	`rune`
Secret door	`secret_door`
Niche (facing left or up)	`niche_lu`

Niche (facing right or down)	`niche_rd`
Keyhole wall	`keyhole`
Door (box style)	`door_box`
Corner curve	`wall_corner`

Terrain Type Material Suffixes

The right-hand column gives the text appended to the material name when the 'One per-Terrain Type' option is enabled.

Terrain Type	**Material Suffix**
Block	`block`
Inside	`inside`
Outside	`outside`
Water	`water`
Lava	`lava`
Mountain	`mountain`
Rock	`rock`
Trees	`trees`
Vegetation	`veg`
Sand	`sand`
Snow	`snow`
Track	`track`
Wood	`wood`
Ooze	`ooze`

Metal	`metal`

.

APPENDIX A – TABLES

Table of Contents

Edge Table **27**
Marker Table **28**
Terrain Table **34**

Edge Table

This is a list of edge styles sorted by their code value.

Value	Edge Style
0	None / empty
1	Wall
2	Standard door (capacitor style)
3	Locked door
4	Hidden door
5	One-way door (exiting left or up)
6	One-way hidden door (exiting left or up)
7	One-way wall (exiting left or up)
8	One-way door (exiting right or down)
9	One-way hidden door (exiting right or down)
10	One-way wall (exiting right or down)
12	Empty door frame
13	Secret wall
14	Trapped door
15	Half door (left side)
16	Half door (right side)
17	Half wall (left side)
18	Half wall (right side)
19	Button (facing left or up)
20	Button (facing right or down)
21	Torch (facing left or up)

22	Torch (facing right or down)
23	Lever (facing left or up)
24	Lever (facing right or down)
25	Bars
26	Torch (double sided)
27	Gate / portcullis
28	Message / rune wall
29	Secret Door
30	Niche (facing left or up)
31	Niche (facing right or down)
32	Keyhole wall
33	Standard door (box style)
34	Trapped wall

Marker Table

This is a list of markers sorted by their code value.

Value	Edge Style
0	None / empty
1	Stairs Up
2	Stairs Down
3	NPC
4	Teleport In
5	Teleport Out

6	Rotating Room
7	Pit (Open)
8	Death
9	Start
10	Exit
11	Turntable
12	Treasure chest (open)
13	Key
14	Monster
15	Switch
16	Fountain
17	Save point
18	Target
19	Pressure
20	Pentagram
21	Elevator
22	Zap!
23	Unknown
24	Event
25	Message
26	Ladder up
27	Ladder down
28	Block edge (horizontal). The `msub` attribute specifies the edge style. See the Edge Table above.

29	Block edge (vertical). The msub attribute specifies the edge style. See the Edge Table above.
42	Block edge (diagonal left '\'). The msub attribute specifies the edge style. See the Edge Table above.
43	Block edge (diagonal right '/'). The msub attribute specifies the edge style. See the Edge Table above.
44	Ladder (two ways)
46	Treasure chest (closed)
47	Treasure chest (trapped)
48	Treasure chest (locked)
49	Ore
50	Pit (covered)
51	Pit (trapped)
52	Well
20	Pentagram
21	Elevator
22	Zap!
23	Unknown
24	Event
25	Message
26	Ladder up
27	Ladder down
28	Block edge (horizontal). The msub attribute specifies the edge style. See the Edge Table above.
29	Block edge (vertical). The msub attribute specifies the edge style. See the Edge Table above.

42	Block edge (diagonal left '\'). The `msub` attribute specifies the edge style. See the Edge Table above.
43	Block edge (diagonal right '/'). The `msub` attribute specifies the edge style. See the Edge Table above.
44	Ladder (two ways)
46	Treasure chest (closed)
47	Treasure chest (trapped)
48	Treasure chest (locked)
49	Ore
50	Pit (covered)
51	Pit (trapped)
52	Well
53	Triangle
54	Small square
55	Square
56	Small circle
57	Circle
58	Diamond
59	Emerald
60	Ruby
61	Crystal
62	Arrow up
63	Arrow right
64	Arrow down
65	Arrow left

66	Sack
67	Map
68	Purse
69	Barrel
70	Ramp up
71	Ramp down
72	Boulder
73	Stone
74	Pressure plate + stone
78	Arrow left and right
79	Arrow up and down
80	Arrow diagonally up and left
81	Arrow diagonally up and right
82	Arrow diagonally down and right
83	Arrow diagonally down and left
97	Tree
98	Shop
99	Bed
100	Tavern
101	Health
102	Moveable block
103	Trainer
104	Skull
105	Bones

106	Boat
107	Bridge
108	Signpost
109	Pillar
110	Armor
111	Grave
112	Statue
117	Arrow up then left
118	Arrow up then right
119	Arrow down then left
120	Arrow down then right
121	Arrow left then up
122	Arrow right then up
123	Arrow left then down
124	Arrow right then down
125	Weapons
126	Boots
127	Altar
128	Food
129	Corner. The `msub` attribute specifies the corner style. `0` = North-west Quadrant, `1` = North East, `2` = South West, `3` = South East.
130	Scroll
131	Book
132	Harvest Plant
133	Timber Pile

134	Doorway
135	Tent
136	Spring (jump pad)

Terrain Table

Below is a list of terrain types sorted by their code value.

Value	Edge Style
0	None / empty
31	Inside (classic)
32	Outside (classic)
33	Water (classic)
34	Lava (classic)
35	Rock (classic)
36	Vegetation (classic)
38	Ooze (classic)
39	Block
40	Sand (classic)
41	Wood (classic)
75	Metal (classic)
76	Trees (classic)
95	Snow (classic)
113	Mountain (classic)
115	Track (classic)

.