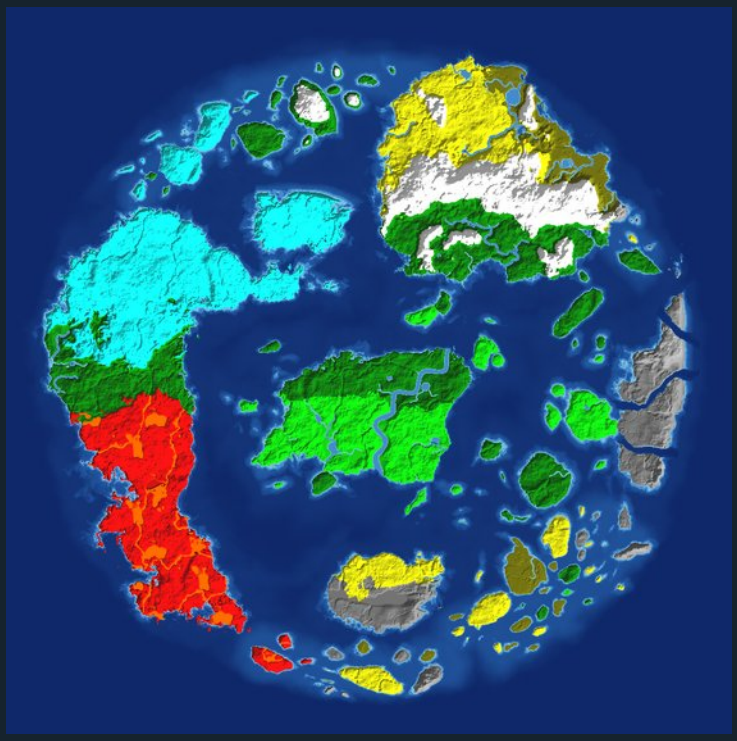


Better Continents Map-Making

A skill for AI agents

Author a Valheim world that reads as natural geography and gives every game location room, and prove both before a human plays it.

Written by Wubarrk, with Claude (Anthropic) as the drafting agent

Learned on the ProximaMaxi realm map: thirteen versions and four test worlds, 22 to 23 September 2026

For Valheim 1.0 and Better Continents 0.9.x (github.com/RGlabs84/BetterContinents10)

For an agent, Claude, Gemini or another, with a shell on the machine that holds the game, the mod and the toolkit

Licensed under the GNU General Public License, version 3, WITHOUT ANY WARRANTY. The licence text is the final appendix; the Markdown source, the build scripts and the toolkit are attached to this PDF (AI-skill-source.zip).

Contents

About this document 5

License 6

1 The skill 7

What you are being asked to do 7

How the work goes 7

The contract with the human 7

The workspace 8

2 How images become a world 10

The square, the pixel and north 10

The files 10

The height maths 11

The biome map 12

The heat map 12

The lava map 13

The forest map 13

The location map 13

The alt-biome map (0.8.1) 14

Baking, the config and the world 14

3 What the game demands of a map 16

The game's grid 16

Regions and alt biomes 16

Locations 17

What the important locations want 18

4 What "reads natural" means 21

The tells, and the fixes 21

Relief that reads, in numbers 22

The reference world 22

Where the human's taste showed 23

5 The generator, as a pattern 24

Skeleton 24

The parts 24

Changing the map . . . 25

6 The workflow . . . 27

Step 0: know the state . . . 27
Step 1: agree the change . . . 27
Step 2: snapshot before you touch anything . . . 27
Step 3: change the generator . . . 27
Step 4: generate at full size . . . 28
Step 5: measure . . . 28
Step 6: render, and show the human . . . 29
Step 7: publish the version . . . 29
Step 8: stage into the game profile . . . 29
Step 9: the human builds a NEW world . . . 29
Step 10: read the log . . . 30
Step 11: replay the log . . . 30
Step 12: hand off . . . 31

7 The traps that cost worlds . . . 32

8 A validated build, end to end . . . 35

The generator (gen.log) . . . 35
check.py . . . 36
relief.py . . . 36
altbiomes.py . . . 36
locations.py . . . 37
The game (log-13-ProxEra01.txt) . . . 38
The replay (locations.py --log) . . . 38

9 Export and import . . . 40

bc_export: what it writes . . . 40
Cutting and pasting between an export and a generated map . . . 41
Presets and bc_import . . . 42
Biome precision . . . 43
Multiplayer: the settings download . . . 44
Traps . . . 44

Appendix A Tool reference . . . 46

What the checkers reproduce, and what they cannot . . . 47
When the game changes . . . 47

Appendix B The config that goes with the map 48

The Gale profile 48

Biome precision and the Export group (0.9.0, not written by the staging script) 48

The world 49

Appendix C The game's own numbers 50

Every enabled location, in the game's placement order 50

The 32 alt biomes and their placement windows 54

The biome legend 55

ProximaMaxi's alt-biome legend (version 13) 56

Appendix D Glossary and sources 57

Glossary 57

Sources, and how to check a claim 58

The map's versions, as a study 58

Appendix E GNU General Public License, version 3 60

About this document

This is a training document for an AI agent: Claude, Gemini, or any other model that can run commands on a computer. It teaches one skill, exactly as it is practised in Wubarrk's Better Continents workspace:

> **Rule:** Author Better Continents image maps for Valheim so that the world reads as natural geography AND every location the game wants to place has room, and prove both before a human plays it.

It is written for Valheim 1.0 and Better Continents 0.9.x, the fork maintained by Wubarrk at github.com/RGlabs84/BetterContinents10. Everything in it was learned on one real project, the ProximaMaxi realm map, between 22 and 23 September 2026: thirteen map versions, four test worlds, and every check written because a world had already been lost without it.

How to use it, if you are the agent:

1. Read chapters 1 to 3 in full before touching anything. They are the model of how the game and the mod turn images into a world, and the demands the game makes of a map. Nothing later makes sense without them.
2. Chapter 4 is the design goal. Chapter 5 is how the generator is built, so that you can change it or write your own. Chapter 6 is the workflow, step by step, with the exact commands and the acceptance criterion for each step. Do the steps in order and never skip a check.
3. Chapter 7 is the list of traps. Each one cost a test world. Read it before your first build and again whenever a result looks wrong.
4. Chapter 8 walks through a real validated build, so you know what good numbers look like.
5. Chapter 9 is the other way to start a map: exporting a real world (vanilla or Better Continents) instead of, or alongside, `gen.py`, and turning a folder back into a New World with `bc_import`.
6. The appendices are the reference: the tools, the config, the game's own placement numbers, a glossary, and where every fact came from.

The toolkit (the generator and every checker) is attached to this PDF as a zip; see the License page. On Wubarrk's machine it lives at `/home/rohan/WubarrkCODING/BetterContinents-TestMaps/`, and the paths in this document are that machine's. On another machine, unpack the zip and substitute your paths.

What this document is not: it is not the Better Continents guide. The guide (`BetterContinents-Guide.pdf`, the edition for Valheim 1.0 and Better Continents 0.9.x) documents every setting and map type of the mod, and a separate `BetterContinents-Export-Guide.pdf` walks a newcomer through exporting and importing. This document assumes the mod works as the guide says and teaches how to make a map that the game will accept and a human will enjoy.

License

Better Continents map-making: a skill for AI agents was written by Wubarrk, with Claude (Anthropic's model) as the drafting agent, and is released under the **GNU General Public License, version 3** (GPL-3.0), the licence of the Better Continents Guide it accompanies, so that the two documents can be shipped and modified together under one licence. Copyright 2026 Wubarrk.

It is distributed **WITHOUT ANY WARRANTY**; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. The full text of the licence is reproduced in the final appendix, GNU General Public License, version 3.

The tools attached to this PDF are Wubarrk's and carry the same licence. The two data files `locations_1.0.15.json` and `altbiomes_1.0.15_full.json` are tables of placement parameters read out of Valheim's game data by the attached extractors: facts about the game, reproduced for interoperability. Valheim is the property of Iron Gate Studio.

Better Continents is LGPL-2.1 (created by billw2012, maintained by Jere Kuusela from 2023 to 2026 and by Wubarrk since September 2026). The Better Continents Guide is GPL-3.0 (written by billw2012). Neither is reproduced here beyond short statements of behaviour.

Attachment. The file `AI-skill-source.zip` is attached to this PDF; open your PDF viewer's attachments panel to save it. It contains:

- `docs/`: the Markdown source of this document and its images;
- `build_ai_skill_pdf.py` and `build_pdf.py`: the build scripts (the second is the guide's renderer, which the first reuses), and `make_c_data.py`, which writes appendix C from the game data;
- `LICENSE`: the GPL-3.0 text;
- `toolkit/`: the generator, the checks, the renderers, the extractors and the data they extracted, the staging script, and the ProximaMaxi map's README and legends, laid out as they are in the workspace.

1 The skill

What you are being asked to do

A Better Continents world is made from a handful of images: a heightmap, a biome map, a heat map, a forest map, a lava map, a location map and, since Better Continents 0.8.1, an alt-biome map. The mod reads them once, when a world is created, and bakes them into the world. The game then places its several hundred location types (bosses, dungeons, villages, wrecks, altars) on that terrain by its own rules, and chooses which of its 32 alt biomes go on which regions, by its own rules.

The skill has three halves that pull against each other:

1. **The world must read as natural geography.** Not blobs and stripes: fractal coasts, ragged biome fronts, a mountain crest that wanders, rivers that drain, rolling ground, cliffs where the wind hits. The human judges this from pictures. Chapter 4.
2. **Every location the game wants must have room.** Hildir's cavern needs mountain 200 m up; the frozen ships need Deep North water 5 to 15 m deep, 8 to 9.75 km from the centre; Bonemass wants flat swamp under 2 m above the water; the goblin towers want 100 plains sites 512 m apart. A beautiful map that breaks the game's progression is a failed map. Chapter 3.
3. **It must be proved before a human plays it.** A world takes a human minutes to build and hours to judge, and a map bug (the wrong waterline, a stray heat pixel, the wrong height scale) is invisible in a picture and total in the game. Every claim in this document comes with the check that proves it. Chapter 6.

How the work goes

You do not draw the images by hand. You write and run a **generator** (a Python program that evaluates the world at every pixel and writes the PNGs), you **measure** its output with the checkers, you **render** a preview and **show it** to the human, the human **steers**, and you iterate. When the human is satisfied you **stage** the map into the game's mod profile, the human builds a **new world**, you read the game's **log** and **replay** it with the checker to confirm that the game agreed with your prediction. The full loop is chapter 6.

The unit of work is a map **version**. Every version is a snapshot of the generator and its output in `generator/history/`, numbered, with the numbered previews the human saw in `previews/`. ProximaMaxi went through 13 versions in two days; the human looked at every one.

The contract with the human

These are Wubarrk's working rules. Each was stated after something went wrong, so none is optional.

> **Human: Show a picture after every meaningful change**, with a one-line caption of what changed and what is next. Do not batch several rounds of changes before showing anything. The human steers the design from the pictures ("i like it 06", "refine and work off of it"); a picture they have not seen is a decision they could not make. Deliver pictures through a channel that actually reaches them. For Wubarrk that is an Artifact page (claude.ai/artifact/QW4ySXeGTiMj4FwSKLUxxt), because file cards sent to the chat never arrived. Ask once how your human wants them.

> **Human: Keep their layout.** What goes where (which continent is spawn, where the Ashlands arm points, that the Mistlands are an eastern island chain) is the human's. Make the shapes natural inside that layout. Ask before moving a landmass.

> **Human: The design goal is a realistic-feeling world**, "official-feeling". Judge every change against that. Chapter 4 lists the tells that break it.

> **Human: Their saves are read-only.** Never delete, rename, copy over or "fix" a world in `worlds_local` unless they say so ("nuke Proxima" is the form it takes). Valheim's own save plumbing drops mod sidecars during rename and copy, so even a helpful copy can destroy a world.

> **Human: Mods live in Gale profiles, never in the Steam game folder.** Gale hardlinks plugin DLLs from a shared cache: copying over one writes through to every profile. Check the link count (`stat -c %h <file>`) before writing to any file in a profile; rename the old file out of the way instead of overwriting it. The staging script does this correctly; use it.

> **Human: Run every heavy command capped and niced**: `(ulimit -v 6000000; nice -n 10 <command>)`. A preview render that was handed pixel coordinates instead of fractions tried to build an image hundreds of times the map's size, exhausted memory and crashed the human's editor. The cap is the safety net; if a command dies at the cap, fix the input, do not raise the cap.

> **Human: Commit or push only when asked. Be terse**: a short status line, then act. Keep the handoff file (`HANDOFF.md`) current so the next session, human or agent, can pick up where you stopped.

> **Human: The human launches the game and builds the world.** You stage the files, tell them the exact sequence (chapter 6, step 9), and read the log afterwards.

The workspace

On Wubarrk's machine:

Path	What
`/home/rohan/WubarrkCODING/BetterContinents-TestMaps/`	the map workspace (not a git repository)
`generator/gen.py`	the ProximaMaxi generator: `python3 gen.py 2048 <out dir>`
`generator/mapscale.py`	the ONE place the Heightmap Amount lives (2.0); every tool reads it
`generator/check.py`, `relief.py`, `altbiomes.py`, `locations.py`	the checks (appendix A)
`generator/preview.py`, `altmap_preview.py`	the renderers
`generator/history/`	snapshots: `gen-NN-name.py`, `out-NN-2048/` with their `*.log`, copies of game logs
`previews/`	the numbered pictures the human saw
`fire-and-ice/`	the current map (version 13 of ProximaMaxi) and its README
`locations/`	the game's location list (`locations_1.0.15.json`) and its extractor
`altbiomes/`	the game's alt-biome list (`altbiomes_1.0.15_full.json`), its extractor, research notes
`stage-into-gale.sh`	writes the mod DLL and the config into a Gale profile
`HANDOFF.md`	the current state, kept for the next session
`/home/rohan/WubarrkCODING/BetterContinents10/`	the Better Continents fork (git; `ALTBIOMES.md`, `MIGRATE-1.0.md`)
`/home/rohan/WubarrkCODING/BetterContinents-Guide/`	the guide's Markdown and `build_pdf.py`

Path	What
`/home/rohan/WubarrkCODING/libs-Tools/`	the Valheim 1.0 reference: decompiled game code, `*-FACTS.md` notes
`~/.local/share/com.kesomannen.gale/valheim/profiles/BC-Tester/`	the Gale test profile: `BepInEx/config/BetterContinents.cfg`, `BepInEx/LogOutput.log`
`~/.config/unity3d/IronGate/Valheim/worlds_local/`	the worlds; each is a folder with a `BetterContinents` settings file inside
`~/upy/bin/python`	a Python with Pillow (for `altmap_preview.py`); everything else is the standard library

Read the workspace's own documents before the decompiled game code: `HANDOFF.md`, `fire-and-ice/README.md`, the fork's `ALTBIOMES.md`, `altbiomes/RESEARCH-*.md` and the `libs-Tools/*-FACTS.md` files. They cite the game code by file and line, so a claim can be checked in a minute instead of re-derived in an hour.

2 How images become a world

This chapter is the model you must hold in your head. Every number in it is derived from the mod's and the game's code, not guessed, and the derivations are given so that you can check them when versions change.

The square, the pixel and north

Every Better Continents map covers the same square: from -10,500 m to +10,500 m on each axis, 21 km by 21 km, whatever the world size settings say. The image centre is the world centre (0, 0). The top of the image is north (+z), the left edge is west (-x). Better Continents flips the image vertically on load so that this holds; author what you see.

Image size	Metres per pixel
2048 x 2048	10.25
4096 x 4096	5.13

ProximaMaxi is 2048. The game's own biome grid (chapter 3) is 12 m, so 2048 is already finer than anything the game samples for biomes and locations; 4096 only buys terrain detail, at four times the generation time.

The playable world ends at a radius of 10,000 m and the game's edge falloff drowns everything past about 10,500 m. Land within about 700 m of the 10,000 m edge is possible but the edge water bites at it; ProximaMaxi's rim chains stop at 0.90 to 0.96 of the half-width.

The files

The mod loads every correctly named file from the config's `Directory` (the "project directory"):

File	Format	What it does
`heightmap.png`	16-bit greyscale	terrain height; the whole vertical range of the world
`biomemap.png` + `biomemap.txt`	8-bit RGB + legend	the biome of every pixel; colours from the legend
`heatmap.png`	16-bit greyscale	Ashlands heat; ALSO decides what is Ashlands (heat > 0)
`forestmap.png`	16-bit greyscale	tree density; higher is more forest (with the config in appendix B)
`lavamap.png`	16-bit greyscale	lava in the Ashlands; the game's Ashlands vegetation mask
`locationmap.png` + `locationmap.txt`	8-bit RGB + legend	pinned sites for the start temple, bosses and vendors
`altbiomemap.png` + `altbiomemap.txt`	8-bit RGB + legend	planted alt biomes (Better Continents 0.8.1 and later)

Use 16-bit PNG for the greyscale maps: at the ProximaMaxi height scale one 8-bit step would be 1.6 m, enough to terrace every beach. The generator's `png.py` writes 16-bit greyscale and 8-bit RGB PNGs with nothing but the standard library.

Other map types exist (paint, terrain, moss, spawn, vegetation, roughness); ProximaMaxi does not use them and this document does not teach them. The guide does.

The height maths

This is the single most important derivation in the document. Getting it wrong drowned a whole world once (every lowland biome 30 m under water) and mis-scaled another (25 m under water everywhere).

From the mod's code (`WorldGeneratorPatch.GetBaseHeightV3`, `GetBiomeHeightWithHeight`) and the game's (`ZoneSystem.m_waterLevel = 30f`):

```
finalHeight        = pixelValue * HeightmapAmount - 0.15 + SeaLevelAdjustment
SeaLevelAdjustment = Lerp(1, -1, SeaLevel)          = 0 at the default Sea Level of 0.5
world height (m)   = finalHeight * 200               (no biome curve with a heightmap)
water surface      = 30 m absolute                    = finalHeight 0.15
```

Therefore, with `A` the Heightmap Amount:

```
waterline pixel      = 0.30 / A
metres above water   = (pixel * A - 0.30) * 200
ceiling above water  = (A - 0.30) * 200             (pixel 1.0)
floor below water    = -60 m                        (pixel 0.0, at any A)
```

Amount A	Waterline pixel	Ceiling above water
1.0 (the mod's default)	0.300	140 m
1.6	0.1875	260 m
2.0 (ProximaMaxi)	**0.150**	**340 m**

> **Rule:** The maps are authored for Amount 2.0 because the game's mountain locations need more than the 140 m a 1.0 map can reach: Hildir's Howling Cavern asks for 200 m, the mountain tower ruins 150 m, Moder's altar 150 m, the frost caves and mountain runestones 100 m, and the Deep North boss room 80 m of level ground. On a 1.0 map the cavern placed 0 of 3 in a real build, and Hildir's mountain quest could not be done.

> **Rule:** The Amount lives in exactly one place, `generator/mapscale.py` (`AMOUNT = 2.0`), and the generator, `check.py`, `preview.py`, `altbiomes.py`, `locations.py`, `relief.py` and `stage-into-gale.sh` all read it from there. Never hardcode it anywhere else. `mapscale.metres(v)` and `mapscale.value(m)` convert both ways.

At Amount 2.0 the pixel values mean:

Pixel value	Metres above water	Meaning
0.000	-60	deepest seabed
0.146	-1.4	wadeable shallows
0.150	**0**	**waterline**
0.215	+26	where vanilla starts to call ground mountain
0.650	+200	Hildir's cavern wants at least this
0.785	+254	the highest peak on ProximaMaxi
1.000	+340	the ceiling

The game's location rules speak of "altitude": that is height above the water, i.e. absolute height minus 30. A location with `minAltitude 200` wants pixel 0.65 at Amount 2.0.

> **Trap:** The Amount is copied into a world from the config **as it stood when the game was launched**. Stage a new config into a running game and that session keeps the old values while the files on disk say otherwise; the next world it creates is built from the old Amount against the new image. Always: stage, quit Valheim, relaunch, then create. The staging script refuses to run while Valheim runs, for

> this reason.

Two more things the mod does to heights with a heightmap loaded:

- Vanilla's rivers are bypassed (`Heightmap Override All = true`, the default), so **rivers and lakes must be carved into the heightmap** or the world has none.
- Unless `Mountains Allowed At Center` is on, the mod caps ground within about 600 m of the world centre at 46 m above the water, blending back to the heightmap by 1,000 m. Keep high ground away from the centre, or turn the setting on.

The biome map

The legend `biomemap.txt` (written with the defaults if it is missing) has one line per biome, `Name: RRGGBB`. Only these ten names exist in Valheim 1.0; any other name makes the mod reject the whole legend:

```
None: 000000        Meadows: 00FF00     BlackForest: 007F00   Swamp: 7F7F00      Mountain:
  FFFFFF
Plains: FFFF00      Mistlands: 7F7F7F   AshLands: FF0000      DeepNorth: 00FFFF  Ocean: 0000FF
```

Facts that matter when you paint it:

- **Pixels snap to the nearest legend colour**, so keep every edge hard. A blended pixel between Meadows and Ocean (0, 128, 128) snaps to Mistlands, and a soft coastline grows a fringe of Mistlands. The generator writes exact colours.
- **None** (black) means "let vanilla decide". Never leave it on a map you control: vanilla's biome there will not match your heights, and a None region counts as valid for every alt biome.
- **A biome you leave out is skipped**, and every location that needs it is reported as failed. Since 0.8.0 that no longer hangs world creation, but a map without Mistlands has no Queen.
- **Where water gets a land biome.** Vanilla keeps the land's biome in the water down to 26 m depth and calls it Ocean only deeper. Some coastal locations depend on this: the frozen ships lie in *Deep North* water, the dvergr harbours stand in *Mistlands* water. On a map, water is whatever you paint it, and painting Ocean right up to the waterline leaves those locations nowhere to go. But the alt-biome rules (chapter 3) punish a region whose edge is under water. ProximaMaxi labels Deep North water to 16 m depth and Mistlands water to 1 m, never against another biome's shore, and nothing else.
- The **Deep North** in Valheim 1.0 is a real biome with its own locations; vanilla decides its weather, snow and calm sea by a fixed geographic test, and the mod makes those follow your Deep North pixels instead.
- The mod votes the biome at a world point from the four surrounding pixels (bilinear weights, the heaviest biome wins). The checkers reproduce that vote.

The heat map

The mod answers the game's "is this Ashlands?" question with `heat > 0`, literally.

> **Trap:** `heatmap.png` must be **exactly zero** everywhere that is not Ashlands. One stray non-zero pixel turns that ground into Ashlands, silently. The generator asserts this before writing, and `check.py` counts `heat > 0 on land that is not AshLands` and requires 0.

Heat should still extend into the sea around the Ashlands as a ramp rising toward the coast: that is vanilla's own ocean gradient and it is what damages ships on the approach. So bound the heat field geometrically (an ellipse around the Ashlands landmass, faded), not by tracing the biome edge. The

config's `Heatmap Scale` (default 10) multiplies the pixel; most heat effects cap at 1, so a pixel value of 0.1 is already the full effect, and ProximaMaxi's maximum pixel is 0.16.

The lava map

The game decides Ashlands lava from its Ashlands vegetation mask; the mod loads `lavamap.png` (16-bit) as that mask. Lava appears where the value is at least 0.6. Locations read the same mask through their vegetation window: `LeviathanLava` wants 0.6 to 1.0 (it stands in lava), the charred fortress and the Ashlands boss location want at most 0.1, the Morgen holes at most 0.3, the fortress ruins at most 0.8. So a map with no lava map has no lava leviathans, and a map with lava everywhere has no fortresses. ProximaMaxi paints pools (about a tenth of the Ashlands) and streams down the tributary channels, kept off the beaches; the ground is let down into the pools so that they read as basins.

The forest map

Vanilla's forest factor is a noise value where **lower means more forest** (`InForest` is `factor < 1.15`). The mod normalises it through `InverseLerp(1.850145, 0.145071, factor)` into a 0-to-1 "forestness" (0 bare, 1 dense) and combines it with the map value `fmap` as `Lerp(n, n * fmap, Multiply) + fmap * Add`.

> **Rule:** Set `Forestmap Multiply = 0` and `Forestmap Add = 1`, so that the result is `n + fmap` and a **higher map value means more trees**, predictably. At the defaults (1 and 1) it is `fmap * (n + 1)`, which saturates almost everywhere and gives no control. Leave `Forest Amount` at 0.5 (neutral: its offset is `Lerp(1, -1, value)`) and set `Forest Factor Overrides All Trees = true`, or the map barely touches the meadow birches and oaks.

Do not hold tree density down to protect the minimap. Vanilla paints the forest overlay only in Meadows, Plains and Black Forest; Swamp, Mountain and Deep North draw none and Mistlands has its own mist channel. A minimap that reads as forest everywhere was a bug in the mod's minimap patch (fixed in 0.8.0), not honest density.

The location map

`locationmap.png` pins specific sites: black is "nothing here", and any pixel of a legend colour asks for that location at that spot. The mod picks the position among adjacent pixels of the same colour, so paint a 3 x 3 block, not a lone pixel that one awkward sample could lose. The legend `locationmap.txt` is `LocationName: r,g,b` with the game's prefab names; ProximaMaxi pins nine:

```
StartTemple: 255,0,0      Eikthyrnir: 255,153,0     GDKing: 0,255,0
  Vendor_BlackForest: 0,0,255
Hildir_camp: 128,0,255    Bonemass: 0,255,255       Dragonqueen: 74,134,232  GoblinKing:
  255,255,0
Mistlands_DvergrBossEntrance1: 255,0,255
```

The mod registers these sites before the game places anything, then the game's own placement runs for everything else (unless `Skip Default Locations` is on, which you do not want: it skips every default location except the start temple). Pinning a boss does not exempt it from the game's rules: a site must still be the right biome at the right height, or the game refuses it and tries at random. The generator therefore searches for a suitable pixel near each intended site (right biome, off the river banks) rather than painting the colour at a fixed coordinate.

Everything you paint here is a matter of the map's story; everything you do not paint is decided by the algorithm in chapter 3.

The alt-biome map (0.8.1)

`altbiomemap.png` plants the game's alt biomes on regions of your choice; black is unplanted (the game's random placement decides there). The legend `altbiomemap.txt` has one line per colour:

```
<alt biome>[ + <alt biome> ...]: <colour>       plant these where the image has this colour
<alt biome>[ + <alt biome> ...]: at <x>, <z>    plant them on the whole region under world
  position x, z
none: <colour>                                  a protected region: no alt biome, planted or
  random
```

Names are the game's internal names (appendix C lists all 32, with the mod's default colour for each). `*Mistlands` is a wildcard, `!Name` forces past the content rules, `+` stacks. Colours are `RRGGBB`, `#RRGGBB` or `r,g,b`; pixels within 20 RGB units of a legend colour count as it, and a colour within 40 units of another is warned about. Black is refused as a legend colour. At most 254 entries.

How planting behaves (the fork's `ALTBIOMES.md` is the authority):

- A colour **claims only the base biomes its alt biomes belong to**. Dark Meadows painted across a meadow and its beach takes the meadow and leaves the beach alone. So painting past a biome edge is harmless, except for a multi-biome alt (Lantern takes seven biomes, Mushroom two): its colour claims all of them, so it must not spill onto neighbouring land you did not mean.
- Every 4-connected patch of one colour becomes **a region of its own**, carved out of the game's region; what is left of the game's region stays one region. Nothing is merged. On Ashlands, Deep North and Ocean, all patches of one colour form ONE region (the game treats those biomes as one region each), counted once.
- Planted regions **ignore the game's occurrence rules** (size, distance, height, position, neighbour, chance) and keep its content rules (disabled alt biome, wrong base biome, incompatible pair) unless forced with `!`.
- Planted regions **count toward each alt biome's minimum and maximum** when the game then places the rest at random on unplanted regions (the quota rule). Plant Mushroom three times and the game plants no more (its maximum is 3); plant Fortress Mountain once and the game may add one.
- The grid is 12 m: a patch narrower than a cell or two vanishes or breaks up. Paint whole features.
- A world carries its own baked copy; editing the PNG or the legend changes nothing for an existing world.

The game log reports the outcome at every world load: `Alt biomes: planted: N region(s), N alt biome(s) applied, 0 refused`, one line per colour, and one line per alt biome with the reason it did or did not place (`+3 planted (counted toward the maximum)`).

Baking, the config and the world

> **Rule: Maps are baked into the world at creation.** The mod stores every map, with the effect of its legend, in the world's `BetterContinents` settings file (`worlds_local/<name>/BetterContinents`, an extension-less file inside the world folder). Editing a PNG afterwards does nothing to that world. Every map change needs a brand new world.

> **Rule: The config is read once, at game launch.** `BepInEx/config/BetterContinents.cfg` in the profile is read when Valheim starts, and its values are baked into a world when the world is created. The sequence is always: stage, quit the game, relaunch, then create the world.

> **Rule: The preset must be `From Config`.** At New World, the dropdown beside the seed must say `From Config` (the config's `SelectedPreset = From Config` pre-selects it). Anything else and the mod turns itself off for that world and generates vanilla, without a warning.

The server and other players need only the mod (Better Continents 0.8.1 or later), not the images: the world carries them. A dedicated server cannot create a Better Continents world from a config; worlds are created on a client and then copied.

3 What the game demands of a map

The game does not care how a map looks. It samples the world on a grid, cuts it into regions, and runs two placement algorithms over it: one for alt biomes, one for locations. Both are replayed by the workspace's checkers (`altbiomes.py`, `locations.py`), which were validated line for line against real builds. This chapter is what those algorithms want, so that you can design for them instead of discovering their refusals in the game log.

The game's grid

At every world load, `AltBiomeWorldData.GenerateBiomePoints` samples the biome and the height on a 2048 x 2048 grid of 12 m cells centred on the origin: point (i, j) is at `(i - 1024) * 12 + 6` metres on each axis. Points further than 10,500 m from the centre are marked Ocean at height -1000. This grid is the source of truth for the base biome of every terrain chunk's corners, for alt biomes, and for location candidates. It is regenerated on every load, on the server and on every client (the game never reads its cache), and nothing about it is sent over the network: clients and server agree only because they hold the same maps, which the world file guarantees.

Regions and alt biomes

`GenerateSectors` flood-fills 4-connected grid points of one biome into **regions** (the game calls them sectors). Ashlands, Deep North and Ocean are each ONE world-wide region, however many patches they have. For every region the game records:

- `EdgeCount`: the number of border cells;
- `DistanceFromCenter`: the distance of the mean border position from the origin (a region wrapped round the origin measures near 0);
- `HeightAvg`: `(lowest + highest) / 2` of the border points' absolute heights.

Then `GenerateAltBiomes` goes through each base biome and each of its alt biomes, shuffles the biome's regions with a seed from the world seed, and gives the alt biome to regions in that order while its count is below its maximum (ignoring the chance roll while below the minimum). Every region must pass `BiomeSector.CanAddModifier`:

- distance from the centre at least the alt biome's `minDistanceFromCenter` (500 to 2,000 m);
- `EdgeCount` inside the alt biome's window, e.g. [66, 199) for Mushroom, [199, 299) for the three Mountain alts (which are mutually exclusive), [99, 599) for Rockless Mistlands, [20, 80) for Kalhygge Black Forest;
- `HeightAvg` at least 30 absolute, i.e. the border's (min + max) / 2 at or above the waterline;
- no incompatible alt biome already on the region;
- `m_requireNeighbor` is broken in vanilla: any non-zero value means never placed (no vanilla alt biome sets it).

What this means for a map:

> **Rule: Big regions get nothing; islands and mid-sized patches are the hosts.** A continent's forest has an edge count in the thousands and qualifies for no alt biome. For a roundish region the edge count is about half its radius in metres, so a window of [66, 199) means regions about 130 to 400 m in radius, [199, 299) about 400 to 600 m. A map that wants the game to place alt biomes at random needs islands and patches of those sizes, in every biome, beyond the distance minimum. A map that plants them (chapter 2) can put them anywhere.

> **Rule: A host must have a real cliff somewhere on its shore.** An all-beach island's border sits at the waterline, HeightAvg 30 exactly, a coin flip. Give host islands some high ground on the coast.

> **Trap: Water labelled with a land biome pulls the region's edge under water.** With Deep North water labelled to 16 m, the Deep North region's border lies on the seabed and its HeightAvg fails; on a global biome that costs nothing (Deep North hosts only Lantern anyway), but on a Meadows island it takes the host away. The first attempt at labelling every coast to 26 m took the hosts from Birch Meadows and the Mistlands alts and made 123 planted regions out of 21. Label only the biomes whose locations need water, only as deep as they need (Deep North 16 m, Mistlands 1 m), never against a foreign shore, and pull a multi-biome alt's painted water 3 px back from foreign shores.

Other facts you will need:

- In the game's data **Lantern is the only alt biome that takes Deep North** (it takes seven biomes), and **no alt biome takes Ashlands**. A "Deep North alt island" can only be Lantern; an Ashlands alt would need a modded alt biome (Expand World Data), untested with 0.8.1 planting.
- Alt biomes do not change the displayed biome name; they change spawns, vegetation, locations, weather and music, and some add locations of their own (Ruin Black Forest adds 4, Goblin Plains 4, Hut Swamp 3, Death Plains 3, Dark Meadows 1, Fortress Mountain 1). Those add-on locations are placed only inside that alt biome's regions and appear in the log with suffixes such as `_1` and `_sunk`.
- Planting (chapter 2) overrides all of this for the regions you paint, and counts toward the maxima. ProximaMaxi plants 19 of the 32 (23 regions) on natural features and leaves the rest to chance; `altbiomes.py` showed all 32 reaching their minimum in 100% of 200 seeds, and the game agreed.
- Dev console (cheats): `biomeinfo`, `altbioms` (sic), `findbiome <text>`, `findbiometp <text>`, `genloc alt` (re-rolls placement, server). The mod's `bc ab info` reports every legend line's result.

Locations

`ZoneSystem.GenerateLocations` runs once, when the world is created, over every location in the game's list: the `_ZoneSystem` prefab's own 130 plus the `_LocationList_*` add-ons for the Ashlands, Mistlands, Deep North, Hildir, the mountain caves and the 1.0 content. Prioritized locations (bosses, vendors, the progression dungeons) go first with 60,000 attempts each; the rest get 12,000. For each attempt:

1. A candidate **zone** (64 m square) is taken from a random point of the grid above, of one of the location's biomes (`AllPointsAboveSeaLevel` unless the location allows negative altitude), or from around the centre for a centre-first location. A zone that already holds a location is skipped.
2. The zone centre must have the location's **biome area**: `Median` means the same biome at the eight points 64 m around it; `Edge` means not; `Everything` means either.
3. **Six random points** in the zone are tried in turn. Each must pass: distance from the world centre (`minDistanceFromCenter`, `maxDistanceFromCenter`), biome (the mod's four-pixel vote), **altitude** (bilinear height less the 30 m water level, within `minAltitude..maxAltitude`), forest (the forest factor within `forestTresholdMin..Max` if `inForest`), **terrain delta** (highest minus lowest of ten random heights within the exterior radius, within `minTerrainDelta..max`), distance from similar locations (`minDistanceFromSimilar`, and the same for the location's group), and vegetation (the Ashlands lava mask, `minimumVegetation..maximum`).
4. A **unique** location (Haldor, Hildir's camp, the Bog Witch, the big rock clearing, the ancient upgrade station) gets up to its quantity of candidate sites like any other; when a player first reaches one it is spawned and the other candidates are removed. One legal site is therefore enough, and the checker places one and stops.

The mod registers the location map's sites first, so a pinned boss occupies its zone before the random pass. One location per zone means a 64 m grid caps density: a biome of 14 km^2 has at most 3,400 zones, and the goblin towers alone want 100 sites 512 m apart.

When a location cannot fill its quantity the game logs `Failed to place all <name>, placed N out of M`. The checker prints the same count with the reason counts by category: `zoneTaken`, `area`, `biome`, `altitude`, `terrainDelta`, `similar`, `notSimilar`, `forest`, `vegetation`.

What the important locations want

Numbers from the game data (`locations/locations_1.0.15.json`); altitude is metres above the water, delta is the terrain delta window, distances in metres. Appendix C has the full table of the 170 enabled location types.

Location	Biome, area	Qty	Altitude	Delta	Other
StartTemple	Meadows, Median	1	3..1000	0..3	prioritized; forest factor 1..5
Eikthyrnir	Meadows, Median	3	1..1000	0..3	prioritized
GDKing (the Elder)	BlackForest, Median	4	1..1000	0..5	3,000 m from similar
Bonemass	Swamp, Median	5	**0..2**	0..4	3,000 m from similar: flat bog at the water
Dragonqueen (Moder)	Mountain, Median	3	**150..500**	0..4	3,000 m from similar
GoblinKing (Yagluth)	Plains, Median	4	1..1000	0..4	3,000 m from similar
Mistlands_DvergrBossEntrance1 (the Queen)	Mistlands, Median	5	1..20	0..40	group DvergrBoss, 2,048 m
FaderLocation (the Ashlands boss)	AshLands, Median	3	1..300	0..40	lava mask at most 0.1
DN_Bossroom (the Deep North boss)	DeepNorth, Everything	3	**80..5000**	**0..6**	1,024 m from similar: 80 m up AND level over 32 m
Vendor_BlackForest (Haldor)	BlackForest, Median	10, unique	1..1000	0..2	512 m from similar
Hildir_camp	Meadows, Everything	10, unique	1..1000	0..2	1,000 m from similar
Hildir_crypt	BlackForest, Median	3	1..1000	0..2	3,000 m
Hildir_cave (the Howling Cavern)	Mountain, Median	3	**200..5000**	0..40	2,000 m
Hildir_plainsfortress	Plains, Median	3	8..1000	0..4	3,000 m
BogWitch_Camp	Swamp, Median	10, unique	1..1000	0..2	1,000 m
StoneTowerRuins04 / 05 (mountain ruins)	Mountain, Everything	50 each	**150..1000**	6..40	group Mountainruin, 128 m
MountainCave02 (frost caves)	Mountain, Everything	120	**100..5000**	0..40	group mountaincaves, 200 m
Runestone_Mountains	Mountain, Everything	100	100..1000	0..3	128 m
FrozenShip01..03_DN	DeepNorth, Everything	50 each	**-15..-5**	0..100	**8,000..9,750 m from the centre**; snaps to water
ShipWreck01_DN	DeepNorth, Everything	170	-0.5..0.5	0..2	the shoreline itself

Location	Biome, area	Qty	Altitude	Delta	Other
Mistlands_Harbour1	Mistlands, **Edge**	100	**-1..-0.25**	2..20	snaps to water
Mistlands_Viaduct1	Mistlands, Median	100	-1..15	2..40	group Harbour, 128 m
Mistlands_DvergrTownEntrance1 (infested mines)	Mistlands, Median	120	12..1000	5..40	prioritized
Mistlands_DvergrTownEntrance2	Mistlands, Median	120	1..20	0..40	prioritized
Mistlands_Giant1 / Giant2	Mistlands, Median	250 / 85	-1..1000	0..3 / 0..4	group Giant, 180 / 256 m
SunkenCrypt4	Swamp, Median	175	**0..2**	0..4	prioritized; 64 m
Crypt2 / 3 / 4 (burial chambers)	BlackForest, Everything	200 each	1..1000	0..2	128 m
TrollCave02	BlackForest, Median	200	3..1000	**5..10**	256 m: wants a slope
GoblinCamp2	Plains, Everything	200	2..1000	0..3	250 m
StoneTower1 / StoneTower3 (goblin towers)	Plains, Everything	50 each	1..1000	0..3	group Goblintower, **512 m**
StoneHenge1..5	Plains, Everything	5 each	5..1000	0..3	group Stonehenge, **1,000 m**
TarPit1	Plains, Median	100	5..60	0..2	128 m
LeviathanLava	AshLands, Everything	100	10..1000	0..10	**lava mask 0.6..1.0**
CharredFortress	AshLands, Everything	20	20..1000	0..10	prioritized; lava mask at most 0.1
NorthVillage	DeepNorth, Median	135	2..1000	0..3	70 to 85 m from similar (huts cluster)
MorkBorg	DeepNorth, Everything	40	30..10000	0..300	275 m

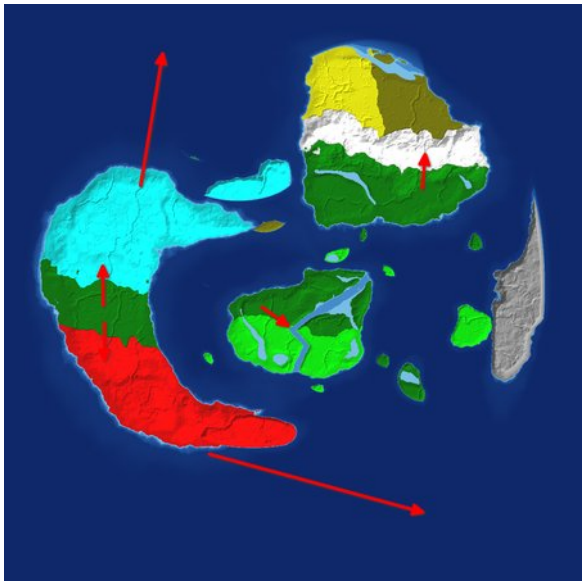
Read the table as design requirements:

- **Mountains** must reach 200 m above the water somewhere (the cavern) and 150 m across a wide area (100 tower ruins 128 m apart, Moder's altar). Hence Amount 2.0 and peaks of 250 m.
- **Deep North** must have flat ground 80 m up (the boss room: three sites 1 km apart, level within 6 m over 32 m) and its water must be Deep North to 15 m depth in the 8 to 9.75 km ring (the frozen ships). ProximaMaxi's icefields are flat-topped plateaus at 105 to 125 m.
- **Swamp** must be nearly flat and nearly at the waterline: Bonemass and the sunken crypts want 0 to 2 m. ProximaMaxi's bog is 1.5 to 5.5 m (swamp mean 5.8 m); Bonemass places 4 of 5, and one is enough (the five want 3 km between them).
- **Mistlands** must have lowland shore under 20 m (harbours, viaducts, lighthouses, mine entrances) as well as its high ground; ProximaMaxi's lowland valleys meet the sea as shore, sunk 10 m.
- **Plains** are quantity-limited by area: 100 goblin towers 512 m apart need about 26 km^2 of plains and ProximaMaxi has 14 km^2. Shortfalls in the Goblintower, Stonehenge and Giant groups are area, not bugs.
- **Ashlands** need lava for the leviathans and lava-free ground for the fortresses.
- **Black Forest** needs slopes (troll caves want a 5 to 10 m delta) and flat forest (crypts, ruins) both.

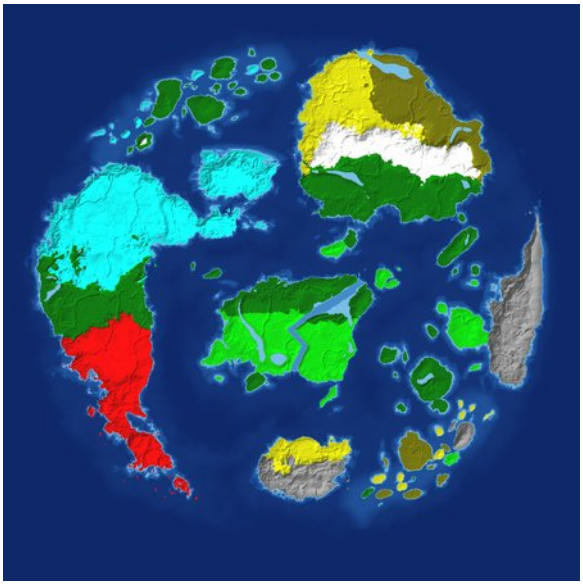
Rule: Layout is radial. Vanilla places bosses and dungeons by distance from the world centre (Meadows at the centre, Mistlands, Ashlands and Deep North at the rim), and several locations carry explicit distance windows (the frozen ships 8,000 to 9,750 m; the big rock clearing at least 1,000 m; the ancient upgrade station at least 500 m). A map that puts the Meadows at radius 6,000+ has nowhere legal for meadow structures. ProximaMaxi: spawn 2,203 m out, the Elder 916 m, Haldor 4,284 m, Moder 6,015 m, Yagluth 6,574 m, Hildir 6,629 m, Bonemass 7,871 m, the Queen 7,976 m.

4 What "reads natural" means

The human's standing goal: "we want a realistic feeling map", "official-feeling". This chapter is what that meant in practice on ProximaMaxi: the tells that broke the illusion, each found in a picture the human looked at, and how each was fixed. The pictures are the evidence.



`01-parked-before.png`



`06-archipelagos.png`

Left: version 01, the layout as first drawn: elliptical landmasses, biome bands as stripes, round snow discs, islands parked in a row (a warp bug had displaced them all by the same vector). Right: version 06, the first the human liked ("i like it 06"): the same layout with fractal coasts, ragged fronts, a wandering crest and two archipelagos. Everything after 06 was refinement.

The tells, and the fixes

1. **Biome borders as ruled lines or latitude stripes.** A front drawn with ordinary fractal noise (gain 0.5) is a smooth wave at every scale. Use rough noise (gain 0.62, so that the fine octaves keep real power), give it a second-order domain warp so that the lobes hook into tongues instead of rippling, and let fronts follow terrain: Deep North climbs ridges, forest sits in valleys. ProximaMaxi's `front_offset()` combines a 4.5 km drift, 1.5 km hooked lobes and 70 m raggedness.
2. **Heights that step at biome borders.** Blend each biome's terrain across the front rather than switching at it.
3. **A mountain band of constant width, and round white snow discs.** Use a wandering crest (a polyline spine with spurs and side valleys off it), radial ridges on isolated peaks, and label snow from the FINISHED height so that it climbs valleys and thins on spurs instead of drawing a circle.
4. **Smooth oval coastlines and uniform shelves.** Coast noise must have power at 200 m to 1 km: three scales (a gulf, bays and headlands, and a fine roughness so that the waterline is never smooth), plus a domain warp. Every landmass field must be on ONE shared scale so that islands get the same crinkle as continents (measured per landmass, a 370 m island got a tenth of a continent's detail).
5. **Evenly scattered round island dots.** Rotate and elongate islands, give them satellites, put chains on a shared shelf, throw skerries off cliffed coasts.
6. **Every coast a gentle beach.** Windward coasts and coasts under high ground are cliffs.
7. **Straight rivers, or no rivers.** With a heightmap the game draws no rivers, so carve them, and carve them where water would actually drain: ProximaMaxi computes the drainage network on the terrain (Priority-Flood on a 20 m grid, accumulation, streams from 0.2 km^2, full rivers at 15 km^2) and cuts

every stream downhill into a bigger river and out to sea, with lakes where rivers cross low ground and in natural basins. The starter river beside spawn is a hand-drawn Catmull-Rom meander because the human said the first one "looks too sharp and unnatural".

8. **Ground that reads flat.** Kilometre-scale ridges plus footstep hummocks still read as flat. The hills people notice are at 100 to 300 m wavelength, about 4 m RMS in meadows. Fade relief in within about 100 m of the shore, not over a 500 m beach ramp, or every coast is a tilted plain.



12i-start-coast.png



13i-start-coast.png

The spawn coast in version 12 (left) and version 13 (right), the same crop. The human said of 12 "the world reads too flat"; 13 adds rolling relief at 220 m and below and a 210 m beach with the relief starting 125 m inland.

Relief that reads, in numbers

`relief.py` prints, per biome, the RMS height change over 50 m (the hills you walk over) and over 250 m (the hills you see), and the mean height in 50 m bands out from the water. The numbers behind "it reads flat":

	Meadows, 50 m	Meadows, 250 m	Coast, first 50 m	50 to 100 m	100 to 150 m
Version 12 (too flat)	2.9 m	7.4 m	3.4 m	6.8 m	10.4 m
Version 13 (approved)	4.0 m	9.1 m	5.5 m	11.8 m	15.6 m

Version 13's `ROLL` (RMS metres per biome): Deep North 3, Ashlands 3.5, Mountain 4, Black Forest 4.5, Plains 2.5, Meadows 3.5, Mistlands 3 inside its own relief, the bog 0.5. Re-measure after touching `ROLL`, the beach ramp or any biome's relief. If the human still finds it flat, raise Meadows to 4.5 and Black Forest to 5.5 and re-run `locations.py`; if the plains get too rough for the goblin towers (delta 0 to 3), lower Plains to 2.0.

> **Trap:** A value-noise FBM's output is compressed round 0.5 (its standard deviation is about 0.13, not 0.29). `(noise - 0.5) * 2 * amplitude` is therefore worth about a quarter of the amplitude you asked for. The first relief pass "looked like nothing" for this reason. Scale by 1 / std (`ROLL_K = 7.68`) before treating amplitude as metres, and measure the result with `relief.py` rather than trusting the constant.

The reference world

Wubarrk considers **Era02**, a real pre-1.0 Better Continents world, correct: Amount 1.0, 33.8% of the image land, land averaging 33.6 m. Its high plateau and abrupt continental drop-off are artefacts of how it was drawn, not targets. ProximaMaxi is 27.7% land, with the per-biome means in chapter 8.

Where the human's taste showed

Things the human asked for, verbatim, and what they became: "the river looks too sharp and unnatural" (the meander redraw); "make various alt deep north islands, move wolf mountain" (the Deep North isles of the cold archipelago, Lantern on all of them, Wolf Mountain on the west massif's twin peaks with random Wolf Mountain switched off); "we def want some mushroom meadows" (Mushroom planted three times, the game's maximum: the whole meadow east of the starter river, bounded by river, forest and coast instead of a circle, plus two islands); "the Ashlands is too small, add island chains along the edges" (the arm fattened, burning isles off its tip, four rim chains); "the shoreline is a little too flat, the world reads too flat" (version 13). Designed for a dark realm: the dangerous alt variants got the landmarks, the quiet ones were left to chance, and Hildir's island stayed plain.

5 The generator, as a pattern

`generator/gen.py` is 2,300 lines and specific to ProximaMaxi, but it is built from a small set of parts that any map generator for this game needs. Learn the parts; then change the map by changing the constants and selectors, or write your own generator on the same skeleton. Run it with `python3 gen.py 2048 <out dir>` (about 40 s on all cores at 2048; use 512 for a quick look). It needs only the standard library.

Skeleton

```
value noise  ->  FBM(octaves, base frequency, seed, gain): tileable, wrapping lattices
domain warp  ->  (u, v) -> (wu, wv); moderate, leashed per landmass, anchored per island
landmass fields  ->  one shared scale: crescent, spawn continent, second continent, Mistlands
                     chain, southern landmass, islands, satellites, archipelagos, skerries
coast noise  ->  three scales added to every field; the leash contour carries it too
biome fronts ->  front_offset(): drift + hooked lobes + rough edge; snow from the finished
  height
heights      ->  per-biome relief blended across fronts, beach and inland ramps, cliffs,
                 massifs, a spine with spurs, Deep North icefields, Mistlands valleys, ROLL
per pixel    ->  evaluate(u, v) -> (height value, biome, heat, forest, location region, alt
  plant)
parallel     ->  multiprocessing over row bands -> arrays
post passes  ->  carve_rivers() (drainage), label_shallows(), write_altbiomemap(), location
  sites
write        ->  png.py: 16-bit greyscale and 8-bit RGB, plus the legends
```

The parts

Value noise and FBM. `FBM(octaves, base, seed, gain=0.5)` sums bilinear samples of random lattices whose resolution doubles per octave. The lattices wrap in both axes, so callers can sample past [0, 1] (the warp pushes coordinates negative) and no octave introduces a seam. Named instances carry their purpose: `coast`, `coast2`, `detail`, `ridge`, `riv1..3`, `lakes`, `front1`, `front2`, `frontn` (gain 0.62), `cliffn`, `spurn`, `skn`, `coastn`, `lavan`, `rolln`. One seed constant, `SEED = 20260922`, offsets them all, so a map is reproducible.

Domain warp, leashed and anchored. Warped coordinates make every outline non-elliptical, but a warp big enough to be interesting evaluates a continent's field far from where the continent is and strands its biomes across the map. Each landmass field is capped against its own UNWARPED ellipse plus slack (the "leash"), and the leash contour carries the coastal noise, because wherever it clips it IS the coastline. Small features (islands) subtract the warp at their own centre ("anchored"), which is what fixed the parked-islands bug of version 01: unanchored, every island in an archipelago was displaced by the same warp vector and landed in a row.

One scale for every landmass. Fields are measured in the crescent's units (about 4.2 km), not each in its own radii, so that coastal noise and beach ramps, added in those units, come out the same size on a 370 m island and a 4 km continent.

Biome fronts. `front_offset(u, v)` displaces a nominal front line; fronts follow terrain where the biome should (snow from the finished height, forest into valleys). Biome is decided from a handful of scalar fields per landmass (latitude on the crescent, distance along the spine, the side of a divide on the second continent), each offset by the front noise.

Heights. Every biome has a base relief, a `CLIFF` weight (how much of its coast is cliff) and a `ROLL` RMS, blended across the fronts. Ramps: `beach = sstep(0, 0.05 * (1 - 0.8 * cliff), land)`, about 210 m, is the platform the shore rises on; `inland = sstep(0, 0.03 ..., land)`, about 125 m, is where the relief and the hummocks fade in. Special terrain: `MASSIFS` (isolated peaks with radial ridges), `SPINE` (the range crest as a polyline; `spine_query()` gives the distance along and across it, spurs from `spurn`), `dn_relief()` (flat-topped icefields 105 to 125 m for the Deep North boss room), `mist_relief()`

(relief plus lowland valleys sunk 10 m so that the harbours have shore). The `H(metres)` helper converts metres to a pixel value through `mapscale`.

Islands. `ISLANDS` entries are (x, z, rx, rz, kind); `add_satellites()` adds smaller companions cleared of their parent; `add_archipelagos()` lays chains along polylines (`ARCHIPELAGOS`, with a size factor for the rim chains) and buckets them into a grid (`ISL_GRID`) so that `evaluate()` only tests nearby islands. Every island gets its own rotated, elongated, noise-shaped outline (`island_shape`).

evaluate(u, v, dbg=None). The whole world at one point: the heightmap value, the biome, the heat, the forest density, the location-search region and the alt plant. Pass a dict as `dbg` to get every intermediate value back:

```
cd generator && python3 -c "import gen; d = {}; print(gen.evaluate(0.495, 0.605, d)); print(d)"
```

probes one point without generating anything. This is how you debug "why is this pixel Plains".

Rivers by drainage. `carve_rivers(hm, bc)` averages the height onto a 20 m grid with half a metre of noise (so that the flood's tie-break wanders instead of running along the grid's eight angles), runs Priority-Flood from the sea inward to get a downhill direction for every cell and the fill level of every basin, accumulates drainage area, starts streams at 0.2 km^2, widens to full rivers at 15 km^2, and stamps the channels into the heightmap (`river_profile`). Basins whose fill level is above the terrain become lakes. Version 13: 65 mouths, 144 reaches, 43 km of water, 16 lakes. The return value is how far each pixel was cut, so that location sites keep off the banks.

Shallows with a biome. `label_shallows(hm, bc)` floods outward from Deep North and Mistlands land into the sea, giving water down to 16 m and 1 m respectively the land's biome, never within 1 px of another biome's shore, and never heated water.

Alt planting. `ALT_PLANTS` is a list of (alt biome, base biome, selector): `island k`, `massif k`, `landmass name`, `patch (landmass, cx, cz, r)` with a ragged edge, `eastbank` (the whole meadow east of the starter river), `mistseg k` (one island of the Mistlands chain, bounded by its straits), `isles` (every island of a kind in a box). `alt_plant()` evaluates them per pixel; `write_altbiomemap()` keeps each plant's largest connected piece (specks would become separate one-point regions), grows whole-feature plants 2 px onto other biomes and the sea (so that the game plants the region whole, with no fringe left behind), keeps patches 2 px inside their own biome (a spit or a forest tongue is a lone point on the game's grid, and painting it made a one-point planted region that used up a slot), pulls a multi-biome alt's painted water 3 px back from foreign shores, and writes the legend.

Location sites. `LOCATIONS` holds the nine pinned sites as (name, colour, region, x, z) in world fractions; after generation the nearest pixel of the wanted region that no river cut into (3 px margin) is chosen and painted as a 3 x 3 block, and `locationmap.txt` is written.

Heat and lava. Heat is keyed to the crescent's Ashlands arm and the burning isles as a geometric ramp, extended into the sea, and asserted zero elsewhere. Lava: `lavan` pools at 11 cycles, about a tenth of the Ashlands, the ground let down into them, plus streams down the tributary channels, kept off the beach.

Changing the map

- Move or resize a landmass: its tuple (`CR_OUT`, `CR_ARM_S`, `MIST`, `PM`, `ISLANDS[k]` ...). Ask the human first; that is their layout.
- More or less of a biome: the front constants. The Ashlands/Black Forest front went from 0.09 to 0.05, in three places, when the human said the Ashlands was too small; the plains/swamp divide (`g = wx - 0.34`) moved 840 m east for more plains.
- Higher peaks: the range uplift (`62 + 158 * arch^0.6`, peaks 254 m), the `MASSIFS` heights, and the snow line (58 m; raising the uplift without raising it doubles the Mountain biome).
- Relief: `ROLL`, `CLIFF`, the beach and inland ramps.

- Alt biomes: `ALT_PLANTS` and `ALT_COLOUR`.
- Everything else: the generator's own comments explain each constant's reason, often with the version it changed in. Keep that habit: a constant without its reason is the next session's trap.

After any change: snapshot, regenerate, measure, render, show. Chapter 6.

6 The workflow

One round, from a change to a proven world. Do the steps in order. Every step has a command and an acceptance criterion; a step whose criterion fails is not done. `$T` is the workspace, `/home/rohan/WubarrkCODING/BetterContinents-TestMaps`; `NN` is the version number you are making. Every heavy command is written with the cap; keep it.

This chapter starts from nothing, with `gen.py`. Chapter 9 is the other starting point: `bc_export` a real world (vanilla or Better Continents) instead, at the same Amount 2.0 / Sea Level 0.5 encoding, and cut pieces of it into `fire-and-ice/` (or the reverse) before Step 1 below.

```
cd /home/rohan/WubarrkCODING/BetterContinents-TestMaps/generator
```

Step 0: know the state

Read `$T/HANDOFF.md` (the current version, what the human last said, what is open), then `$T/fire-and-ice/README.md`. Check that the mod and the game are what you think:

```
grep -m1 "Loading \[Better Continents" ~/.local/share/com.kesomannen.gale/valheim/profiles/BC-T
  ester/BepInEx/LogOutput.log
head -20 /home/rohan/WubarrkCODING/libs-Tools/CURRENT-GAME-VERSION.md
python3 -c "import mapscale; print(mapscale.AMOUNT, mapscale.SEA, mapscale.CEILING)"
```

Criterion: Better Continents 0.8.1 or later (the alt-biome map needs it); Valheim 1.0 (the data files match the current build; a new game build means re-running the extractors, appendix A); Amount 2.0, waterline 0.15, ceiling 340.

Step 1: agree the change

State in one line what you will change and why, in the human's terms ("fatten the Ashlands arm and add burning isles off its tip"). If it moves a landmass or changes the layout, ask first. If it is a refinement of shape, relief, rivers, alt plants or locations, proceed.

Step 2: snapshot before you touch anything

```
diff -q history/gen-13-relief.py gen.py || cp gen.py history/gen-NN-<short-name>.py
```

The last snapshot is the previous version; `diff history/gen-12-locations-lava-rim.py gen.py` shows what the last session changed. Never lose a version the human has seen.

Step 3: change the generator

Edit `gen.py`. Write the reason for every new constant in a comment beside it, with the version. Probe single points with `evaluate(u, v, dbg)` while you work, and run at 512 for a quick look:

```
(ulimit -v 6000000; nice -n 10 python3 gen.py 512 /tmp/quick) && \
(ulimit -v 6000000; nice -n 10 python3 preview.py /tmp/quick /tmp/quick.png 1)
```

Step 4: generate at full size

```
mkdir -p history/out-NN-2048
(ulimit -v 6000000; nice -n 10 python3 gen.py 2048 history/out-NN-2048) 2>&1 | tee
  history/out-NN-2048/gen.log
```

Criterion: it ends with `done 2048`; no `!! no site found` line (every pinned location found a pixel); the rivers line and the `alt ...` lines look like version 13's (chapter 8). About 40 s.

Step 5: measure

Four checks, in this order, each saved beside the output:

```
(ulimit -v 6000000; nice -n 10 python3 check.py history/out-NN-2048)   | tee
  history/out-NN-2048/check.log
(ulimit -v 6000000; nice -n 10 python3 relief.py history/out-NN-2048)  | tee
  history/out-NN-2048/relief.log
(ulimit -v 6000000; nice -n 10 python3 altbiomes.py history/out-NN-2048
  ../previews/NNh-altbiome-regions.png 200) | tee history/out-NN-2048/altbiomes.log
(ulimit -v 6000000; nice -n 10 python3 locations.py history/out-NN-2048) | tee
  history/out-NN-2048/locations.log
```

Check: `check.py` (seconds). Land 25 to 35% of the image. Every lowland biome mostly dry: Meadows, Plains and Black Forest above 90% dry with means of 10 to 25 m; Swamp about 90% dry with a mean of 4 to 8 m; Mountain 100% dry with a maximum above 200 m; Deep North maximum above 100 m. The three invariants: `dry ground labelled Ocean: 0`, `heat > 0 on land that is not AshLands: 0`, `AshLands pixels with no heat: 0`. Every pinned location listed with its biome and a sane height. A lowland biome near 0% dry, or a mean within a metre or two of the waterline, is a drowned world: stop.

Check: `relief.py` (about a minute). Meadows about 4 m over 50 m and 9 m over 250 m; the coast profile climbing about 10 m in the first 100 m. Compare with the previous version's log, not with an absolute; the human's eye is the judge.

Check: `altbiomes.py` (a few minutes with 200 seeds). Planted regions: the number you planted (23 on version 13), each on its intended base biome, no one-point regions (edge under 20). Every alt biome `reaches min` at 100% (or you know why not). It also writes the regions overlay picture.

Check: `locations.py` (a minute or two). VERDICT `placed nothing:` should name only `Mistlands_Giant2` (crowded out by Giant1; cosmetic). Every progression and quest site at full count: StartTemple, Eikthyrnir, GDKing, Bonemass (at least 1 of 5; version 13 has 4), Dragonqueen, GoblinKing, the Queen's entrance, FaderLocation, DN_Bossroom, Haldor, Hildir's camp, crypt, cave and plains fortress, the Bog Witch, CharredFortress, the three PlaceofMystery. Every mountain, Deep North, harbour, frozen ship and lava location at full count. The `!` lines (short of quantity) should be the area-limited ones you expect: goblin towers, henges, giants, swords, some swamp huts, shipwrecks. Read the reason counts on each `!` line: `altitude` dominating means the biome sits at the wrong height; `area` means the biome is too fragmented for a Median location; `similar` means the group's spacing, i.e. area.

Step 6: render, and show the human

```
(ulimit -v 6000000; nice -n 10 python3 preview.py history/out-NN-2048 ../previews/NN-full.png
  2)
(ulimit -v 6000000; nice -n 10 python3 preview.py history/out-NN-2048
  ../previews/NNd-spawn.png -2 0.40 0.52 0.62 0.72)
(ulimit -v 6000000; nice -n 10 ~/upy/bin/python altmap_preview.py history/out-NN-2048
  ../previews/NN-full.png ../previews/NNg-alt-labelled.png)
```

`preview.py` draws land in its biome colour, shaded by height and lit from the north-west, water by depth with the wadeable shelf in light blue, lava orange, and land labelled Ocean in magenta (a bug flag). The scale argument downsamples (2 turns 2048 into 1024); a negative scale enlarges a crop. A crop is **four fractions of the map** (x0 y0 x1 y1, 0..1, y down); the tool refuses anything else. Number every picture so that the human can name it ("i like it 06").

> **Human:** Put the picture where the human will see it (for Wubarrk: the Artifact page, republished at the same URL), with one line: what changed, what is next. Then stop and wait. Do not go round again on your own; the next change is theirs to choose. If they say it is good, go on to step 7. If they ask for a change, go back to step 1.

Step 7: publish the version

When the human approves:

```
cp history/out-NN-2048/*.png history/out-NN-2048/*.txt ../fire-and-ice/
cp gen.py history/gen-NN-<short-name>.py
```

Update `../fire-and-ice/README.md` (the layout, the height scale table, the alt plants) and `../HANDOFF.md` (what version NN is, what changed, the measured numbers, what the human said).

Step 8: stage into the game profile

```
../stage-into-gale.sh "BC-Tester"
```

It refuses to run while Valheim is running (the config is read at launch). It copies the Better Continents DLL (the path you give it, else the repo's `dist/plugins/` build) into `BepInEx/plugins/Wubarrk-BetterContinents/` (renaming any other Better Continents DLL to `.bak` rather than overwriting a hardlinked file), and writes `BepInEx/config/BetterContinents.cfg` with `Directory` pointing at `fire-and-ice/`, `Heightmap Amount` from `mapscale.py`, the forest settings, `SelectedPreset = From Config`, `Debug Mode = true` (reveals the whole map on respawn, so that the world can be checked without sailing it; turn it off for a world meant to be played) and the alt-biome override `Wolf Mountain: enabled=false`. Appendix B shows the file. If the config did not change since the last staging (same Amount, same directory), restaging is not needed: the game reads the images at world creation.

Step 9: the human builds a NEW world

Tell them exactly this:

1. Quit Valheim if it is running. Launch it from Gale with the `BC-Tester` profile.
2. New World. A NEW name (never reuse a name: the old world's folder and cache would alias it). Set the preset dropdown beside the seed to **From Config**. Create.
3. It takes about 20 s longer than vanilla (`Genloc duration: 18089 ms` on version 13).

Do not build it for them, and do not touch their `worlds_local`.

Step 10: read the log

The log is `~/.local/share/com.kesomannen.gale/valheim/profiles/BC-Tester/BepInEx/LogOutput.log`. It is overwritten at every launch: copy it to `history/log-NN-<world>.txt` at once.

```
L=~/.local/share/com.kesomannen.gale/valheim/profiles/BC-Tester/BepInEx/LogOutput.log
cp "$L" history/log-NN-<world>.txt
grep -c "Failed to place all" "$L"
grep -E "Genloc duration|Alt biomes: planted|Alt biomes \(world load\)|refused|Altbiomemap
  file|Lavamap file|Selected .* for spawn|Exception" "$L" | grep -v "GetBiome error"
grep "Failed to place all" "$L" | sed -E 's/.*Failed to place all ([^,]+), placed ([0-9]+) out
  of ([0-9]+).*/\1 \2\/\3/'
```

Criterion, in order:

- `Loading [Better Continents 0.9.x]` (the exact version you staged), then `Alt biomes: patches bound` at startup.
- At creation: `applying selected preset 'From Config'`; `Time to load .../heightmap.png` and the other six files, `Altbiomemap file (2048)`, `Lavamap file (2048)`; nine `Selected <name> for spawn position` lines (one per pinned site).
- `Alt biomes: split 23 planted region(s)`, `Alt biomes: planted: 23 region(s), 23 alt biome(s) applied, ... 0 refused`; per alt biome, `+N planted (counted toward the maximum)` where you planted.
- `Genloc duration: ... ms, ... locations: 183` and a `Failed to place all` count near the checker's `!` count (34 on version 13). Every progression site absent from the failed list (Bonemass 4/5 is fine).
- No `Exception` from BetterContinents. `GetBiome error A -> B pos -> x, y` warnings once a second are vanilla's `Player.UpdateBiome` noticing that its 12 m sector grid disagrees with the exact biome at a shoreline: harmless.

Step 11: replay the log

```
(ulimit -v 6000000; nice -n 10 python3 locations.py ../fire-and-ice --log
  history/log-NN-<world>.txt) | tee history/out-NN-2048/locations-vs-game.log
```

Every line now carries `game: N/M` or `game: all placed`. Criterion: the checker and the game agree on which locations are short, and the counts are within a handful (Unity's random numbers differ from Python's; version 13 matched 148 of 170 lines exactly, the biggest gap GoblinCamp2 147 vs 118). A location the game placed far fewer of than the checker, or the reverse, is a modelling gap: read the line's reason counts and the location's rules in appendix C before changing the map.

> **Human:** Report in three lines: the counts (locations placed, short lines, planted regions), the sites (all progression sites placed, or which are not), and the one thing to look at in game. Then the human plays and judges. Their words ("looks good on my end", "reads too flat") start the next round at step 1.

Step 12: hand off

Update `HANDOFF.md`: the state table at the top, what the version is, the numbers, the open ideas and the working rules. Record the world's name and where its log copy is. A session that ends without a current handoff wastes the next one's first hour.

7 The traps that cost worlds

Each of these was found the hard way. The symptom is what you see; check the cause before you touch the map.

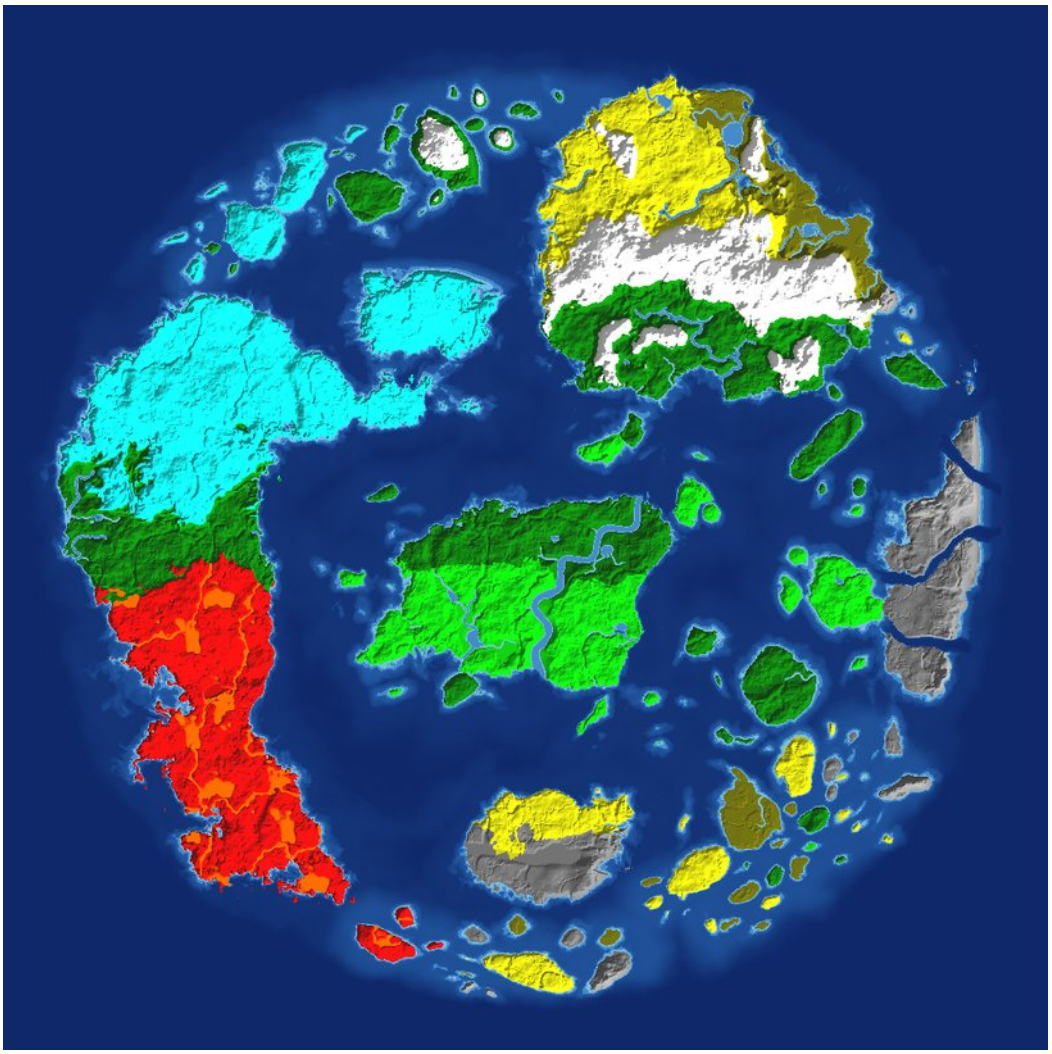
#	Symptom	Cause	Fix
1	Every lowland biome under water; the world is bare peaks in an empty sea	The waterline was taken as pixel 0.15 (the pre-1.0 value) instead of 0.30 / Amount	Derive the waterline (chapter 2); `check.py` on the correct scale
2	The whole map 25 to 30 m under water, evenly	The map was authored for one Heightmap Amount and the world was created at another	`mapscale.py` is the one source; stage, quit, relaunch, create
3	The staged config had no effect on the world just built	The config is read at game launch and baked at creation; the game was running during staging	Quit and relaunch before creating; the staging script refuses to run while Valheim runs
4	Ground far from the Ashlands behaves as Ashlands (heat, lava, ash spawns)	A non-zero heat pixel there; the mod's IsAshlands is `heat > 0`	Heat exactly 0 off the Ashlands; `check.py` requires it
5	You edited a PNG and the world did not change	Maps are baked into the world at creation	New world, new name
6	Hildir's cavern 0 of 3, tower ruins 0 of 50, frost caves 12 of 120	Amount 1.0: a 140 m ceiling; the locations want 100 to 200 m	Amount 2.0; peaks above 250 m
7	Frozen ships 0 of 50, Deep North wrecks 22 of 170, harbours 7 of 100	Water painted Ocean to the waterline; those locations want water of their biome	`label_shallows()`: Deep North to 16 m, Mistlands to 1 m
8	After labelling water, Birch Meadows and the Mistlands alts lost their hosts; 123 planted regions	A region whose edge is under water fails HeightAvg; a multi-biome colour spilled onto foreign shores	Label only what needs it, never against a foreign shore; pull multi-biome alt water 3 px back
9	Dozens of one-point alt-biome regions (71 "Lantern" Black Forest regions)	A multi-biome alt's colour claimed single grid points of a neighbouring biome	The pull-back above; patches painted 2 px inside their own biome
10	A planted patch used up a slot but reads as nothing in game	A spit or tongue was a lone point on the 12 m grid, a region of its own	Keep specks off; paint whole features; check `altbiomes.py` for regions with edge under 20
11	Relief added and nothing visible; "reads too flat"	FBM output is compressed round 0.5 (std 0.13); `ROLL` was worth a quarter of its metres	`ROLL_K = 1 / std`; measure with `relief.py`
12	Every coast a tilted plain climbing 20 m over 500 m	Relief faded in over the same ramp as the beach platform	Beach 210 m, relief from 125 m inland
13	Islands "parked" in a row	Every island displaced by the same warp vector	Anchor the warp at each feature's centre
14	A patch of Plains beside the Deep North	The warp evaluated a continent's field far from it	Leash each field to its unwarped ellipse plus slack

#	Symptom	Cause	Fix
15	The preview tool ran the machine out of memory and crashed the editor	Crop given in pixels; the tool wanted an image hundreds of times the map	Crops are fractions 0..1 (refused otherwise); every heavy command under `ulimit -v 6000000`
16	The Deep North list calls its ships "TarPit1"	`m_prefabName` is stale in the add-on location lists	Name a location by its prefab asset path (`_prefab_path`), as the checker does
17	A vanilla world, no mod, no warning	The preset dropdown was not `From Config`	Set it; the config's `SelectedPreset` pre-selects it
18	Two Better Continents DLLs; which one loads is a coin toss	The Thunderstore package and the fork share a BepInEx GUID	The staging script renames the other to `.bak`
19	A mod file corrupted in every Gale profile at once	`cp` over a hardlinked DLL wrote through Gale's shared cache	Check `stat -c %h`; rename, never overwrite
20	A world vanished from the list, or its map was orphaned, after a rename or copy	Valheim's save plumbing carries only its own extensions and drops the mod's sidecar	Never touch the human's saves; the mod's `SaveSystemPatch` compensates in 0.8.x
21	Weather flips at sea on a map with Deep North in its middle rows	Vanilla passes the camera HEIGHT to `IsDeepnorth`; the mod fixes it on worlds with a biome map	Use 0.8.1; the log says `Deep North weather: EnvMan asks IsDeepnorth at the camera's x and z`
22	World creation hangs forever on the progress bar (0.7.x)	A biome missing from the map made a vanilla index throw inside the location coroutine	Fixed in 0.8.0; on older versions include every biome
23	`GetBiome error A -> B pos` once a second	Vanilla's `Player.UpdateBiome` comparing the 12 m sector grid with the exact biome at a shoreline	Harmless; ignore
24	Wolf Mountain appears on the snowy archipelago island as well as where painted	Random placement added one (about one world in five)	`Overrides = Wolf Mountain: enabled=false` in the alt-biome config
25	A painted colour planted nothing	It matched no legend colour within 20 RGB units, or the alt biome does not belong to that base biome	The mod's warning at bake lists the unmatched colours; `bc ab info` in game
26	A quantity-10 location placed 1, or 5 of 10	It is unique: candidates are placed, one is spawned, the rest removed	Expected; the checker's `!` mark ignores it
27	Alt biomes differ between server and clients	Vanilla syncs nothing; clients regenerate from their own maps	The same world file everywhere; 0.8.1 pushes the server's placement and logs a match or a warning
28	A per-type decompile says the game caches the alt-biome grid	A stale `Decompiled_<version>/` folder from an older build; since 1.0.15 the game regenerates the grid on every load	Confirm details in `libs-Tools/1.0/DECOMPILED/assembly_valheim.decompiled.cs`, and check the per-type folder is the current build's
29	You used `libs-Tools/OLD/` as the reference	It is 1.0.7, with a complete, current-looking decompile	`libs-Tools/1.0/` and the current `Decompiled_<version>/` (with trap 28); compare the md5 of `assembly_valheim.dll` with the installed game's
30	The minimap reads as forest everywhere	A minimap mask patch in the mod (before 0.8.0) collapsed biome channels	Not tree density; update the mod

> **Trap:** The one that repeats: staging while the game runs (3). It is silent, it looks like a generator bug, and it wastes a world. The staging script's refusal is the guard; do not work around it.

8 A validated build, end to end

Version 13 of ProximaMaxi, built by the human as the world `ProxEra01` on 23 September 2026 and approved ("looks good on my end"). These are the numbers a good round produces; compare yours with them.



Version 13 as `preview.py` renders it: the 21 km square, north up. The crescent in the west (Deep North on its northern arm; the Ashlands arm to the south-west with burning isles and lava), the spawn continent at the centre with the starter river, the second continent to the north-east with its 250 m range, the Mistlands chain on the eastern edge, a Plains and Mistlands landmass to the south, two archipelagos and four rim chains.

The generator (`gen.log`)

```
  79 islands with satellites, 74 in archipelagos
  terrain 35s
  rivers: 65 mouths, 144 reaches, 43 km of water, 29 km of dry gully, 16 lakes (33 basins
  offered)
  shallows: 65277 px of Deep North and Mistlands water given the biome of their coast
  alt Mushroom                34795 px
  ... 22 more alt lines (Lantern: global biome, every piece kept) ...
  StartTemple          -> world (   -108,   -2201)  [c1meadows]
  ... eight more pinned sites ...
done 2048
```

check.py

```
2048x2048   land 27.7% of the image   (reference world Era02: 33.8%)
highest 253 m   deepest -60 m

biome         % map  % dry   mean    min    max
Ocean         70.13    0.0  -46.2  -60.0   -0.0
DeepNorth      6.69   79.2   64.6  -16.0  131.3
BlackForest    6.41   96.3   20.2   -8.2   80.3
AshLands       3.79  100.0   43.7   -1.2   78.4
Plains         3.17   98.2   20.1   -7.3   67.1
Meadows        3.04   94.9   14.5   -6.8   35.6
Mountain       2.72  100.0  132.7   39.5  252.9
Mistlands      2.60   93.6   23.1   -1.0   98.3
Swamp          1.44   90.7    5.8   -7.8   64.3

dry ground labelled Ocean: 0
heat > 0 on land that is not AshLands: 0
AshLands pixels with no heat: 0
max heat 0.160

  StartTemple                     Meadows        16.0 m     2203 m from centre
  Dragonqueen                     Mountain      181.4 m     6015 m from centre
  Mistlands_DvergrBossEntrance1   Mistlands      25.9 m     7976 m from centre
  ... (nine sites)
```

Deep North is 79% dry because its water to 16 m carries the biome (the frozen ships); the other sub-100% figures are rivers and lakes. Swamp's mean of 5.8 m is the bog Bonemass needs.

relief.py

```
biome        rms dh 50 m   rms dh 250 m   (metres)
BForest          8.05        25.36
DeepNorth       12.08        25.39
Ashlands         9.07        17.65
Plains           6.82        19.48
Mountain        16.91        53.06
Meadows          4.02         9.07
Mistlands        6.11        17.78
Swamp            9.12        31.89

mean height (m) by distance from the water, 50 m bands from the shore outward:
Meadows     5.5  11.8  15.6  17.7  18.5  19.4  20.6  21.6  21.9  21.5 ...
Swamp       2.9   5.9   7.7   8.8  10.0  10.6  13.2  14.1  14.6  12.0 ...
```

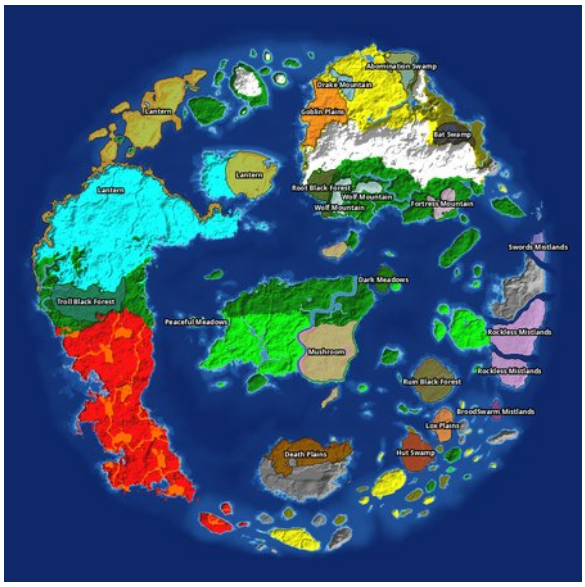
altbiomes.py

```
309 sectors; 295 are land; 23 planted
  AshLands (global): edge 4010, dist 7583 m
  DeepNorth (global): edge 6335, dist 5589 m

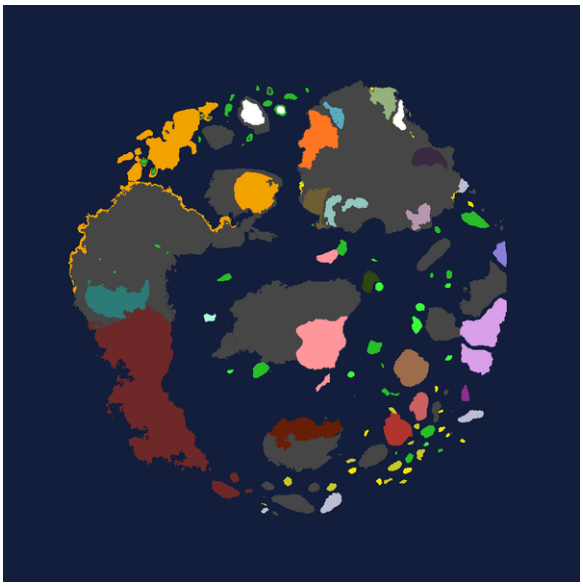
Planted regions (they ignore the placement rules):
  #303   Wolf Mountain            Mountain    at (  1709,  3508)  3903 m  edge  341  h 106.4
  #297   Lantern                  DeepNorth   at ( -5554,  4599)  7211 m  edge 5265  h  82.2
  ... 21 more ...

alt biome                 targets                      dist      edge min-max plant valid
  total  reaches min
Mushroom                  Meadows+BlackForest           500 [ 66,199)   1-3       3    20
  3.00  100.0%
Wolf Mountain             Mountain                     2000 [199,299)   1-5       2     2
  2.46  100.0%
Rockless Mistlands        Mistlands                    2000 [ 99,599)   2-5       2     3
  2.63  100.0%
```

```
... all 32 at 100.0%
```



13g-alt-labelled.png



13h-altbiome-regions.png

Left: the planted alt biomes labelled over the preview (`altmap_preview.py`). Right: the regions overlay `altbiomes.py` writes: legend colour = planted, bright = can host an alt biome at random, grey = cannot (too big, too small, too close to the centre, or an edge under water).

`locations.py`

```
170 locations from the game (... + the add-on lists), 16 more added by alt biomes (listed only)
location map sites registered first: Bonemass (4191, 6663), GoblinKing (1472, 6407), ...
  StartTemple                          1/1    P Meadows
  Eikthyrnir                           3/3    P Meadows
! Bonemass                             4/5    P Swamp      [zoneTaken 90, area 25324, biome
  97, altitude 197688, terrainDelta 1937, similar 7776]
! SunkenCrypt4                      136/175  P Swamp      [zoneTaken 3839, area 25272, biome
  163, altitude 182721, terrainDelta 97, similar 1812]
  Dragonqueen                          3/3    P Mountain
  DN_Bossroom                          3/3    P DeepNorth
  Hildir_cave                          3/3    P Mountain
  StoneTowerRuins04                   50/50     Mountain
  MountainCave02                     120/120    Mountain
  FrozenShip01_DN                     50/50     DeepNorth
  ShipWreck01_DN                     170/170    DeepNorth
  LeviathanLava                      100/100    AshLands
! StoneTower3                          1/50     Plains     [... similar ...]

VERDICT
  placed nothing: Mistlands_Giant2
  progression and quest sites: StartTemple 1/1 Eikthyrnir 3/3 GDKing 4/4 Bonemass 4/5
  Dragonqueen 3/3 ...
```

Bonemass's reason counts say what the swamp is: of 60,000 attempts, 197,688 point tests failed on altitude (the bog is mostly above 2 m) and 25,324 zones on area; 4 altars placed, which is enough. The Goblintower group's `similar` count is the plains' 14 km^2 against 100 towers 512 m apart.

The game (`log-13-ProxEra01.txt`)

```
[BetterContinents] [Saving][ProxEra01] bWorldBeingCreated flag set, first time save of
  ProxEra01, applying selected preset 'From Config'
[BetterContinents] Time to load .../fire-and-ice/heightmap.png: 157 ms
[BetterContinents] Alt-biome legend altbiomemap.txt: 19 colours, 0 points.
[BetterContinents] Alt-biome map .../altbiomemap.png: 2048x2048, 329922 planted pixels,
  decoded in 74 ms.
[BetterContinents] Selected StartTemple for spawn position 0.4941406, 0.3955078      (and
  eight more)
[BetterContinents] Alt biomes: split 23 planted region(s) out of 22 region(s) (13 fully
  planted and reused), 227235 of 4194304 grid points planted.
[BetterContinents] Alt biomes (world load): mode Random, placement seed world seed, grid 2048
  x 12 m to 10500 m (WorldEdge), 296 sectors, 63 with alt biomes (23 planted, 40 random), 23
  planted region(s)
[BetterContinents]   Mushroom [Meadows|BlackForest] 0/1-3 chance 0.2 +3 planted (counted
  toward the maximum): 20 of 110 regions qualify ...
[BetterContinents]   Wolf Mountain [Mountain] 0/0-5 chance 0.2 +2 planted (counted toward the
  maximum): disabled for random placement
Loading: Done. ... Genloc duration: 18089.898 ms, (... locations: 183 ...)
Failed to place all Bonemass, placed 4 out of 5 with 0 tries in 00:00:00.8604500
  (34 such lines)
```

The 34 short lines, as `placed/quantity`:

```
Bonemass 4/5   SunkenCrypt4 135/175   Mistlands_DvergrTownEntrance1 64/120
  Mistlands_DvergrTownEntrance2 42/120
BogWitch_Camp 5/10 (unique)   SwampRuin2 22/30   GoblinCamp2 118/200   StoneTower1 15/50
  StoneTower3 1/50
StoneHenge4 4/5   TrollCave02 144/200   SwampHut5 11/25   SwampWell1 15/25   ShipWreck03 6/25
  ShipWreck04 2/25
Runestone_Swamps 92/100   BigRockClearing 1/10 (unique)   Mistlands_GuardTower3_ruined_new
  33/50
Mistlands_Lighthouse1_new 40/100   Mistlands_Viaduct2 63/150   Mistlands_Giant1 158/250
  Mistlands_Giant2 0/85
Mistlands_Swords2 26/33   Mistlands_Swords3 2/33   NorthVillage 98/135   WoodVillage2 5/15
StoneTowerRuins07..10_sunk 0/10 each   SwampHut1_1 8/50   SwampHut2_1 1/50   SwampHut3_1 2/50
  GoblinCamp2_1 0/5
```

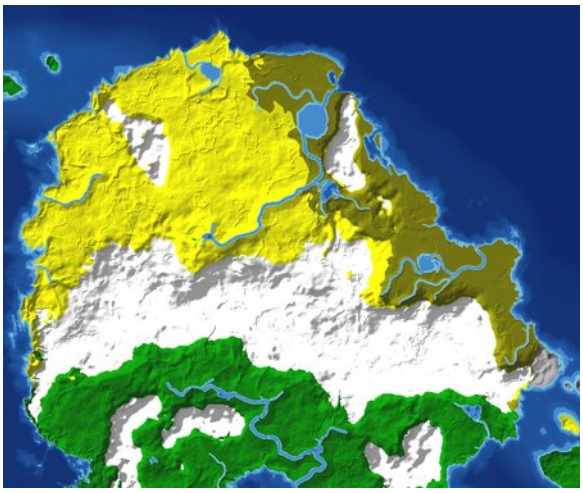
The last nine are alt-biome add-on locations, placed only inside their alt biome's regions (Ruin Black Forest's sunken towers want Black Forest shallows the map does not label; a cosmetic gap). Everything mountain, Deep North, harbour, frozen ship and lava, and every progression site, is absent from the list: placed in full.

The replay (`locations.py --log`)

148 of 170 lines matched the game exactly; the other 22 were within a handful, with the checker mostly a little higher (GoblinCamp2 147 vs 118, TrollCave02 160 vs 144, SunkenCrypt4 136 vs 135) and sometimes lower (GuardTower3 19 vs 33, Lighthouse 36 vs 40). That is the checker's known error: Python's random numbers are not Unity's, and the vanilla forest and Mistlands vegetation noise are modelled, not reproduced. The verdicts agreed.

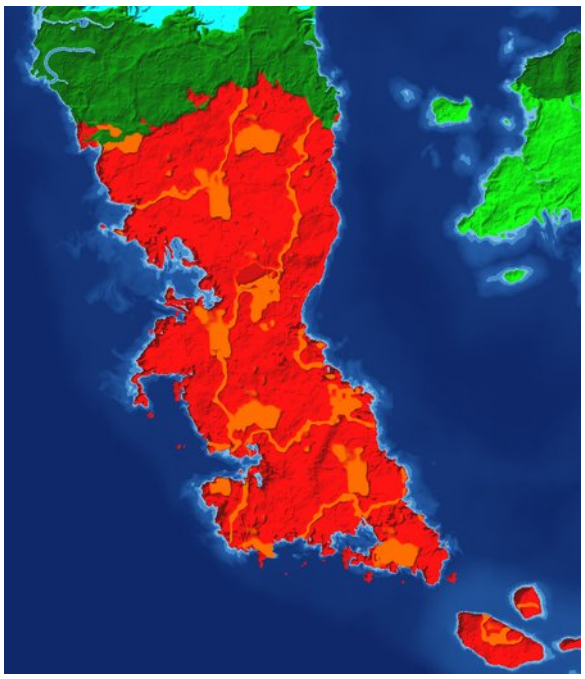


13d-spawn.png



13a-northeast.png

Close-ups from the previews the human saw: the spawn meadow with the starter river and Mushroom east of it (left); the second continent's range with Wolf Mountain's twin peaks and the Deep North isles beyond (right).



The Ashlands arm, with lava pools and streams drawn orange, the burning isles off its tip and the heat ramp's sea.

The whole round, from "the world reads too flat" to the approved world, took one working session: measure (`relief.py` was written for the purpose), change (`ROLL`, the ramps, the bog), regenerate, four checks, three previews on the page, the human's approval, publish, no restaging (the config had not changed), the human's new world, the log, the replay.

9 Export and import

A second way to start a map: instead of `gen.py` building one from nothing, `bc_export` writes one out of a world that already exists — vanilla or Better Continents, yours or someone else's — as exactly the kind of files chapter 2 already taught you to read. `bc_import` (or a New World preset the export makes for you) turns such a folder, edited or not, back into a new world. Both are console commands built into Better Continents 0.9.0 and later; neither needs this workspace's toolkit.

- **A real starting point.** Exporting a vanilla world gives you a heightmap, biome map and location map that Valheim's own generator built, with every location already placed on it — something to paint alt biomes or a terrain map onto, or to compare against `gen.py`'s output.
- **A second opinion on your own map.** Export the world you built from `fire-and-ice/` and read its `manifest.json` and `locationmap.txt` back, instead of trusting only a hand-copied log (chapter 6, step 10-11).
- **A piece to paste.** A river mouth, an island, a crater: cut it from an exported world's `heightmap.png` and `biomemap.png` and paste it into `fire-and-ice/`'s, or the reverse. See "Cutting and pasting" below.

> **Rule:** `bc_export`'s own defaults are Heightmap Amount 2 and Sea Level 0.5 — the same two numbers `mapscale.py` uses (chapter 2). Never override them (`bc_export amount=... sealevel=...`) for a map you mean to combine with this workspace's own maps; left at the default, the two line up pixel for pixel.

`bc_export`: what it writes

Console, any player, not a cheat command (it works for a guest on someone else's server too, within what that guest's client can see):

```
bc_export [size] [amount=2] [sealevel=0.5] [heatscale=10] [edge=auto|on|off]
  [forest=additive|exact]
          [noheight] [nobiomes] [nolocations] [noforest] [noheat] [noaltbiomes] [nolava]
  [nomoss]
          [nopaint] [nosources] [nopreset] [perpoint]
bc_export status | cancel | server <the same options> | help
```

`size` is one number, 128 to 8192 px a side (the in-game window offers 1024 / 2048 / 4096 / 8192; 4096 is the default). In a loaded world the host also has `bc export run|status|cancel` in the debug tree. A status box (**Hud Hotkey**, F9 by default) and a window with an Export tab and an Import tab (**Window Hotkey**, F7) do the same thing with buttons instead of typing, once `[09 BetterContinents.Export] Hud` is turned on; on a server, its **Hud** and **Allow Export** apply to every connected client too.

It writes into `<save data>/BetterContinents/<world>/export-<time>/` — on Linux, `~/.config/unity3d/IronGate/Valheim/BetterContinents/<world>/export-2026-09-24-14-05-33/`, a sibling of `worlds_local/`, not inside it:

File	What it holds
`heightmap.png`	16-bit height, chapter 2's own encoding
`biomemap.png` + `.txt`	the biome, in Better Continents' own colours, and their legend
`locationmap.png` + `.txt`	one pixel per location, and its legend (see the client trap below)
`altbiomemap.png` + `.txt`	every alt biome exactly as placed — only written if the exported world had Better Continents 0.8.1+ and alt biomes on it
`forestmap.png`, `heatmap.png`	forest factor, Ashlands heat

File	What it holds
`lavamap.png`, `mossmap.png`, `paintmap.png` (+`.txt`)	Ashlands lava, Mistlands ground cover, terrain paint — the guide teaches these; ProximaMaxi does not use them
`sources/`	only if the exported world was itself Better Continents: its own maps as its settings kept them, for reference (not loaded back)
`export.cfg`	the settings that rebuild the world (a `[NN Group]` file, the same shape as appendix B's)
`README.txt`	a walkthrough for someone who has never done this: the ways to make a world from the export, what every file is, the numbers
`manifest.json`	the same numbers (`size`, `heightmapAmount`, `seaLevel`, `totalSize`...) as machine-readable JSON

Every map is on by default (the `no...` switches turn one off), and a New World preset is made too unless you pass `nopreset` — see "Presets and `bc_import`" below. `bc_export status` and `bc_export cancel` work while it runs; a large export (8192 px) takes minutes and close to a gigabyte of memory while it runs, and close to two more while its preset is built.

> **Trap:** an export made on a guest's own client only knows the locations the server has shown that player on the map so far — not the full set. `locationmap.png` is written anyway, with Skip Default Locations left off so the game places the rest, and both `bc_export status` and the export's own README.txt say so. Export on the host, or have an admin run `bc_export server ...` on the dedicated server, for a location map worth trusting.

> **Trap:** below 2048 px a pixel is wider than the game's 12 m alt-biome sector grid (chapter 3), so `altbiomemap.png`'s region borders can move by up to a pixel — the same reason `gen.py` never runs below 2048 (chapter 2). Export at 2048 px or more whenever alt biomes matter.

Cutting and pasting between an export and a generated map

Two maps line up pixel for pixel, so an area cut from one pastes straight into the other at the same place, when all of these agree:

- **Size** (pixels per side): `gen.py 2048 ...` pastes onto a `bc_export 2048 ...`, not a `4096` one.
- **Heightmap Amount and Sea Level**: `mapscale.py`'s `AMOUNT` and Better Continents' default Sea Level of 0.5 — which is also `bc_export`'s own default (its `amount=`/`sealevel=` options, or the config's `Default Heightmap Amount` / `Default Sea Level` under `[09 BetterContinents.Export]`, appendix B).
- **World Size and Edge Size**, i.e. the square's total width (`totalSize` in `manifest.json`). `gen.py` assumes the vanilla square, 21 km (10,000 m World Size + 500 m Edge Size, chapter 2). Exporting a **vanilla** world always uses those same two numbers. Exporting a **Better Continents** world uses whatever World Size/Edge Size *that* world was built with, which only matches the generator's square if it, too, used 10,000/500.

When all three agree: paste the same area from every map you keep (heights, biomes, forest, alt biomes, heat) so they stay consistent, and copy over any legend line (`.txt`) the other side's legend lacks. Heights blend over about a pixel at the seam; a biome pixel is read as whichever of the four nearest pixels is in the majority; a location dot must stay a single pixel touching no other dot of its colour. Only the *generated* world is exported — buildings, terrain edits and anything placed after world creation are not — so an export is a starting point, never a savegame.

> **Check:** before trusting a paste, open both folders' `manifest.json` and compare `size`, `heightmapAmount`, `seaLevel` and `totalSize`. All four equal is the whole rule; anything else and the pasted area lands at the wrong place, the wrong height, or both.

Presets and `bc_import`

A folder of maps — an export, or `fire-and-ice/` itself — becomes a New World choice one of three ways:

- **A preset** (`bc_import`'s default): a self-contained `<name>.BetterContinents` file plus a picture, in `<save data>/BetterContinents/presets/`. It keeps working if the folder is later moved or deleted, and it freezes the maps as they were at import — edit the PNGs afterwards and re-import to update it. Every successful `bc_export` already makes one, named `"<world> <time>"`; it appears in the New World dropdown but is **not** selected, so exporting a world never silently changes what the next New World is made from.
- **The config** (`bc_import ... config`, or the HUD's "Use as config"): copies `export.cfg`'s settings into `BepInEx/config/BetterContinents.cfg` (the old file is kept beside it, timestamped), points `Directory` at the folder, and sets `SelectedPreset = From Config`. Every New World made with **From Config** afterwards reads the folder's PNGs as they are *then* — the same effect `stage-into-gale.sh` gets for `fire-and-ice/` by hand (appendix B), except this way works from inside a running game, because Better Continents now re-reads its config file whenever it changes, not only once at launch (no more quitting and relaunching to pick up an edited config before the next New World).
- `Directory` **alone** (Better Continents 0.9.4 and later): set only `[00 BetterContinents.Debug] Directory` to an export folder and create the New World with **From Config**. When the folder holds an `export.cfg`, Better Continents lays it over `BetterContinents.cfg` for that world exactly as an import builds its preset: `export.cfg` wins for every key it lists; `Directory` stays the config's own, so a moved or copied export loads from where it is now (the `Directory` line inside `export.cfg` names the folder it was written in, and is ignored); nothing is written to either file; keys that are not world settings (`Debug Mode`, the `[09 BetterContinents.Export]` group, `SelectedPreset`, unknown keys) are skipped as in an import. `BepInEx/LogOutput.log` says `Directory <folder> holds a world export, so the new world uses its export.cfg over BetterContinents.cfg` and then one line per setting that differs, e.g. `[02 BetterContinents.Heightmap] Heightmap Amount = 2 (BetterContinents.cfg: 1)`. A folder with no `export.cfg` — `fire-and-ice/` included — is read exactly as before, every setting from `BetterContinents.cfg`. The overlay also applies after `bc_import ... config`, which points `Directory` at the folder too: to use a value of your own, edit `export.cfg` or delete it, never just `BetterContinents.cfg`. Presets carry their own settings and are unaffected; so is every existing world.

> **Trap:** a heightmap read at the wrong Heightmap Amount drowns the world. An export stores heights for Amount 2 and Sea Level 0.5: `metres = (pixel/65535 × amount − 0.15 + Lerp(1, −1, sea level)) × 200`, waterline at pixel 0.15, black −30 m, white 370 m. Read at the config's default Amount 1, the waterline moves to 0.30: the export reported on 2026-10-02 had 30.7 % land at Amount 2 and 2.5 % at Amount 1, a world about 99 % ocean. Before 0.9.4 that is what `Directory` alone did with an export folder; it also kept the config's Forestmap Multiply/Add (1/1, the export needs 0/1: wrong forest density), Skip Default Locations (false: a second set of locations next to a complete location map) and alt-biome `Mode` (Random: extra alt biomes next to the planted ones). In any version, a bare `heightmap.png` copied without its `export.cfg` needs Heightmap Amount 2 and Sea Level Adjustment 0.5 set by hand: the same two numbers as `mapscale.py`, so this workspace's own staging already matches.

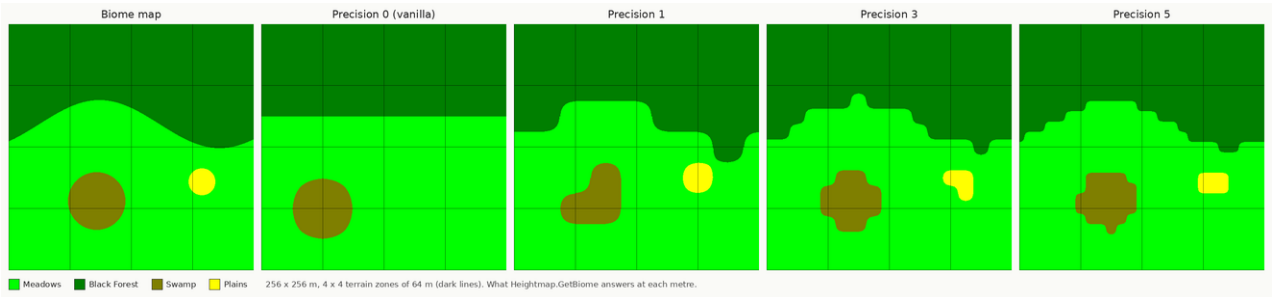
Command	What it does
`bc_import`	the newest finished export of the loaded (or last-played) world; makes and selects a preset
`bc_import list`	numbers every export folder, newest first, with world, time, size
`bc_import <number>`	the export on that line of the list
`bc_import <world>`	that world's newest finished export
`bc_import <folder>`	any folder of maps: a full path, or one under `BetterContinents/`
`bc_import ... config`	the same target, applied to `BetterContinents.cfg` instead of a preset
`bc_import status` / `help`	what the last import did / this table

`bc_import` is not a cheat: it works for any player, in the main menu or in a world (there is no New World screen to refresh from the main menu, but the console still builds and selects the preset, ready for when you open one). The host's debug tree has the same thing as `bc import run|list|status`.

> **Rule:** whichever way you load a folder, always **create a new world**. Better Continents only reads maps when a world is created (chapter 6's trap 5, "you edited a PNG and the world did not change"); a preset or `From Config` changes what the *next* New World is built from, never a world that already exists.

Biome precision

`[03 BetterContinents.Biomemap] Biome precision` (0-5, console `bc b p`) works again from Better Continents 0.9.0. At 0 (vanilla), everything inside a 64 m terrain zone — ground texture, grass, vegetation, spawn points — takes its biome from the zone's four corners, so a painted border only really shows up on that 64 m grid no matter how precisely you painted it. At 1 to 5 the zone is split into `(precision + 1) x (precision + 1)` cells, each reading the biome map on its own, down to a corner every 11 m at precision 5.



What `Heightmap.GetBiome` answers across four 64 m zones, drawn offline from the mod's own grid code (not a screenshot): at precision 0 a Swamp island or a Plains sliver either swallows a whole zone or disappears, because only the four corners are ever asked; by precision 5 the shape you painted survives.

> **Check:** precision changes what a point **looks like and grows**, never where a **location** or a **planted alt biome** is *allowed*. `IsBiomeEdge`, a location's biome-area test (chapter 3) and an alt-biome region's own corner/sector rules (chapter 3, appendix C) all keep vanilla's four-corner, 12 m answer regardless of Biome precision — `locations.py` and `altbiomes.py` model them correctly whatever precision you set. Precision never changes terrain height either.

> **Human:** nothing about Biome precision has run in a real game yet as of 0.9.0 — not the ground textures, not vegetation and spawns at precision 1-5, not `bc b p` while zones are loading. Say so plainly if you turn it on for someone. It is checked offline against the mod's own test suite, including a byte-identical comparison with vanilla's terrain builder at every precision level, but the human is the first to actually see it rendered.

Multiplayer: the settings download

A joining client receives the world's settings and maps (the same data an export holds, packed for the network) from the host or dedicated server before the join finishes, and caches them by id, so the transfer happens once per world revision. Valheim pins every Steam connection to about 150 KB/s; a 2048 px world is 11.8 MB (about 93 s), a 4096 px world with a paint map several times that. Two 0.9.0 additions:

- `[07 BetterContinents.Misc] Settings Transfer Rate` — `Vanilla` (leave Valheim's rate), `KB256`, `KB384`, `KB512` (the default), `KB768`, `MB1`, `MB1_5`, `MB3` or `Unlimited` (a fixed 100 MB/s, not a literal 0). Read on the sending side only, at the moment a client joins: a client's value does nothing, a PlayFab (crossplay) connection cannot be raised at all, and Better Continents re-reads the file on change, so no restart. Two facts that bit on 2026-09-25: GameNetworkingSockets has NO adaptive bandwidth, a connection sends at its SendRateMin, so both SendRateMin and SendRateMax are set on the one connection (raising only Max changed nothing); and a dedicated server's sockets belong to the game-server Steam interface (`SteamGameServerNetworkingUtils`), where the client wrapper throws "Steamworks is not initialized". The transfer loop itself (two 128 KiB chunks in flight per server frame) tops out near 4 MB/s. The log shows `Settings transfer rate: ... (server setting)` and `Settings sent: <bytes> in <s> (<KB/s>)`.
- `bc_cache` — `bc_cache export` on the machine that runs the world (or on a client that already holds the world) writes `<world>-<id>.bcworld` plus a `.txt` note into `<save data>/BetterContinents/<world>/`; `bc_cache import <file or folder>` (main menu or in a world) puts it in the local cache, never overwriting, with the id recomputed from the bytes; `bc_cache list` shows the cache. At startup every `.bcworld` under `BepInEx/plugins/` and in `BepInEx/config/BetterContinents/seed/` is imported, so a mod pack holding nothing but the file pre-seeds every installer. The id changes with any change to the world's Better Continents settings, so re-export after editing the world.

> **Human:** the rate presets ran for real on 2026-09-25, a client joining a local 1.0.15 dedicated server with an 11.8 MB world: 78 s at `Vanilla`, 23 s at `KB512`, 3 s at `Unlimited`. A pre-seeded join (`bc_cache`) has not run in a real game yet; it is covered by the offline suites (tools/livecfg-tests) only. The 93 s baseline is the ProximaMaxi server's, at Valheim's own rate.

Traps

#	Symptom	Cause	Fix
1	Two maps (an export and a generated one, or two exports) pasted together, but the piece lands in the wrong place or at the wrong height	Their Size, Heightmap Amount, Sea Level or World Size/Edge Size (`totalSize`) differ	Compare all four in `manifest.json` first; see "Cutting and pasting" above
2	A pasted or hand-authored area sits a fixed number of metres off from the rest of the map	Sea Level (or Heightmap Amount) differs between that piece and the whole	Keep every piece at the export defaults, Amount 2 / Sea Level 0.5, the same as `mapscale.py`
3	An imported export is missing most bosses, dungeons and vendors	It was `bc_export`ed on a client, which only knows the locations the server has shown that player	Export on the host, or `bc_export server ...` on the dedicated server (admins)
4	You imported a folder, or edited its PNGs afterwards, and the world you are in did not change	Maps are only read when a world is **created**	Create a NEW world from the preset or `From Config`; an existing world keeps what it was made with
5	An exported alt-biome map's regions do not match what `altbiomes.py` predicted for the same PNGs	The export was below 2048 px, wider than the game's 12 m alt-biome grid	Export at 2048 px or more whenever alt biomes matter

#	Symptom	Cause	Fix
6	Players still wait a minute or more to join after `Settings Transfer Rate` was raised	It was set on a client, or the connection is PlayFab (crossplay)	Set it in the server's (or host's) config; or hand out a `.bcworld` (`bc_cache export`) so the join needs no download
7	A world made `From Config` with `Directory` set to an export folder is almost all ocean	Before 0.9.4 that path ignored `export.cfg`: the heightmap was read at the config's Heightmap Amount 1, not the export's 2	Better Continents 0.9.4 or later; or the export's preset, `bc_import`, or `bc_import ... config`. A bare `heightmap.png` needs Amount 2 / Sea Level Adjustment 0.5 by hand
8	A setting changed in `BetterContinents.cfg` makes no difference to the new world	`Directory` points at a folder whose `export.cfg` lists that key; since 0.9.4 `export.cfg` wins	Edit `export.cfg`, or delete it; the log lists every setting where the two differ

Appendix A

Tool reference

All in `generator/` unless stated; all standard-library Python 3 except `altmap_preview.py` (Pillow). Run each as `(ulimit -v 6000000; nice -n 10 ...)`. Each tool's docstring is its manual: `head -30` it.

Tool	Usage	Reads	Writes or prints	Time at 2048
`gen.py`	`gen.py <size> <out dir>`	`mapscale.py`	the seven maps and three legends into `<out dir>`; a log of islands, rivers, alt plants and pinned sites	40 s, all cores
`mapscale.py`	imported		`AMOUNT`, `SEA`, `M_PER_UNIT`, `CEILING`, `FLOOR`, `metres(v)`, `value(m)`	
`png.py`	imported		`write_gray16`, `write_rgb8` (zlib only)	
`check.py`	`check.py <map dir>`	height, biome, heat and location maps	land %, per-biome dry % and heights, the three invariants, the pinned sites	20 s
`relief.py`	`relief.py <map dir>`	height, biome	RMS height change over 50 and 250 m per biome; the coastal height profile	1 min
`preview.py`	`preview.py <map dir> <out.png> [scale] [x0 y0 x1 y1]`	height, biome, lava	a shaded picture; scale 2 gives 1024 px, a negative scale enlarges; the crop is in fractions	30 s
`altmap_preview.py`	`~/upy/bin/python altmap_preview.py <map dir> <base.png> <out.png> [crop]`	the alt map and legend, a preview	the preview with the planted regions tinted, outlined and labelled	20 s
`altbiomes.py`	`altbiomes.py <map dir> [overlay.png] [seeds]`	height, biome, alt map; `../altbiomes/altbiomes_1.0.15_full.json`	sectors, planted regions, per-alt-biome eligibility and `reaches min` over the seeds; the regions overlay	2 to 5 min
`locations.py`	`locations.py <map dir> [--amount A] [--log LogOutput.log] [--only a,b] [--seed N] [--scale f]`	all maps; `../locations/locations_1.0.15.json`; the alt data	every location's placed/quantity with reason counts, the SHORT list, the VERDICT; with `--log`, the game's count on each line	1 to 2 min
`../locations/extract_locations.py`	`extract_locations.py [--game-data DIR] [--out DIR]`	the game's bundles, read-only; needs the `lz4` CLI and `tools/`	`locations_1.0.15.json`, `_summary.txt`	1 min
`../altbiomes/extract_altbiomes.py`	`extract_altbiomes.py [--game-data DIR] [--out DIR]`	the game's bundles, read-only; the same `tools/` beside it	`altbiomes_1.0.15_full.json`, `_summary.txt`	1 min
`../stage-into-gale.sh`	`stage-into-gale.sh "<profile>" [dll]`	`mapscale.py`, the DLL	the plugin and `BetterContinents.cfg` in the Gale profile; refuses while Valheim runs	

Two more tools live in the game itself, not in this repository, from Better Continents 0.9.0: chapter 9 covers both.

Tool	Usage	Reads	Writes	Where
`bc_export`	console, in a loaded world	the world you are standing in	a folder of maps under `<save data>/BetterContinents/<world>/export-<time>/`, plus a New World preset	any player (server-gated)

Tool	Usage	Reads	Writes	Where
`bc_import`	console, main menu or in a world	a folder of maps (an export, or one of this workspace's)	a New World preset, or `BetterContinents.cfg`	any player

What the checkers reproduce, and what they cannot

`altbiomes.py` replays `AltBiomeWorldData` step for step (the grid, the flood fill, the global sectors, EdgeCount, HeightAvg, DistanceFromCenter, `CanAddModifier`) and the mod's 0.8.1 planting (the split, the reuse, the quota). It does not reproduce Unity's random numbers, so random placement is simulated over many seeds with the same algorithm (biome order, list order, minimum then chance, maximum, incompatibility): it says how reliably each alt biome finds a home, not where a given world seed puts it.

`locations.py` replays `ZoneSystem.GenerateLocations` (the order, the attempt counts, the candidate source, zone occupancy, biome area, six points per zone, distance, biome vote, altitude, forest, terrain delta, similar and group spacing, vegetation) with the location map's sites registered first. It cannot reproduce Unity's random numbers (counts are typical, not the seed's), the vanilla forest and Mistlands vegetation noise under a forest map (modelled as noise of the right range), or the alt-biome-added locations and blocks (listed, not placed). Validated against real builds: it matched the game line for line on the failures that mattered (the cavern 0/3, tower ruins 0/50, frost caves 12/120, frozen ships 0/50, Deep North wrecks 22/170 on the Amount 1.0 map; 148 of 170 lines exactly on version 13).

When the game changes

A new Valheim build changes the location and alt-biome lists. Re-extract them (both extractors are read-only on the game data and need only Python, the `lz4` CLI and the small `tools/` copies of the libs-Tools asset readers), diff the summaries against the old ones, and re-run both checkers. Check the game's version by the md5 of `assembly_valheim.dll` (every Valheim assembly is stamped 0.0.0.0; compare it with the installed game's copy), and confirm any claim about the game in the whole-assembly decompile at `libs-Tools/1.0/DECOMPILED/assembly_valheim.decompiled.cs` (the per-type `Decompiled_<version>/` is faster to read; check it is the current build's, and `libs-Tools/OLD/` is 1.0.7).

A new Better Continents version: read its CHANGELOG and `ALTBIOMES.md`. The height formula, the legend formats and the log lines in this document are 0.9.x's (unchanged from 0.8.1: 0.9.0 adds export, import and Biome precision, chapter 9, none of which touch how heights, biomes or alt biomes are placed).

Appendix B

The config that goes with the map

`stage-into-gale.sh` writes this into `<profile>/BepInEx/config/BetterContinents.cfg`. The comments are the reasons.

```
[00 BetterContinents.Debug]
Enabled = true
Debug Mode = true            # reveals the map on respawn and turns on cheats: a test profile;
  off for play
Directory = /home/rohan/WubarrkCODING/BetterContinents-TestMaps/fire-and-ice

[02 BetterContinents.Heightmap]
Heightmap Amount = 2.0       # from generator/mapscale.py; MUST match the maps

[04 BetterContinents.Forest]
Forest Amount = 0.50         # neutral
Forest Factor Overrides All Trees = true
Forestmap Multiply = 0
Forestmap Add = 1            # so that the forest map is a plain additive density

[07 BetterContinents.Misc]
SelectedPreset = From Config # the literal sentinel; anything else disables the mod for the
  new world

[08 BetterContinents.AltBiomes]
Overrides = Wolf Mountain: enabled=false   # painted on the west massif; no random one
  elsewhere
```

Everything not listed stays at the mod's default: `Heightmap Override All = true` (vanilla's rivers and height curves are bypassed), `Sea Level = 0.5` (adjustment 0), `Heatmap Scale = 10`, alt-biome `Mode = Random` and `Grid = WorldEdge`. The `Directory` setting auto-loads every correctly named file in the folder, `altbiomemap.png` and `lavamap.png` included. Since 0.9.4 it also reads an `export.cfg` there, whose settings then win over this file (chapter 9); `fire-and-ice/` has none, so this file is the whole story.

The world then carries a baked copy of all of it. To see what a world was built from, the fork's `tools/altbiome-harness` has an `inspect` command that reads a COPY of a world's `BetterContinents` settings file; never point it at the live one.

The Gale profile

`BC-Tester`, at `~/.local/share/com.kesomannen.gale/valheim/profiles/BC-Tester/`: the BepInEx pack 5.4.2350, Better Continents 0.9.x in `BepInEx/plugins/Wubarrk-BetterContinents/`, plus Upgrade World and Server devcommands (optional; the mod's console integration with Server devcommands 1.51 or later gives autocomplete). Create profiles in Gale itself (it tracks them in a database; a folder made by hand does not show up), install the BepInEx pack into the profile, then stage.

Biome precision and the Export group (0.9.0, not written by the staging script)

Two more groups exist since Better Continents 0.9.0. `stage-into-gale.sh` sets neither, so ProximaMaxi runs at their defaults; chapter 9 covers the console side of both.

```
[03 BetterContinents.Biomemap]
Biome precision = 0          # 0-5; 0 = vanilla (a zone's 4 corners); untouched by ProximaMaxi
  so far
[09 BetterContinents.Export]
```

```
Hud = false                  # F9 status box / F7 window; on turns them on for this game (and
  its clients)
Allow Export = true          # whether a connected player may bc_export this world at all
Hud Hotkey = F9
Window Hotkey = F7
Default Size = 4096          # bc_export's size when none is given, 128-8192
Default Heightmap Amount = 2 # what bc_export encodes heights for by default - matches
  mapscale.py
Default Sea Level = 0.5
```

Unlike every group above, `[09 BetterContinents.Export]` applies while the game runs, and on a server its `Hud`/`Allow Export` reach connected clients too. Better Continents 0.9.0 also re-reads the whole config file whenever it changes on disk, not only once at launch (chapter 9): edit `BetterContinents.cfg` while the game is running, and a **New** world you then create in that same session reads the new values, no relaunch needed. This narrows the premise behind trap 3 (chapter 7) and behind `stage-into-gale.sh`'s own refusal to run while Valheim is running - both were written for a version that read the file only at launch. Whether the script's refusal is still needed for some other reason (such as replacing a Gale-hardlinked DLL safely) was not checked while writing this chapter; treat it as still in force and quit before staging until someone has.

The world

`~/.config/unity3d/IronGate/Valheim/worlds_local/<name>/` is Valheim 1.0's folder-per-world layout, with `_main.N.fwl2` and `.db2` files and the mod's extension-less `BetterContinents` file beside them. Copy a world by copying the whole folder. Never rename or move a modded world through the game's own save menu unless the mod version is known to carry the sidecar through it (0.8.x does).

Appendix C

The game's own numbers

Generated from the extracted game data (`locations/locations_1.0.15.json`, `altbiomes/altbiomes_1.0.15_full.json`) by `make_c_data.py`. Regenerate it after re-extracting for a new game build.

Every enabled location, in the game's placement order

Prioritized locations first (the `_ZoneSystem` prefab's own list, then the add-on lists), then the rest. Area is the biome-area rule (Median, Edge or Any); altitude is metres above the water; delta is the terrain delta window; centre is the distance window from the world centre; sim is the minimum distance from similar locations, with the group's name where the rule is by group; veg is the vegetation (lava mask) window where it is not 0..1. Names are the prefab asset names, which the game log uses.

Location	Biome	Area	N	Altitude	Delta	Centre	Sim	Notes
StartTemple	Meadows	Median	1	3..1000	0..3			prioritized, centre first, forest 1..5
Eikthyrnir	Meadows	Median	3	1..1000	0..3			prioritized
GoblinKing	Plains	Median	4	1..1000	0..4		3000	prioritized
GDKing	BlackForest	Median	4	1..1000	0..5		3000	prioritized
Bonemass	Swamp	Median	5	0..2	0..4		3000	prioritized
SunkenCrypt4	Swamp	Median	175	0..2	0..4		64 (SunkenCrypt)	prioritized
Vendor_BlackForest	BlackForest	Median	10	1..1000	0..2		512	prioritized, unique
Dragonqueen	Mountain	Median	3	150..500	0..4		3000	prioritized
DN_Bossroom	DeepNorth	Any	3	80..5000	0..6		1024 (dn_boss)	prioritized
FaderLocation	AshLands	Median	3	1..300	0..40		2048 (FaderBoss)	prioritized, veg 0..0.1
PlaceofMystery1	AshLands	Median	1	0..1000	0..5		2048 (PlaceofMystery)	prioritized, unique, veg 0..0.1
PlaceofMystery2	AshLands	Median	1	0..1000	0..5		2048 (PlaceofMystery)	prioritized, unique, veg 0..0.1
PlaceofMystery3	AshLands	Median	1	0..1000	0..5		2048 (PlaceofMystery)	prioritized, unique, veg 0..0.1
CharredFortress	AshLands	Any	20	20..1000	0..10		256 (FaderBoss)	prioritized, veg 0..0.1, not near group zigg
BogWitch_Camp	Swamp	Median	10	1..1000	0..2		1000	prioritized, unique
Mistlands_DvergrTownEntrance1	Mistlands	Median	120	12..1000	5..40		256 (DvergrDungeon)	prioritized
Mistlands_DvergrTownEntrance2	Mistlands	Median	120	1..20	0..40		256 (DvergrDungeon)	prioritized
Mistlands_DvergrBossEntrance1	Mistlands	Median	5	1..20	0..40		2048 (DvergrBoss)	prioritized
AncientUpgradeStation	Mountain	Median	10	0..5000	0..40	500+	400 (AncientUpgradeStation)	prioritized, unique
Hildir_cave	Mountain	Median	3	200..5000	0..40		2000	prioritized
Hildir_camp	Meadows	Any	10	1..1000	0..2		1000	prioritized, unique
Hildir_crypt	BlackForest	Median	3	1..1000	0..2		3000	prioritized
Hildir_plainsfortress	Plains	Median	3	8..1000	0..4		3000	prioritized

GPL-3.0 (Wubarrk, 2026-09-24); see the License page

Location	Biome	Area	N	Altitude	Delta	Centre	Sim	Notes
StoneCircle	Meadows	Any	25	1..1000	0..3		200	
Greydwarf_camp1	BlackForest	Median	300	1..1000	0..3		128	
Runestone_Greydwarfs	BlackForest	Any	25	1..1000	0..3		128 (Runestones)	
Grave1	Swamp	Median	200	0.5..1000	0..3			
SwampRuin1	Swamp	Median	30	-0.5..1000	0..3		256 (SwampRuin)	
SwampRuin2	Swamp	Median	30	-0.5..1000	0..3		256 (SwampRuin)	
FireHole	Swamp	Median	75	-1..1000	0..5		16 (FireHole)	
Runestone_Draugr	Swamp	Any	50	0.5..1000	0..3		128 (Runestones)	
Crypt2	BlackForest	Any	200	1..1000	0..2		128	
Ruin1	BlackForest	Any	200	1..1000	0..3			
Ruin2	BlackForest	Any	200	1..1000	0..3			
StoneHouse1_heath	Plains	Any	0	1..1000	0..3			
StoneHouse2_heath	Plains	Any	0	1..1000	0..3			
StoneHouse5_heath	Plains	Any	0	1..1000	0..3			
StoneHouse3	BlackForest	Any	200	1..1000	0..3			
StoneHouse4	BlackForest	Any	200	1..1000	0..3			
Ruin3	Plains	Any	50	1..1000	0..3		512 (Goblintower)	
GoblinCamp2	Plains	Any	200	2..1000	0..3		250	
StoneTower1	Plains	Any	50	1..1000	0..3		512 (Goblintower)	
StoneTower3	Plains	Any	50	1..1000	0..3		512 (Goblintower)	
StoneHenge1	Plains	Any	5	5..1000	0..3		1000 (Stonehenge)	
StoneHenge2	Plains	Any	5	5..1000	0..3		1000 (Stonehenge)	
StoneHenge3	Plains	Any	5	5..1000	0..3		1000 (Stonehenge)	
StoneHenge4	Plains	Any	5	5..1000	0..3		1000 (Stonehenge)	
StoneHenge5	Plains	Any	20	2..1000	0..3		500 (Stonehenge)	
StoneHenge6	Plains	Any	20	2..1000	0..3		500 (Stonehenge)	
WoodHouse1	Meadows	Any	20	1..1000	0..4			
WoodHouse2	Meadows	Any	20	1..1000	0..4			
WoodHouse3	Meadows	Any	20	1..1000	0..4			
WoodHouse4	Meadows	Any	20	1..1000	0..4			
WoodHouse5	Meadows	Any	20	1..1000	0..4			
WoodHouse6	Meadows	Any	20	1..1000	0..4			
WoodHouse7	Meadows	Any	20	1..1000	0..4			
WoodHouse8	Meadows	Any	20	1..1000	0..4			
WoodHouse9	Meadows	Any	20	1..1000	0..4			
WoodHouse10	Meadows	Any	20	1..1000	0..4			
WoodHouse11	Meadows	Any	20	1..1000	0..4			
WoodHouse12	Meadows	Any	20	1..1000	0..4			
WoodHouse13	Meadows	Any	20	1..1000	0..4			
WoodFarm1	Meadows	Any	10	1..1000	0..4		128 (woodvillage)	
WoodVillage1	Meadows	Any	15	1..1000	0..4		256 (woodvillage)	
TrollCave02	BlackForest	Median	200	3..1000	5..10		256	
Dolmen01	Meadows, BlackForest	Any	100	1..1000	0..2			
Dolmen02	Meadows, BlackForest	Any	100	1..1000	0..2			
Dolmen03	Meadows, BlackForest	Any	50	1..1000	0..2			
Crypt3	BlackForest	Any	200	3..1000	0..2		128	
Crypt4	BlackForest	Any	200	1..1000	0..2		128	
InfestedTree01	Swamp	Any	700	-1..1000	0..3			

Location	Biome	Area	N	Altitude	Delta	Centre	Sim	Notes
SwampHut1	Swamp	Any	50	-2..1000	0..3		128 (Swamphut)	
SwampHut2	Swamp	Any	50	2..1000	0..3		128 (Swamphut)	
SwampHut3	Swamp	Any	50	2..1000	0..3		128 (Swamphut)	
SwampHut4	Swamp	Any	50	-1..1000	0..3		128 (Swamphut)	
SwampHut5	Swamp	Any	25	-1..1000	0..3		128 (Swamphut)	
SwampWell1	Swamp	Any	25	-1..1000	0..3		1024	
StoneTowerRuins04	Mountain	Any	50	150..1000	6..40		128 (Mountainruin)	
StoneTowerRuins05	Mountain	Any	50	150..1000	6..40		128 (Mountainruin)	
StoneTowerRuins03	BlackForest	Any	80	2..1000	0..2		200 (Stonetowerruins)	
StoneTowerRuins07	BlackForest	Any	80	2..1000	0..2		200 (Stonetowerruins)	
StoneTowerRuins08	BlackForest	Any	80	2..1000	0..2		200 (Stonetowerruins)	
StoneTowerRuins09	BlackForest	Any	80	2..1000	0..2		200 (Stonetowerruins)	
StoneTowerRuins10	BlackForest	Any	80	2..1000	0..2		200 (Stonetowerruins)	
ShipSetting01	Meadows	Any	100	1..1000	0..3		128	
DrakeNest01	Mountain	Any	200	100..2000	0..3		100	
Waymarker01	Mountain	Any	50	100..1000	0..2			
Waymarker02	Mountain	Any	50	100..1000	0..2			
AbandonedLogCabin02	Mountain	Any	33	100..1000	0..3		128 (Abandonedcabin)	
AbandonedLogCabin03	Mountain	Any	33	100..1000	0..3		128 (Abandonedcabin)	
AbandonedLogCabin04	Mountain	Any	50	100..1000	0..3		128 (Abandonedcabin)	
MountainGrave01	Mountain	Any	100	100..1000	0..2		50	
DrakeLorestone	Mountain	Any	50	100..1000	0..3		128 (Runestones)	
MountainWell1	Mountain	Any	25	100..1000	0..3		256	
ShipWreck01	Swamp, BlackForest, Plains, Ocean	Any	25	-1..1	0..10		1024 (Shipwreck)	
ShipWreck02	Swamp, BlackForest, Plains, Ocean	Any	25	-1..1	0..10		1024 (Shipwreck)	
ShipWreck03	Swamp, BlackForest, Plains, Ocean	Any	25	-1..1	0..10		1024 (Shipwreck)	
ShipWreck04	Swamp, BlackForest, Plains, Ocean	Any	25	-1..1	0..10		1024 (Shipwreck)	
Runestone_Meadows	Meadows	Any	100	1..1000	0..3		128 (Runestones)	
Runestone_Boars	Meadows	Any	50	1..1000	0..3		128 (Runestones)	
Runestone_Swamps	Swamp	Any	100	0..1000	0..3		128 (Runestones)	
Runestone_Mountains	Mountain	Any	100	100..1000	0..3		128 (Runestones)	
Runestone_BlackForest	BlackForest	Any	50	1..1000	0..3		128 (Runestones)	
Runestone_Plains	Plains	Any	100	1..1000	0..3		128 (Runestones)	
CombatRuin01	Meadows	Any	5	5..1000	0..1.5	1500+	2000	forest 0.5..1
BigRockClearing	BlackForest	Median	10	5..1000	0..2	1000+	2000	unique, forest 0.8..1
TheHole01	DeepNorth	Median	40	2..1000	0..3		256	not near group thehole
NorthVillage	DeepNorth	Median	135	2..1000	0..3		70, at most 85	not near group thehole

Location	Biome	Area	N	Altitude	Delta	Centre	Sim	Notes
MorkBorg	DeepNorth	Any	40	30..10000	0..300		275 (morkborg)	not near group 0
NorthMemorialPlace	DeepNorth	Median	15	2..1000	0..3		400 (memorialplace)	not near group 0
FrozenShip01_DN	DeepNorth	Any	50	-15..-5	0..100	8000..9750	64 (FrozenShip)	snaps to water
FrozenShip02_DN	DeepNorth	Any	50	-15..-5	0..100	8000..9750	64 (FrozenShip)	snaps to water
FrozenShip03_DN	DeepNorth	Any	50	-15..-5	0..100	8000..9750	64 (FrozenShip)	snaps to water
IcePond1	DeepNorth	Median	40	10..200	0..1.5		128 (icepond)	forest 0..1
DN_hut01	DeepNorth	Median	40	40..1000	0..4		100 (northvillage)	not near group 0
ShipSetting02	DeepNorth	Any	100	2..1000	0..2		64 (shipsetting)	
ShipSetting03	DeepNorth	Any	50	2..1000	0..2		64 (shipsetting)	
LumberCamp	DeepNorth	Median	50	4..1000	0..3		100 (thehole)	forest 0..1
ShipWreck01_DN	DeepNorth	Any	170	-0.5..0.5	0..2			
ShipWreck02_DN	DeepNorth	Any	120	-0.5..1	0..2			
Runestone_DeepNorth	DeepNorth	Any	70	0..1000	0..10		128 (Runestones)	veg 0..0.1
DN_gammeltrollFrac01	DeepNorth	Any	30	5..1000	0..10			
DN_gammeltrollFrac02	DeepNorth	Any	30	10..1000	0..10			
BearCave	BlackForest	Median	50	5..1000	5..10		256	
FortressRuins	AshLands	Any	100	0..1000	0..10		, at most 70	veg 0..0.8, not near group zigg
CharredRuins1	AshLands	Median	75	-5..1000	0..6		256 (zigg)	
Runestone_Ashlands	AshLands	Any	70	0..1000	0..10		128 (Runestones)	veg 0..0.1
LeviathanLava	AshLands	Any	100	10..1000	0..10			veg 0.6..1
SulfurArch	AshLands	Median	100	10..1000	0..6		40	veg 0..0.8
CharredStone_Spawner	AshLands	Any	300	0..1000	0..10		100	veg 0..0.2
VoltureNest	AshLands	Any	350	0..1000	0..5		100	veg 0..0.1
CharredTowerRuins1	AshLands	Any	30	0..1000	0..4		100 (towerruins)	veg 0..0.5
CharredTowerRuins1_dvergr	AshLands	Any	30	0..1000	0..4		100 (towerruins)	veg 0..0.1
CharredTowerRuins2	AshLands	Any	40	0..1000	0..4			veg 0..0.3
CharredTowerRuins3	AshLands	Any	30	0..1000	0..4		500	veg 0..0.3
AshlandRuins	AshLands	Median	100	0..1000	0..4		16	veg 0..0.2
MorgenHole1	AshLands	Median	40	0..1000	0..10		300 (MorgenHole)	veg 0..0.3
MorgenHole2	AshLands	Median	40	0..1000	0..10		200 (MorgenHole)	veg 0..0.3
MorgenHole3	AshLands	Median	40	0..1000	0..10		200 (MorgenHole)	veg 0..0.3
CharredRuins2	AshLands	Median	100	-5..1000	0..30		128	
CharredRuins3	AshLands	Median	100	-5..1000	0..30		128	
CharredRuins4	AshLands	Median	100	-5..1000	0..30		128	
Mistlands_GuardTower1_new	Mistlands	Median	75	2..1000	0..10		128 (Dvergr)	
Mistlands_GuardTower1_ruined_new	Mistlands	Median	80	2..1000	0..10		128 (Dvergr)	
Mistlands_GuardTower1_ruined_new2	Mistlands	Median	20	2..1000	0..10		128 (Dvergr)	
Mistlands_GuardTower2_new	Mistlands	Median	75	4..1000	0..10		128 (Dvergr)	
Mistlands_GuardTower3_new	Mistlands	Median	50	12..1000	0..10		128 (Dvergr)	
Mistlands_GuardTower3_ruined_new	Mistlands	Median	50	11..1000	0..10		128 (Dvergr)	
Mistlands_Lighthouse1_new	Mistlands	Median	100	1..20	0..40		128 (Dvergr)	
Mistlands_Excavation1	Mistlands	Median	40	4..100	0..4		128 (Excavation)	
Mistlands_Excavation2	Mistlands	Median	40	4..100	0..4		128 (Excavation)	
Mistlands_Excavation3	Mistlands	Median	40	4..100	0..5		96 (Excavation)	

Location	Biome	Area	N	Altitude	Delta	Centre	Sim	Notes
Mistlands_Harbour1	Mistlands	Edge	100	-1..-0.25	2..20		64 (Harbour)	snaps to water
Mistlands_Viaduct1	Mistlands	Median	100	-1..15	2..40		128 (Harbour)	
Mistlands_Viaduct2	Mistlands	Median	150	-1..25	4..40		64 (Harbour)	
Mistlands_RockSpire1	Mistlands	Median	200	-10..1000	2..99		60	
Mistlands_Giant1	Mistlands	Median	250	-1..1000	0..3		180 (Giant)	
Mistlands_Giant2	Mistlands	Median	85	-1..1000	0..4		256 (Giant)	
Mistlands_RoadPost1	Mistlands	Any	500	1..1000	0..10			
Mistlands_StatueGroup1	Mistlands	Any	200	2..1000	0..10		32	
Mistlands_Statue1	Mistlands	Any	200	2..1000	0..10			
Mistlands_Statue2	Mistlands	Any	200	2..1000	0..10			
Mistlands_Swords1	Mistlands	Median	33	-1..1000	0..10		256 (GiantArmor)	
Mistlands_Swords2	Mistlands	Median	33	-1..1000	0..10		256 (GiantArmor)	
Mistlands_Swords3	Mistlands	Median	33	-1..1000	0..10		256 (GiantArmor)	
Runestone_Mistlands	Mistlands	Any	50	-1..1000	0..10		128 (Runestones)	
TarPit1	Plains	Median	100	5..60	0..1.5		128 (tarpit)	
TarPit2	Plains	Median	16	5..60	0..1.5		128 (tarpit)	
TarPit3	Plains	Median	100	5..60	0..1.5		128 (tarpit)	
MountainCave02	Mountain	Any	120	100..5000	0..40		200 (mountaincaves)	

170 location types. Disabled entries (the game ships some switched off) and the duplicate entries of the add-on lists are left out, as the game leaves them out.

The 32 alt biomes and their placement windows

In the game's list order (which is also the order the game places them in, per base biome). Count is the number of regions per world, minimum to maximum; the chance applies above the minimum; edge is the `EdgeCount` window [min, max); centre is the minimum distance from the world centre; the colour is Better Continents' default legend colour (`palettes/altbiomemap.txt`). Every alt biome wants `HeightAvg` 30 to 10,000 (absolute), and none sets a neighbour rule or world-position bounds.

Alt biome	Base biomes	Count	Chance	Edge	Centre	Incompatible with	Adds	Colour
Mushroom	Meadows, BlackForest	1 to 3	0.20	66 to 199	500			FF969C
Lantern	Meadows, Swamp, Mountain, BlackForest, Plains, DeepNorth, Mistlands	3 to 5	0.20	66 to 199	500			F2A300
Bones	Swamp, Plains, Mistlands	1 to 3	0.20	66 to 199	2000			F8E1B6
Menhir	Meadows, Plains	1 to 3	0.20	33 to 199	750	Birch Meadows		DDA776
Dark Meadows	Meadows	2 to 5	0.50	66 to 299	1900		1	2D4613
Peaceful Meadows	Meadows	1 to 2	0.20	99 to 299	1000	Bones, Dark Meadows		ADFFDA
Dandelion Meadows	Meadows	1 to 3	0.20	66 to 199	500			E5C54B
Raspberry Meadows	Meadows	1 to 3	0.20	66 to 199	500			C2185B
Smalltree Meadows	Meadows	1 to 3	0.20	33 to 199	500			6EC03D
Birch Meadows	Meadows	1 to 3	0.20	66 to 199	500			CCFFAC
Troll Black Forest	BlackForest	1 to 3	0.20	66 to 199	750			2C7B76

GPL-3.0 (Wubarrk, 2026-09-24); see the License page

Alt biome	Base biomes	Count	Chance	Edge	Centre	Incompatible with	Adds	Colour
Root Black Forest	BlackForest	1 to 4	0.20	99 to 299	2000	Rock Black Forest, Kalhygge Black Forest		6F5E32
Ruin Black Forest	BlackForest	1 to 5	0.20	33 to 199	500		4	9D6C4B
Rock Black Forest	BlackForest	1 to 5	0.20	33 to 199	750	Pinetree Black Forest, Root Black Forest		5A5264
Pinetree Black Forest	BlackForest	1 to 5	0.20	33 to 199	750	Rock Black Forest		004754
Blueberry Black Forest	BlackForest	1 to 5	0.20	33 to 199	500			4A5CB6
Hut Swamp	Swamp	1 to 3	0.20	33 to 199	1500	Bog Swamp	3	B1332D
Bog Swamp	Swamp	1 to 5	0.20	33 to 145	1500			6E2143
Bat Swamp	Swamp	1 to 3	0.20	33 to 145	2000			382B41
Abomination Swamp	Swamp	1 to 3	0.20	66 to 145	2000			95B17E
Wolf Mountain	Mountain	1 to 5	0.20	199 to 299	2000	Drake Mountain, Fortress Mountain		93C7BC
Drake Mountain	Mountain	1 to 3	0.20	199 to 299	2000	Wolf Mountain, Fortress Mountain		5AAAC2
Fortress Mountain	Mountain	1 to 2	0.20	199 to 299	2000	Wolf Mountain, Drake Mountain	1	B697AF
Lox Plains	Plains	0 to 1	0.20	99 to 299	2000	Goblin Plains, Death Plains		CC605F
Goblin Plains	Plains	0 to 2	0.20	66 to 199	2000	Lox Plains, Death Plains	4	FF7622
Death Plains	Plains	0 to 1	0.20	99 to 299	1000	Lox Plains, Goblin Plains	3	671E07
Kalhygge Black Forest	BlackForest	1 to 2	0.20	20 to 80	500	Root Black Forest		A9AE34
Rockless Mistlands	Mistlands	2 to 5	0.20	99 to 599	2000			D99FE8
Trees Mistlands	Mistlands	1 to 5	0.20	59 to 299	2000			472577
Swords Mistlands	Mistlands	1 to 3	0.20	59 to 199	2000			8A7FDB
Hare Mistlands	Mistlands	1 to 3	0.20	59 to 199	2000			D56EBB
BroodSwarm Mistlands	Mistlands	1 to 3	0.20	59 to 199	2000			8E2F8E

The biome legend

The ten names Better Continents 0.8.1 accepts, with the default colours (`biomemap.txt`):

Biome	Colour	Biome	Colour
None (let vanilla decide)	`000000`	Plains	`FFFF00`
Meadows	`00FF00`	Mistlands	`7F7F7F`
BlackForest	`007F00`	AshLands	`FF0000`
Swamp	`7F7F00`	DeepNorth	`00FFFF`
Mountain	`FFFFFF`	Ocean	`0000FF`

ProximaMaxi's alt-biome legend (version 13)

Nineteen colours, all from the default palette, 23 planted regions (Mushroom three times, Rockless Mistlands twice, Wolf Mountain on two summits that count as two regions, Lantern on the lobe and every Deep North isle, counted once):

```
# ProximaMaxi alt-biome legend (Better Continents 0.8.1): one line per planted colour.
# Black is unplanted land: the game's random placement handles it.
Fortress Mountain: B697AF
Drake Mountain: 5AAAC2
Wolf Mountain: 93C7BC
Goblin Plains: FF7622
Death Plains: 671E07
Lox Plains: CC605F
Hut Swamp: B1332D
Bat Swamp: 382B41
Abomination Swamp: 95B17E
Troll Black Forest: 2C7B76
Root Black Forest: 6F5E32
Ruin Black Forest: 9D6C4B
Dark Meadows: 2D4613
Peaceful Meadows: ADFFDA
Mushroom: FF969C
Swords Mistlands: 8A7FDB
BroodSwarm Mistlands: 8E2F8E
Rockless Mistlands: D99FE8
Lantern: F2A300
```

Appendix D

Glossary and sources

Glossary

- **Amount (Heightmap Amount)**: the config multiplier on the heightmap pixel; sets the waterline pixel (0.30 / Amount) and the ceiling ((Amount - 0.30) * 200 m). ProximaMaxi: 2.0.
- **Waterline**: pixel 0.30 / Amount; finalHeight 0.15; absolute height 30 m.
- **Altitude** (in the game's location rules): absolute height minus 30, i.e. metres above the water.
- **Zone**: the game's 64 m square; one location per zone.
- **Grid, points**: the game's 2048 x 2048 sampling of biome and height at 12 m, regenerated at every load.
- **Sector, region**: a 4-connected area of one biome on that grid; Ashlands, Deep North and Ocean are one each.
- **EdgeCount**: a region's border cell count; the alt-biome size window.
- **HeightAvg**: (lowest + highest) / 2 of a region's border heights, absolute; must reach 30.
- **DistanceFromCenter**: the distance of a region's mean border position from the origin.
- **Alt biome**: one of Valheim 1.0's 32 biome variants, placed per region; changes spawns, vegetation, locations, weather and music, not the biome's name.
- **Planting**: Better Continents 0.8.1 assigning alt biomes to painted regions (`altbiomemap.png`) before the game's random placement; planted regions count toward the maxima (the quota rule).
- **Prioritized**: a location placed first, with 60,000 attempts (bosses, vendors, key dungeons).
- **Unique**: a location of which only one is spawned; several candidates are placed and the rest are removed when the first is reached.
- **Group, minDistanceFromSimilar**: spacing rules between locations of one kind or group.
- **Biome area (Edge, Median, Everything)**: whether the zone centre must be inside the biome (Median: the eight points 64 m around it agree), on its edge, or either.
- **Terrain delta**: the highest minus the lowest of ten random heights within a location's exterior radius; a flatness window.
- **Vegetation window**: in the Ashlands, the lava mask value; the lava map.
- **FBM**: fractional Brownian motion, summed octaves of noise; the gain is the amplitude ratio per octave.
- **Domain warp**: sampling a field at displaced coordinates to bend its outlines.
- **Leash**: capping a warped landmass field against its unwarped shape plus slack.
- **Priority-Flood**: the drainage algorithm that fills pits and gives every cell a downhill route to the sea.
- **ROLL, ROLL_K**: per-biome RMS relief in metres, and the 1 / std factor that makes an FBM's output unit-RMS.
- **Staging**: writing the mod and the config into the Gale profile (`stage-into-gale.sh`).
- **Baking**: the mod copying the maps and settings into the world at creation.
- **Gale**: the mod manager; profiles hold BepInEx and the plugins; DLLs are hardlinked from its cache.
- **BC-Tester**: the Gale profile used for these builds.

- **ProximaMaxi, Fire and Ice**: the map this document was learned on; the realm of Wubarrk's Proxima Mod Pack. Fire and Ice was its first name, and the folder keeps it.

Sources, and how to check a claim

Every number in this document has one of these origins:

1. **The mod's code**, Better Continents 0.9.x (`/home/rohan/WubarrkCODING/BetterContinents10`): the height formula (`BetterContinents.WorldGeneratorPatch.cs`: `GetBaseHeightV3`, `GetBiomeHeightWithHeight`), the forest mapping (`ApplyForest`), heat (`IsAshlands`), the alt-biome map (`ImageMapAltBiome.cs`, `BetterContinents.AltBiomePlanting.cs`, documented in `ALTBIOMES.md`), the save layout (`MIGRATE-1.0.md`).
2. **The game's code**, Valheim 1.0, decompiled in `libs-Tools/1.0/DECOMPILED/assembly_valheim.decompiled.cs` (the client; `_SERVER` for the server): `ZoneSystem.GenerateLocationsTimeSliced` (the attempt counts, the six points, the error categories, the unique rule), `AltBiomeWorldData.VerifyBiomeData` (`RemoveCache`, `GenerateBiomePoints`, `GenerateSectors` on every load), `BiomeSector.CanAddModifier`, `WorldGenerator.GetBiomeSector`, `Player.UpdateBiome` (the GetBiome warning), `EnvMan` (the Deep North weather bug). The per-type files in `libs-Tools/Decompiled_<version>/assembly_valheim/` are quicker to read; check the folder is the current build's; `altbiomes/RESEARCH-vanilla-behaviour.md` cites them by file and line.
3. **The game's data**, extracted read-only from the bundle `d59cfac` (Unity 6000.0.75f1) by the two extractors: `_ZoneSystem.prefab` and the `_LocationList_*` prefabs (340 location entries, 170 enabled distinct types), `AltBiomes_Erik.prefab` (32 alt biomes).
4. **Real builds**: `Proxima` (version 11, Amount 1.0), `ProximaEra01` (version 12), `ProxEra01` (version 13), each with its game log replayed by the checker.
5. **The human's words**, quoted where they set a rule.

To check a claim about the game: find the method in source 2, read it, and note the line. To check a claim about the mod: the fork's source, or `ALTBIOMES.md` section 2 for anything alt-biome. To check a claim about a map: run the checker. Write what you learn into the workspace's notes (`HANDOFF.md`, a `RESEARCH-*.md`, or a `libs-Tools/*-FACTS.md` with a dated banner and file:line citations), so that the next agent does not re-derive it.

The map's versions, as a study

Version	What changed	What the human said
01	the layout, drawn as ellipses; islands parked in a row	"we want a realistic feeling map"
02 to 05	natural fronts, ragged fronts, island groups, the Ashlands arm pivoted 45 degrees south-west (it came within 1.2 km of spawn), a Plains and Mistlands landmass south of spawn	
06	two archipelagos, satellites, skerries, windward cliffs, alt-biome homes	"i like it 06", "refine and work off of it"
07	refined alt-biome homes (the random-placement tuning)	
08	alt biomes planted with colours (0.8.1)	
09	rivers as drainage, lakes	"the river looks too sharp and unnatural"
10	Deep North isles with Lantern, Wolf Mountain moved, the starter river redrawn	"make various alt deep north islands, move wolf mountain"
11	Mushroom three times, streams wetter, natural-basin lakes	"we def want some mushroom meadows"; built as `Proxima`: the cavern 0/3

Version	What changed	What the human said
12	Amount 2.0, icefields, the Ashlands fattened with burning isles and lava, rim chains, Mistlands lowlands, biome-labelled shallows, the location checker	"get everything right this time"; built as `ProximaEra01`: "reads too flat"
13	rolling relief, the short beach, the low bog, sunk Mistlands valleys	built as `ProxEra01`: "looks good on my end"

Appendix E

GNU General Public License, version 3

The GNU General Public License, version 3, reproduced verbatim from LICENSE.

```
                    GNU GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU General Public License is a free, copyleft license for
software and other kinds of works.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
the GNU General Public License is intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.  We, the Free Software Foundation, use the
GNU General Public License for most of our software; it applies also to
any other work released this way by its authors.  You can apply it to
your programs, too.

  When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

  To protect your rights, we need to prevent others from denying you
these rights or asking you to surrender the rights.  Therefore, you have
certain responsibilities if you distribute copies of the software, or if
you modify it: responsibilities to respect the freedom of others.

  For example, if you distribute copies of such a program, whether
gratis or for a fee, you must pass on to the recipients the same
freedoms that you received.  You must make sure that they, too, receive
or can get the source code.  And you must show them these terms so they
know their rights.

  Developers that use the GNU GPL protect your rights with two steps:
(1) assert copyright on the software, and (2) offer you this License
giving you legal permission to copy, distribute and/or modify it.

  For the developers' and authors' protection, the GPL clearly explains
that there is no warranty for this free software.  For both users' and
authors' sake, the GPL requires that modified versions be marked as
changed, so that their problems will not be attributed erroneously to
authors of previous versions.

  Some devices are designed to deny users access to install or run
modified versions of the software inside them, although the manufacturer
can do so.  This is fundamentally incompatible with the aim of
protecting users' freedom to change the software.  The systematic
pattern of such abuse occurs in the area of products for individuals to
use, which is precisely where it is most unacceptable.  Therefore, we
have designed this version of the GPL to prohibit the practice for those
products.  If such problems arise substantially in other domains, we
stand ready to extend this provision to those domains in future versions
of the GPL, as needed to protect the freedom of users.

  Finally, every program is threatened constantly by software patents.
States should not allow patents to restrict development and use of
software on general-purpose computers, but in those that do, we wish to
avoid the special danger that patents applied to a free program could
make it effectively proprietary.  To prevent this, the GPL assures that
patents cannot be used to render the program non-free.
```

```
  The precise terms and conditions for copying, distribution and
modification follow.

                       TERMS AND CONDITIONS

  0. Definitions.

  "This License" refers to version 3 of the GNU General Public License.

  "Copyright" also means copyright-like laws that apply to other kinds of
works, such as semiconductor masks.

  "The Program" refers to any copyrightable work licensed under this
License.  Each licensee is addressed as "you".  "Licensees" and
"recipients" may be individuals or organizations.

  To "modify" a work means to copy from or adapt all or part of the work
in a fashion requiring copyright permission, other than the making of an
exact copy.  The resulting work is called a "modified version" of the
earlier work or a work "based on" the earlier work.

  A "covered work" means either the unmodified Program or a work based
on the Program.

  To "propagate" a work means to do anything with it that, without
permission, would make you directly or secondarily liable for
infringement under applicable copyright law, except executing it on a
computer or modifying a private copy.  Propagation includes copying,
distribution (with or without modification), making available to the
public, and in some countries other activities as well.

  To "convey" a work means any kind of propagation that enables other
parties to make or receive copies.  Mere interaction with a user through
a computer network, with no transfer of a copy, is not conveying.

  An interactive user interface displays "Appropriate Legal Notices"
to the extent that it includes a convenient and prominently visible
feature that (1) displays an appropriate copyright notice, and (2)
tells the user that there is no warranty for the work (except to the
extent that warranties are provided), that licensees may convey the
work under this License, and how to view a copy of this License.  If
the interface presents a list of user commands or options, such as a
menu, a prominent item in the list meets this criterion.

  1. Source Code.

  The "source code" for a work means the preferred form of the work
for making modifications to it.  "Object code" means any non-source
form of a work.

  A "Standard Interface" means an interface that either is an official
standard defined by a recognized standards body, or, in the case of
interfaces specified for a particular programming language, one that
is widely used among developers working in that language.

  The "System Libraries" of an executable work include anything, other
than the work as a whole, that (a) is included in the normal form of
packaging a Major Component, but which is not part of that Major
Component, and (b) serves only to enable use of the work with that
Major Component, or to implement a Standard Interface for which an
implementation is available to the public in source code form.  A
"Major Component", in this context, means a major essential component
(kernel, window system, and so on) of the specific operating system
(if any) on which the executable work runs, or a compiler used to
produce the work, or an object code interpreter used to run it.

  The "Corresponding Source" for a work in object code form means all
the source code needed to generate, install, and (for an executable
work) run the object code and to modify the work, including scripts to
control those activities.  However, it does not include the work's
System Libraries, or general-purpose tools or generally available free
programs which are used unmodified in performing those activities but
which are not part of the work.  For example, Corresponding Source
includes interface definition files associated with source files for
the work, and the source code for shared libraries and dynamically
linked subprograms that the work is specifically designed to require,
```

```
such as by intimate data communication or control flow between those
subprograms and other parts of the work.

  The Corresponding Source need not include anything that users
can regenerate automatically from other parts of the Corresponding
Source.

  The Corresponding Source for a work in source code form is that
same work.

  2. Basic Permissions.

  All rights granted under this License are granted for the term of
copyright on the Program, and are irrevocable provided the stated
conditions are met.  This License explicitly affirms your unlimited
permission to run the unmodified Program.  The output from running a
covered work is covered by this License only if the output, given its
content, constitutes a covered work.  This License acknowledges your
rights of fair use or other equivalent, as provided by copyright law.

  You may make, run and propagate covered works that you do not
convey, without conditions so long as your license otherwise remains
in force.  You may convey covered works to others for the sole purpose
of having them make modifications exclusively for you, or provide you
with facilities for running those works, provided that you comply with
the terms of this License in conveying all material for which you do
not control copyright.  Those thus making or running the covered works
for you must do so exclusively on your behalf, under your direction
and control, on terms that prohibit them from making any copies of
your copyrighted material outside their relationship with you.

  Conveying under any other circumstances is permitted solely under
the conditions stated below.  Sublicensing is not allowed; section 10
makes it unnecessary.

  3. Protecting Users' Legal Rights From Anti-Circumvention Law.

  No covered work shall be deemed part of an effective technological
measure under any applicable law fulfilling obligations under article
11 of the WIPO copyright treaty adopted on 20 December 1996, or
similar laws prohibiting or restricting circumvention of such
measures.

  When you convey a covered work, you waive any legal power to forbid
circumvention of technological measures to the extent such circumvention
is effected by exercising rights under this License with respect to
the covered work, and you disclaim any intention to limit operation or
modification of the work as a means of enforcing, against the work's
users, your or third parties' legal rights to forbid circumvention of
technological measures.

  4. Conveying Verbatim Copies.

  You may convey verbatim copies of the Program's source code as you
receive it, in any medium, provided that you conspicuously and
appropriately publish on each copy an appropriate copyright notice;
keep intact all notices stating that this License and any
non-permissive terms added in accord with section 7 apply to the code;
keep intact all notices of the absence of any warranty; and give all
recipients a copy of this License along with the Program.

  You may charge any price or no price for each copy that you convey,
and you may offer support or warranty protection for a fee.

  5. Conveying Modified Source Versions.

  You may convey a work based on the Program, or the modifications to
produce it from the Program, in the form of source code under the
terms of section 4, provided that you also meet all of these conditions:

    a) The work must carry prominent notices stating that you modified
    it, and giving a relevant date.

    b) The work must carry prominent notices stating that it is
    released under this License and any conditions added under section
    7.  This requirement modifies the requirement in section 4 to
```

"keep intact all notices".

c) You must license the entire work, as a whole, under this
License to anyone who comes into possession of a copy. This
License will therefore apply, along with any applicable section 7
additional terms, to the whole of the work, and all its parts,
regardless of how they are packaged. This License gives no
permission to license the work in any other way, but it does not
invalidate such permission if you have separately received it.

d) If the work has interactive user interfaces, each must display
Appropriate Legal Notices; however, if the Program has interactive
interfaces that do not display Appropriate Legal Notices, your
work need not make them do so.

A compilation of a covered work with other separate and independent
works, which are not by their nature extensions of the covered work,
and which are not combined with it such as to form a larger program,
in or on a volume of a storage or distribution medium, is called an
"aggregate" if the compilation and its resulting copyright are not
used to limit the access or legal rights of the compilation's users
beyond what the individual works permit. Inclusion of a covered work
in an aggregate does not cause this License to apply to the other
parts of the aggregate.

6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms
of sections 4 and 5, provided that you also convey the
machine-readable Corresponding Source under the terms of this License,
in one of these ways:

a) Convey the object code in, or embodied in, a physical product
(including a physical distribution medium), accompanied by the
Corresponding Source fixed on a durable physical medium
customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product
(including a physical distribution medium), accompanied by a
written offer, valid for at least three years and valid for as
long as you offer spare parts or customer support for that product
model, to give anyone who possesses the object code either (1) a
copy of the Corresponding Source for all the software in the
product that is covered by this License, on a durable physical
medium customarily used for software interchange, for a price no
more than your reasonable cost of physically performing this
conveying of source, or (2) access to copy the
Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the
written offer to provide the Corresponding Source. This
alternative is allowed only occasionally and noncommercially, and
only if you received the object code with such an offer, in accord
with subsection 6b.

d) Convey the object code by offering access from a designated
place (gratis or for a charge), and offer equivalent access to the
Corresponding Source in the same way through the same place at no
further charge. You need not require recipients to copy the
Corresponding Source along with the object code. If the place to
copy the object code is a network server, the Corresponding Source
may be on a different server (operated by you or a third party)
that supports equivalent copying facilities, provided you maintain
clear directions next to the object code saying where to find the
Corresponding Source. Regardless of what server hosts the
Corresponding Source, you remain obligated to ensure that it is
available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided
you inform other peers where the object code and Corresponding
Source of the work are being offered to the general public at no
charge under subsection 6d.

A separable portion of the object code, whose source code is excluded
from the Corresponding Source as a System Library, need not be
included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or

b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or

c) Prohibiting misrepresentation of the origin of that material, or

```
    requiring that modified versions of such material be marked in
    reasonable ways as different from the original version; or

    d) Limiting the use for publicity purposes of names of licensors or
    authors of the material; or

    e) Declining to grant rights under trademark law for use of some
    trade names, trademarks, or service marks; or

    f) Requiring indemnification of licensors and authors of that
    material by anyone who conveys the material (or modified versions of
    it) with contractual assumptions of liability to the recipient, for
    any liability that these contractual assumptions directly impose on
    those licensors and authors.

  All other non-permissive additional terms are considered "further
restrictions" within the meaning of section 10.  If the Program as you
received it, or any part of it, contains a notice stating that it is
governed by this License along with a term that is a further
restriction, you may remove that term.  If a license document contains
a further restriction but permits relicensing or conveying under this
License, you may add to a covered work material governed by the terms
of that license document, provided that the further restriction does
not survive such relicensing or conveying.

  If you add terms to a covered work in accord with this section, you
must place, in the relevant source files, a statement of the
additional terms that apply to those files, or a notice indicating
where to find the applicable terms.

  Additional terms, permissive or non-permissive, may be stated in the
form of a separately written license, or stated as exceptions;
the above requirements apply either way.

  8. Termination.

  You may not propagate or modify a covered work except as expressly
provided under this License.  Any attempt otherwise to propagate or
modify it is void, and will automatically terminate your rights under
this License (including any patent licenses granted under the third
paragraph of section 11).

  However, if you cease all violation of this License, then your
license from a particular copyright holder is reinstated (a)
provisionally, unless and until the copyright holder explicitly and
finally terminates your license, and (b) permanently, if the copyright
holder fails to notify you of the violation by some reasonable means
prior to 60 days after the cessation.

  Moreover, your license from a particular copyright holder is
reinstated permanently if the copyright holder notifies you of the
violation by some reasonable means, this is the first time you have
received notice of violation of this License (for any work) from that
copyright holder, and you cure the violation prior to 30 days after
your receipt of the notice.

  Termination of your rights under this section does not terminate the
licenses of parties who have received copies or rights from you under
this License.  If your rights have been terminated and not permanently
reinstated, you do not qualify to receive new licenses for the same
material under section 10.

  9. Acceptance Not Required for Having Copies.

  You are not required to accept this License in order to receive or
run a copy of the Program.  Ancillary propagation of a covered work
occurring solely as a consequence of using peer-to-peer transmission
to receive a copy likewise does not require acceptance.  However,
nothing other than this License grants you permission to propagate or
modify any covered work.  These actions infringe copyright if you do
not accept this License.  Therefore, by modifying or propagating a
covered work, you indicate your acceptance of this License to do so.

  10. Automatic Licensing of Downstream Recipients.

  Each time you convey a covered work, the recipient automatically
```

receives a license from the original licensors, to run, modify and
propagate that work, subject to this License. You are not responsible
for enforcing compliance by third parties with this License.

  An "entity transaction" is a transaction transferring control of an
organization, or substantially all assets of one, or subdividing an
organization, or merging organizations. If propagation of a covered
work results from an entity transaction, each party to that
transaction who receives a copy of the work also receives whatever
licenses to the work the party's predecessor in interest had or could
give under the previous paragraph, plus a right to possession of the
Corresponding Source of the work from the predecessor in interest, if
the predecessor has it or can get it with reasonable efforts.

  You may not impose any further restrictions on the exercise of the
rights granted or affirmed under this License. For example, you may
not impose a license fee, royalty, or other charge for exercise of
rights granted under this License, and you may not initiate litigation
(including a cross-claim or counterclaim in a lawsuit) alleging that
any patent claim is infringed by making, using, selling, offering for
sale, or importing the Program or any portion of it.

  11. Patents.

  A "contributor" is a copyright holder who authorizes use under this
License of the Program or a work on which the Program is based. The
work thus licensed is called the contributor's "contributor version".

  A contributor's "essential patent claims" are all patent claims
owned or controlled by the contributor, whether already acquired or
hereafter acquired, that would be infringed by some manner, permitted
by this License, of making, using, or selling its contributor version,
but do not include claims that would be infringed only as a
consequence of further modification of the contributor version. For
purposes of this definition, "control" includes the right to grant
patent sublicenses in a manner consistent with the requirements of
this License.

  Each contributor grants you a non-exclusive, worldwide, royalty-free
patent license under the contributor's essential patent claims, to
make, use, sell, offer for sale, import and otherwise run, modify and
propagate the contents of its contributor version.

  In the following three paragraphs, a "patent license" is any express
agreement or commitment, however denominated, not to enforce a patent
(such as an express permission to practice a patent or covenant not to
sue for patent infringement). To "grant" such a patent license to a
party means to make such an agreement or commitment not to enforce a
patent against the party.

  If you convey a covered work, knowingly relying on a patent license,
and the Corresponding Source of the work is not available for anyone
to copy, free of charge and under the terms of this License, through a
publicly available network server or other readily accessible means,
then you must either (1) cause the Corresponding Source to be so
available, or (2) arrange to deprive yourself of the benefit of the
patent license for this particular work, or (3) arrange, in a manner
consistent with the requirements of this License, to extend the patent
license to downstream recipients. "Knowingly relying" means you have
actual knowledge that, but for the patent license, your conveying the
covered work in a country, or your recipient's use of the covered work
in a country, would infringe one or more identifiable patents in that
country that you have reason to believe are valid.

  If, pursuant to or in connection with a single transaction or
arrangement, you convey, or propagate by procuring conveyance of, a
covered work, and grant a patent license to some of the parties
receiving the covered work authorizing them to use, propagate, modify
or convey a specific copy of the covered work, then the patent license
you grant is automatically extended to all recipients of the covered
work and works based on it.

  A patent license is "discriminatory" if it does not include within
the scope of its coverage, prohibits the exercise of, or is
conditioned on the non-exercise of one or more of the rights that are
specifically granted under this License. You may not convey a covered

work if you are a party to an arrangement with a third party that is
in the business of distributing software, under which you make payment
to the third party based on the extent of your activity of conveying
the work, and under which the third party grants, to any of the
parties who would receive the covered work from you, a discriminatory
patent license (a) in connection with copies of the covered work
conveyed by you (or copies made from those copies), or (b) primarily
for and in connection with specific products or compilations that
contain the covered work, unless you entered into that arrangement,
or that patent license was granted, prior to 28 March 2007.

  Nothing in this License shall be construed as excluding or limiting
any implied license or other defenses to infringement that may
otherwise be available to you under applicable patent law.

  12. No Surrender of Others' Freedom.

  If conditions are imposed on you (whether by court order, agreement or
otherwise) that contradict the conditions of this License, they do not
excuse you from the conditions of this License. If you cannot convey a
covered work so as to satisfy simultaneously your obligations under this
License and any other pertinent obligations, then as a consequence you may
not convey it at all. For example, if you agree to terms that obligate you
to collect a royalty for further conveying from those to whom you convey
the Program, the only way you could satisfy both those terms and this
License would be to refrain entirely from conveying the Program.

  13. Use with the GNU Affero General Public License.

  Notwithstanding any other provision of this License, you have
permission to link or combine any covered work with a work licensed
under version 3 of the GNU Affero General Public License into a single
combined work, and to convey the resulting work. The terms of this
License will continue to apply to the part which is the covered work,
but the special requirements of the GNU Affero General Public License,
section 13, concerning interaction through a network will apply to the
combination as such.

  14. Revised Versions of this License.

  The Free Software Foundation may publish revised and/or new versions of
the GNU General Public License from time to time. Such new versions will
be similar in spirit to the present version, but may differ in detail to
address new problems or concerns.

  Each version is given a distinguishing version number. If the
Program specifies that a certain numbered version of the GNU General
Public License "or any later version" applies to it, you have the
option of following the terms and conditions either of that numbered
version or of any later version published by the Free Software
Foundation. If the Program does not specify a version number of the
GNU General Public License, you may choose any version ever published
by the Free Software Foundation.

  If the Program specifies that a proxy can decide which future
versions of the GNU General Public License can be used, that proxy's
public statement of acceptance of a version permanently authorizes you
to choose that version for the Program.

  Later license versions may give you additional or different
permissions. However, no additional obligations are imposed on any
author or copyright holder as a result of your choosing to follow a
later version.

  15. Disclaimer of Warranty.

  THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT
HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY
OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM
IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF
ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

  16. Limitation of Liability.

```
  IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS
THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY
GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE
USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF
DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD
PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF
SUCH DAMAGES.

  17. Interpretation of Sections 15 and 16.

  If the disclaimer of warranty and limitation of liability provided
above cannot be given local legal effect according to their terms,
reviewing courts shall apply local law that most closely approximates
an absolute waiver of all civil liability in connection with the
Program, unless a warranty or assumption of liability accompanies a
copy of the Program in return for a fee.

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

  If you develop a new program, and you want it to be of the greatest
possible use to the public, the best way to achieve this is to make it
free software which everyone can redistribute and change under these terms.

  To do so, attach the following notices to the program.  It is safest
to attach them to the start of each source file to most effectively
state the exclusion of warranty; and each file should have at least
the "copyright" line and a pointer to where the full notice is found.

    <one line to give the program's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This program is free software: you can redistribute it and/or modify
    it under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    This program is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
    GNU General Public License for more details.

    You should have received a copy of the GNU General Public License
    along with this program.  If not, see <https://www.gnu.org/licenses/>.

Also add information on how to contact you by electronic and paper mail.

  If the program does terminal interaction, make it output a short
notice like this when it starts in an interactive mode:

    <program>  Copyright (C) <year>  <name of author>
    This program comes with ABSOLUTELY NO WARRANTY; for details type `show w'.
    This is free software, and you are welcome to redistribute it
    under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate
parts of the General Public License.  Of course, your program's commands
might be different; for a GUI interface, you would use an "about box".

  You should also get your employer (if you work as a programmer) or school,
if any, to sign a "copyright disclaimer" for the program, if necessary.
For more information on this, and how to apply and follow the GNU GPL, see
<https://www.gnu.org/licenses/>.

  The GNU General Public License does not permit incorporating your program
into proprietary programs.  If your program is a subroutine library, you
may consider it more useful to permit linking proprietary applications with
the library.  If this is what you want to do, use the GNU Lesser General
Public License instead of this License.  But first, please read
<https://www.gnu.org/licenses/why-not-lgpl.html>.
```