

Better Continents Guide

For Valheim 1.0 and Better Continents 0.10.x

Rewritten for Better Continents 0.10.x by Wubarrk

with Claude (Anthropic) as the drafting agent, October 2026

Based on the guide originally written by billw2012 and maintained by Jere Kuusela

Checked against the Better Continents source and the Valheim 1.0 game data

Licensed under the GNU General Public License, version 3. This edition is a modified version, released under the same license WITHOUT ANY WARRANTY. The license text is the final appendix; the Markdown source and build script are attached to this PDF (BetterContinents-Guide-0.10-edition-source.zip).

Contents

About this guide . . . 11
How the guide is organised . . . 11
How to read it . . . 11
Other documents . . . 12
Credits . . . 12

License . . . 13

1 What Better Continents does . . . 14
What you can control . . . 14
A world keeps what it was made with . . . 14
What's new in 0.10 . . . 14
Words used in this guide . . . 15

2 Install Better Continents . . . 17
With Gale . . . 17
By hand . . . 17
Check that it loaded . . . 17
The config file . . . 17
The Configuration Manager window . . . 18
The in-game settings window . . . 18
The console . . . 18
On a dedicated server . . . 18
Companion mods . . . 19
Updating . . . 19
Next . . . 19

3 Your first world . . . 20
The New World panel . . . 20
A world from a preset . . . 20
A world from your own maps . . . 20
A world from your config, without maps . . . 21
Check that Better Continents made the world . . . 21
The line in the Esc menu . . . 21
bc info . . . 21
The log . . . 22
Next . . . 23

4 How a world is made . . . 24
When you create a world . . . 24
Three kinds of setting . . . 24
The BetterContinents file . . . 25
Settings versions 11 and 12 . . . 25
How the maps cover the world . . . 25
In multiplayer . . . 26
What can change after the world is made . . . 26

5 Image maps . . . 27
The kinds of map . . . 27
Where maps go . . . 27
Legends . . . 28
Size . . . 28
Format . . . 28
Memory and compact maps . . . 29
Changing a map after the world is made . . . 29
Tools . . . 30
Next . . . 30

6 Heights and the heightmap . . . 31
The height formula and the waterline . . . 31
Choosing an Amount . . . 31
Working with heights at Amount 1 (default) . . . 31
Working with heights at Amount 2 . . . 32
Amounts above 5: high-world rules . . . 32
Other heightmap settings . . . 34
Heightmap Blend and Heightmap Add . . . 34
Heightmap Mask . . . 34
Heightmap Override All . . . 34
Heightmap Alpha . . . 34
Terrain records in maps . . . 34
Fine heights . . . 35
When fine heights are read . . . 35
Making a fine picture . . . 36
Why fine heights are not used . . . 36
The roughmap . . . 36
Height layers . . . 36
Editing and checking your heightmap . . . 37

7 Biomes and the biome map . . . 38
The biome colours . . . 38
The legend . . . 38
How pixels become biomes . . . 39
Leaving a biome out . . . 39
The Ashlands and the Deep North . . . 39
Biome precision . . . 39
Drawing a biome map . . . 40
Expand World Data . . . 40
Settings and commands . . . 41
What to watch for . . . 41
Next . . . 41

8 Alt biomes . . . 42
What alt biomes are . . . 42
How the game places them . . . 42
What this means for map makers . . . 43
The 32 alt biomes . . . 43
What each one changes . . . 44
Planting alt biomes with colours . . . 46
Setting up the map . . . 46
The legend . . . 46
Which pixels are planted . . . 47
The default colours . . . 47
What planting does . . . 48
The quota rule . . . 48
When planting fails . . . 48
Modes . . . 49
Steering the random placement . . . 49
Overrides . . . 50
Grid . . . 50
Commands . . . 51
Multiplayer . . . 52
Log lines . . . 52
A world laid out to its World Size . . . 52
What to watch for . . . 52
Next . . . 53

9 Ground colour, heat, lava and moss . . . 54

The terrain map . . . 54
Raw ground colours . . . 54
Expand World Data grounds . . . 55
The paint map . . . 55
Alpha, moss and lava . . . 55
The paint legend . . . 56
The heat map . . . 56
The lava map . . . 57
The moss map . . . 57
Settings . . . 57
What to watch for . . . 58
Next . . . 58

10 Forests and vegetation . . . 59
How the game decides where forest grows . . . 59
Forest Scale . . . 59
Forest Amount . . . 59
Forest Factor Overrides All Trees . . . 59
The forest map . . . 60
The vegetation map . . . 60
Doubled vegetation and bc_twins . . . 61
Settings and commands . . . 62
What to watch for . . . 62
Next . . . 62

11 Locations, creatures and the start . . . 63
The location map . . . 63
The legend . . . 63
How pins become locations . . . 64
Drawing the location map . . . 64
Checking the result . . . 64
The spawn map . . . 64
The start position . . . 65
Skip Default Locations . . . 66
Settings and commands . . . 66
What to watch for . . . 66
Next . . . 66

12 Worlds from noise . . . 67
Where the terrain comes from . . . 67

The presets that ship with the mod . . . 67
Height layers . . . 67
How layers combine . . . 67
Building layers . . . 68
The Global settings . . . 68
Continent Size . . . 69
Sea Level Adjustment . . . 69
Mountains Allowed At Center . . . 69
Rivers, the gaps and the edge . . . 69
Settings older worlds still use . . . 70
What to watch for . . . 70
Next . . . 70

13 World size and big worlds . . . 71
World Size and Edge Size . . . 71
What follows the world's size . . . 71
Wide Sectors: worlds past 16 km . . . 71
Why big worlds need it . . . 71
What Wide Sectors does . . . 72
The setting . . . 72
Converting an existing world . . . 73
Memory, start time and joining . . . 73
Tall terrain in a big world . . . 74
The sea and the Deep North in a big world . . . 74
Limits . . . 74
Settings . . . 74
Next . . . 75

14 Changing an existing world . . . 76
What reaches an existing world . . . 76
Debug Mode and the bc commands . . . 76
Using the commands . . . 77
A change is kept . . . 77
What a change does . . . 77
Zone regeneration . . . 77
What is left alone . . . 78
What is removed . . . 78
The ground . . . 78
How it runs . . . 79
Where it doesn't run . . . 79

Resetting part of the world . . . 79
bc regen and bc reset . . . 80
Debug Reset Command . . . 80
Reloading a map . . . 81
Converting a big world to Wide Sectors . . . 81
Override version . . . 81
Backups . . . 82
Settings . . . 82
Next . . . 82

15 Multiplayer and dedicated servers . . . 83
Everyone on the same version . . . 83
How the settings reach players . . . 83
How long a join takes . . . 83
Letting players skip the download . . . 84
Running a dedicated server . . . 85
A joining player's own config . . . 85
Settings . . . 86
Next . . . 86

16 Coming from 0.9.x or older . . . 87
Before you update . . . 87
Which worlds get the new behaviour . . . 87
What a version 12 world does differently . . . 88
Changes for every world . . . 89
Your BetterContinents.cfg . . . 89
Changed once . . . 89
Gone . . . 89
New . . . 90
Changed . . . 90
Mixing versions . . . 90
In multiplayer . . . 91
A version 12 world opened in 0.9.x . . . 91
A 0.9.x world in 0.10, and back . . . 91
Going back to an older version . . . 91
Worlds from before Valheim 1.0 . . . 92
Next . . . 92

17 Other mods . . . 93
Expand World Size . . . 93

Expand World Data . . . 93
Its biomes on a biome map . . . 94
In a version 12 world . . . 94
Current releases . . . 94
Map mods and no-map worlds . . . 95
Server Devcommands . . . 95
Upgrade World . . . 95
World radius . . . 96
Locations . . . 96
Resetting by biome . . . 96
Other commands . . . 96
Mods that change how plants are placed . . . 96
Next . . . 97

18 Export and import . . . 98
Exporting a world . . . 98
Who can export . . . 98
The export window . . . 98
Heights in an export . . . 99
Sizes, time and memory . . . 99
Fine heights in an export . . . 100
Making a world from an export . . . 100
Which settings version the new world gets . . . 101
Settings . . . 101
Next . . . 101

19 Every setting . . . 102
[00 BetterContinents.Debug] . . . 102
[01 BetterContinents.Global] . . . 103
[02 BetterContinents.Heightmap] . . . 104
[03 BetterContinents.Biomemap] . . . 105
[04 BetterContinents.Forest] . . . 105
[05 BetterContinents.StartPosition] . . . 105
[06 BetterContinents.Maps] . . . 106
[07 BetterContinents.Misc] . . . 106
[08 BetterContinents.AltBiomes] . . . 109
[09 BetterContinents.Export] . . . 110
Next . . . 111

20 Console commands . . . 112

Where the commands work . . . 112
How bc works . . . 112
World information . . . 113
Regenerating the world . . . 113
bc genloc . . . 113
Maps . . . 114
Reloading a map . . . 114
Choosing a map's file . . . 115
Settings . . . 115
Height layers . . . 116
Alt biomes . . . 117
The minimap, pictures and presets . . . 118
Export, import, cache and twins in the bc tree . . . 118
Commands for every player . . . 118
bc_altbiomes . . . 118
bc_export . . . 118
bc_import . . . 119
bc_cache . . . 119
bc_twins . . . 120
Next . . . 120

21 Troubleshooting and FAQ . . . 121
Finding things . . . 121
Where are my worlds? . . . 121
Where is the log? . . . 121
Where is BetterContinents.cfg? . . . 121
The world is not what I made . . . 121
The menu says Better Continents is DISABLED for my world. . . . 121
My world is all ocean, or much lower than I painted it. . . . 122
I edited my map pictures, but the world did not change. . . . 122
A map was refused as too big. . . . 122
Fine heights are not used. . . . 122
Locations are missing. . . . 123
An alt biome is not where I painted it, or not anywhere. . . . 123
I changed Biome precision, but my world looks the same. . . . 123
A square of water looks deeper or shallower than the water around it. . . . 124
Trees and rocks stand in pairs, a copy within a metre of each one. . . . 124
Multiplayer and big worlds . . . 124
A player cannot join: the versions differ. . . . 124
Joining takes a long time. . . . 124

A world bigger than 16 km loses creatures, objects or portals. . . . 124
Versions . . . 125
I want to open a world in an older Better Continents. . . . 125
Exports . . . 125
I can't find my exports. . . . 125
The export window doesn't offer 16,384 px. . . . 125
A 16,384 px export made no New World preset. . . . 125
Other mods . . . 125

Appendix A Formulas . . . 126
Height from a heightmap . . . 126
Waterline . . . 126
Map span and pixel size . . . 126
The 16-bit step . . . 127
Fine heights . . . 127
Sea Level Adjustment . . . 127
Example: choosing Heightmap Amount . . . 128

Appendix B Credits and history . . . 129
The mod . . . 129
The guide . . . 129
Versions for Valheim 1.0 . . . 129

Appendix C GNU General Public License, version 3 . . . 130

About this guide

This is the Better Continents Guide for Valheim 1.0 and Better Continents 0.10.x. It is for Valheim players, server admins and map makers who use Better Continents, and it assumes no programming. Each chapter starts from something you want to do, such as making your first world, drawing a heightmap or running a server, and takes you through it.

Better Continents 0.10 works differently from earlier versions, so this edition was written from the ground up. It describes how 0.10.x works now. What changed from version to version is in Credits and history and in the mod's changelog.

How the guide is organised

Part	Chapters	What it covers
Getting started	What Better Continents does, Install Better Continents, Your first world, How a world is made	Installing the mod, making a world, and what happens when a world is made
Maps	Image maps, Heights and the heightmap, Biomes and the biome map, Alt biomes, Ground colour, heat, lava and moss, Forests and vegetation, Locations, creatures and the start	The pictures that build a world, one kind at a time
Worlds without maps	Worlds from noise	Procedural terrain and the presets
Running a world	World size and big worlds, Changing an existing world, Multiplayer and dedicated servers, Coming from 0.9.x or older, Other mods	Big worlds, changes to a world you have, servers, updates and other mods
Reference	Export and import, Every setting, Console commands, Troubleshooting and FAQ	Chapters to look things up in
Appendices	Formulas, Credits and history, and the text of the GNU General Public License	

Each chapter can be read on its own. Where a chapter touches a topic that another chapter covers, it says so in a sentence and links to it.

How to read it

- A setting is named as it appears in `BetterContinents.cfg`, in backticks. The first time a chapter uses it, its section follows: `Heightmap Amount` in `[02 BetterContinents.Heightmap]`.
- Console commands are in boxes marked **Console**. You type them in the game's console, which F5 opens.
- **Warning** boxes mark choices that cannot be undone and anything that can lose data. Read them before you act.
- Times and memory figures are measurements from test machines. Yours will differ with your hardware and your maps.
- "Valheim 1.0" means the game as it is now, and "Better Continents 0.10.x" any 0.10 release. An exact version appears only where it matters, as in "a world with Wide Sectors needs Better Continents 0.10.3 or later".

Other documents

The Better Continents package on Hexium carries this guide and two more:

- *Export & Import, the easy guide* (`BetterContinents-Export-Guide.pdf`): step-by-step instructions for exporting a world to pictures, editing them and making new worlds from them, written for first-timers.
- *Map-making, a skill for AI agents* (`BetterContinents-AI-Skill.pdf`): a document that teaches an AI agent how to make Better Continents maps.

The source code of Better Continents is at github.com/RGlabs84/BetterContinents10, where each release is tagged. For help, or to share your worlds, join the community Discord.

Credits

The original Better Continents Guide was written by billw2012 and is maintained by Jere Kuusela, at jerekuusela.github.io/BetterContinents-Docs. This edition was rewritten from the ground up for Better Continents 0.10.x by Wubarrk, who now maintains Better Continents, with Claude (Anthropic) as the drafting agent. It is licensed under the GNU General Public License, version 3: see License.

License

The *Better Continents Guide* is licensed under the **GNU General Public License, version 3** (GPL-3.0). It was originally written by billw2012 and is maintained by Jere Kuusela, at github.com/JereKuusela/BetterContinents-Docs, a fork of github.com/billw2012/BetterContinents-Docs.

This edition is a modified version. In October 2026 Wubarrk rewrote the guide from the ground up for Valheim 1.0 and Better Continents 0.10.x, organised by what the reader wants to do, with Claude (Anthropic) as the drafting agent. It draws on an earlier edition for Valheim 1.0, which Wubarrk updated chapter by chapter from September 2026, for Better Continents 0.8.0 to 0.10.2.

This edition is released under the same licence, the GNU General Public License, version 3. 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 last appendix, *GNU General Public License, version 3*.

Source of this edition. The Markdown source of this edition, the licence and the script that builds this PDF are attached to the PDF as the file `BetterContinents-Guide-0.10-edition-source.zip`. Open your PDF viewer's attachments panel to save it.

The mod's own licence. This licence covers the guide, not the mod. Better Continents itself is free software under the GNU Lesser General Public License, version 2.1, and its source code is at github.com/RGlabs84/BetterContinents10.

1 What Better Continents does

Better Continents is a world-generation mod for Valheim 1.0. You give it pictures, settings or both, and it makes a Valheim world from them. A grey heightmap says how high the ground is, a colour map says which biome is where, and other maps place forests, locations, creatures, lava and more. Without pictures, procedural noise can shape the land.

This chapter says what you can control, what is new in 0.10, and what the words used throughout the guide mean.

What you can control

You want to decide	With	Chapter
How high the ground is, from the sea bed to peaks up to 16 km high	a heightmap, a rough map, height layers	Heights and the heightmap
Which biome is where	a biome map	Biomes and the biome map
Where Valheim's alt biomes go, such as Dark Meadows and Lox Plains	an alt-biome map and the alt-biome settings	Alt biomes
The ground's colour and paint, and where heat, lava and moss are	the terrain, paint, heat, lava and moss maps	Ground colour, heat, lava and moss
Forests, clearings and which plants grow where	the forest settings, a forest map, a vegetation map	Forests and vegetation
Where bosses, traders and other locations stand, which creatures spawn where, and where players start	a location map, a spawn map, the start position	Locations, creatures and the start
How big the world is	`World Size` and `Edge Size` in `[01 BetterContinents.Global]`	World size and big worlds
Land without any pictures	height layers and the presets	Worlds from noise

You don't need every map. A world can use one, several or none.

A world keeps what it was made with

When a world is made, Better Continents stores its settings and maps inside the world's folder, in a file named `BetterContinents`. From then on the world reads them from there. Editing `BetterContinents.cfg` or a picture afterwards changes only the worlds you make next. To change a world you already have, use the `bc` commands: see Changing an existing world.

Players who join a server get the world's settings and maps from the server, so they need neither the pictures nor the config. The server and every player must run exactly the same version of Better Continents.

What's new in 0.10

- **Existing worlds stay as they were.** Each world stores a settings version. A world made by 0.9.x keeps generating its terrain, biomes and locations as it did under 0.9.4. The new behaviour is for new worlds, those of version 12. A few changes reach every world, such as maps that take less memory and zone regeneration. See Coming from 0.9.x or older.
- **One definition of every setting.** The config file, the `bc` console, the world export and the import all take a setting's name, default and range from one place, so they agree.

- **Worlds made to any size.** On a new world, `World Size` and `Edge Size` lay out the whole world: its maps, the alt-biome grid, where locations go, the game's biome bands, the Ashlands and Deep North rings, lakes, streams and the minimap.
- **Zone regeneration.** After a change made with the `bc` commands, Better Continents regenerates the world's zones itself, and leaves alone the zones around what players built or worked on.
- **Maps up to 16,384 × 16,384 pixels.** A plain PNG is read a few rows at a time, with little memory, and a new world with a map more than 8,192 px across keeps its maps compact (`Compact Maps` is now Auto, On or Off).
- **Worlds past 16 km.** By default, a new world whose World Size + Edge Size is more than 16,350 m is made with Wide Sectors, which gives every zone out to 65.5 km from the centre its own place in the game's files. A world with Wide Sectors needs Better Continents 0.10.3 or later. See World size and big worlds.
- **Big worlds load in seconds.** A world of World Size 32,264 m with five 16,384 px maps was up in 16 s on a 32-core server, measured on one machine; Better Continents builds the game's alt-biome grid on every core. Before 0.10.3, the game's lake merge alone took over 7 minutes on that world.
- **Terrain up to 16 km high.** `Heightmap Amount` in `[02 BetterContinents.Heightmap]` goes up to 81, and Better Continents adjusts the game's height rules for a world whose heightmap is read at an Amount above 5 (`High Terrain` decides which worlds get them).
- **Fine heights.** An optional second picture refines a 16-bit heightmap to 1/256 of a step, for tall worlds, and the world export writes one from a Heightmap Amount of about 5. A world made with fine heights needs Better Continents 0.10.3 or later.
- **Exports at 16,384 px.** The world export writes maps up to 16,384 px, and holds about 100 MB of memory while it runs, at any size.

Earlier releases for Valheim 1.0 added painting alt biomes with colours, the world export and import, and a world cache file that lets players join a big world without a download. Credits and history lists them.

Words used in this guide

- **Zone:** a 64 m square of the world. The game fills a zone with its trees, rocks, locations and creature spawners once, the first time somebody comes near, and then keeps what it made.
- **Map:** a picture, saved as a PNG, that decides one thing about the world. Every map covers the whole world, with north at the top. See Image maps.
- **Legend:** the text file beside a colour map that says what each colour means, such as `biomemap.txt` beside `biomemap.png`.
- **Heightmap:** a grey map of how high the ground is, black lowest and white highest. At the default `Heightmap Amount` of 1, black is 60 m below the water and white is 140 m above it.
- **Height layers:** procedural noise that Better Continents stacks to shape the land, on its own or on top of a heightmap. See Worlds from noise.
- **World Size:** the world's radius in metres, from the centre to the edge. Vanilla's is 10,000.
- **Edge Size:** how far the world's edge reaches past World Size, in metres. Vanilla's is 500. A new world's maps span 2 × (World Size + Edge Size) from side to side: 21,000 m at the defaults.
- **New World panel:** the game's screen for making a world. Better Continents adds a dropdown to it, where you choose Disabled, a preset, or From Config.
- **From Config:** the dropdown choice that makes a world from `BetterContinents.cfg` as it is when you create the world.

- **Preset:** a saved recipe for a world, its settings and maps. Presets live in the `BetterContinents/presets` folder beside the game's `worlds_local` folder, and appear in the dropdown.
- **Directory:** the setting in `[00 BetterContinents.Debug]` that points at a folder of maps with standard names.
- **Settings version:** a number each world stores. Version 12 is a world made by 0.10 with the newest rules. Version 11 is a world made by 0.9.x, or one made in 0.10 from an old preset, and it keeps the 0.9.4 rules. See How a world is made.
- **The `BetterContinents` file:** the file named `BetterContinents`, with no extension, in a world's folder. It holds the world's settings and maps. A world whose folder has no such file is a world without Better Continents.
- **World, Live and Local settings:** the three kinds of setting. World settings are defaults for new worlds, Live settings apply while the game runs, and Local settings are this machine's own switches. See How a world is made.
- **Debug Mode:** the switch for editing a world live with the `bc` commands and the settings window. See Changing an existing world.
- **Export:** a folder of maps, with an `export.cfg`, that rebuilds the world it was made from. You make one with `bc_export` or the export window. See Export and import.
- **Compact maps:** maps kept as compressed tiles in memory, in the world file and in what a server sends to joining players. By default a new world keeps its maps compact when any of them is more than 8,192 px across, or when its heightmap has fine heights.
- **Sectors:** how the game files the world's objects (trees, rocks, creatures, buildings) for saving and for sending to players. The game's own sectors stop about 16.4 km from the centre; Wide Sectors extends them to 65.5 km.

2 Install Better Continents

Better Continents is a plugin for BepInEx, a mod loader for Valheim. Install it on your own computer to make and play worlds, and on a dedicated server if you run one. Every machine that plays a Better Continents world, the server and every player, must run exactly the same version of Better Continents.

With Gale

Better Continents is published on Hexium. The easiest way to install it is from that page with the Gale mod manager, which installs BepInEx along with it.

1. Install Gale and make a profile for Valheim, or use one you have.
2. Install Better Continents into that profile from its Hexium page.
3. Start Valheim from Gale, with that profile.

A mod manager keeps each profile in a folder of its own, with its own `BepInEx` folder. Better Continents' config file and log are in that folder, not in the game's folder (see below).

By hand

1. Back up your worlds and characters.
2. Install BepInEx for Valheim in the game folder, following its instructions. The Better Continents package lists `BepInExPack_Valheim` 5.4.2350 as its dependency.
3. Copy `BetterContinents.dll` from the Better Continents package into the game's `BepInEx/plugins/` folder.
4. Start the game once. BepInEx loads Better Continents, and Better Continents writes its config file.

Check that it loaded

- The New World panel has a Better Continents dropdown. Its first choice is **Disabled**.
- The BepInEx log, `BepInEx/LogOutput.log`, has a line `Loading [Better Continents 0.10.x]` near the start, with the version you installed.

The log is written again each time the game starts, so copy it before you restart if you want to keep it.

The config file

Better Continents' settings are in `BepInEx/config/BetterContinents.cfg`: in the game folder, or in the profile's folder when you use a mod manager. Better Continents writes the file the first time the game starts with it.

- Edit it with any text editor. Numbers use a dot as the decimal separator.
- Better Continents reads the file again whenever it changes on disk, so you don't need to restart the game.
- Most settings are defaults for the worlds you make next. A world keeps the settings it was made with, and editing the file never changes a world you already have. How a world is made explains which settings apply when.
- A new file has the New World panel's dropdown on **Disabled**, so a world you make without changing it does not use Better Continents.

Every setting is listed in Every setting.

The Configuration Manager window

The BepInEx Configuration Manager is a separate, optional mod that shows every mod's settings in the game. Press F1, open the Better Continents sections and change a value. There is no save button: it saves the file as you go, and Better Continents uses a change straight away.

The in-game settings window

`Debug Mode` in `[00 BetterContinents.Debug]` (default off) is for building worlds. In a world that uses Better Continents, on the machine that runs the world, it:

- turns on `devcommands` when you spawn, which the `bc` console commands need;
- shows *Better Continents Debug Mode Enabled!* at the top left of the screen;
- adds a **Better Continents** button to the Esc menu and to the large map, which opens the settings window. Alt+F8 does the same while the Esc menu or the large map is open.

The settings window changes the world you are in, as the `bc` commands do. It does not change `BetterContinents.cfg`. Leave Debug Mode off for a world you play, and read Changing an existing world before you use it.

The console

Better Continents' commands are typed in the game's console, which F5 opens. If F5 does nothing, turn the console on in the game's Gameplay settings, or start Valheim with the `-console` launch option.

The `bc` commands are cheat commands, so they need `devcommands` on, and they run on the machine that runs the world: single player, or the host. `bc_export`, `bc_import`, `bc_cache`, `bc_twins` and `bc_altbiomes` work for every player. See Console commands.

On a dedicated server

A dedicated server needs BepInEx and the same version of Better Continents as its players.

1. Stop the server, and back up its worlds.
2. Install BepInEx on the server. How depends on your host: some have a switch that enables mods, and others need the BepInExPack_Valheim server install.
3. Put the same `BetterContinents.dll` in the server's `BepInEx/plugins/` folder.
4. Make the world on your own computer and check it there. A dedicated server cannot make a Better Continents world itself: a world it creates comes out vanilla.
5. Copy the whole world folder, `worlds_local/<World>/`, to the server's `worlds_local/` folder. It holds the game's files (`_main.N.fwl2`, `.db2`, `.chunks`, `.ok` and the chunk files) and a file named `BetterContinents`, with no extension, which holds the world's settings and maps. If your host filters uploads by file extension, check that `BetterContinents` arrived: without it the world loads as a vanilla world.
6. Point the server at the world by its name, start the server, and join.

On your own computer, the world folders are in `%AppData%\..\LocalLow\IronGate\Valheim\worlds_local` on Windows and `~/.config/unity3d/IronGate/Valheim/worlds_local` on Linux.

Players do not need the world's pictures or your config: the server sends them the world's settings when they join. For big worlds that takes time; see Multiplayer and dedicated servers.

Companion mods

The Better Continents README recommends these mods. Each is optional.

- Server Devcommands: admin tools for testing a world, such as flying, god mode and a free camera. With version 1.51 or later, Better Continents adds tab completion for its `bc` command.
- Upgrade World: commands that reset zones and place locations in a world you are building. Better Continents regenerates zones itself after a `bc` change, and can run Upgrade World's `zones_reset start` instead: see Other mods.
- Infinity Hammer: for building ruins, landmarks and settlements into your terrain.
- TortalPortal: a portal mod with destination previews, favourites and server synchronisation, for travel across big continents.

Updating

Update the server and every player together: players with another version of Better Continents are refused when they join a Better Continents world. Back up your worlds before you update. A world made with a newer version may not open correctly in an older one: Coming from 0.9.x or older lists what cannot be undone.

Next

- Your first world: make a world and check that Better Continents is working.
- How a world is made: what happens to your settings when you create a world.

3 Your first world

This chapter makes a world three ways: from one of the presets that ship with the mod, from your own maps, and from your config alone. Then it shows how to check that Better Continents made the world: the line in the Esc menu, `bc info` and the log.

Install Better Continents first: see Install Better Continents.

The New World panel

In the main menu choose **Start Game**, pick your character and press **Start**, then **New World**. Beside the name and the seed, Better Continents adds a dropdown with these choices:

Choice	The new world
Disabled	does not use Better Continents. This is the choice a new config starts with.
A preset	takes the preset's settings and maps. The four presets that ship with the mod are Continents, Pangaea, Swirls and Vanillaish; presets you save, and the presets that world exports make, appear here too.
From Config	takes `BetterContinents.cfg` as it is when you create the world.

The world takes its settings once, when you create it, and keeps them. A world name needs at least five characters.

A world from a preset

The quickest way to see Better Continents at work:

1. Open the New World panel.
2. Pick **Continents**, **Pangaea**, **Swirls** or **Vanillaish** in the dropdown.
3. Type a name and a seed, and create the world.

These presets hold no pictures. Their land comes from height layers, Better Continents' procedural noise, so another seed gives another world. They are old preset files (settings version 7), so a world made from one is a version 11 world: the world that preset has always made. Worlds from noise says more about presets and how to save your own.

A world from your own maps

1. **Put your maps in one folder**, with the standard names: `heightmap.png`, `biomemap.png`, `forestmap.png` and so on. A heightmap alone is enough to start. Image maps lists every name and the picture each map needs.
2. **Point Better Continents at the folder.** In `BetterContinents.cfg`, set `Directory` in `[00 BetterContinents.Debug]` to the folder's full path:

   ```
   Directory = C:\Users\<you>\Maps\MyWorld
   ```

   On Linux it looks like `Directory = /home/<you>/Maps/MyWorld`.
3. **Set the Amount your heightmap was drawn for.** `Heightmap Amount` in `[02 BetterContinents.Heightmap]` decides how tall white is, and where the waterline falls in your picture: at the default of 1 the waterline is the grey value 0.30 (77 of 255), and at 2 it is 0.15. A heightmap read at the wrong Amount comes out drowned or too high. See Heights and the heightmap.
4. **Save the file.** Better Continents reads it again as soon as it changes.

5. **Create the world:** open the New World panel, choose **From Config** in the dropdown, type a name and a seed, and create it. Better Continents reads your maps and stores them in the world.
6. **Check it**, as below.

Making a world from big maps takes a while: from 16,384 px maps, 17 to 39 s (measured on one machine), while the game shows **Please wait** and Better Continents reads them. The game keeps running, and the world is in the list when the popup closes.

A world made this way is a version 12 world, which gets every new-world rule of 0.10. Other players don't need your pictures: the world carries them.

> **Note:** A folder that holds an `export.cfg` is a world export. Its `export.cfg` comes with the maps, and its settings win over your config for the keys it lists. See Export and import.

A world from your config, without maps

With **From Config** and no maps (`Directory` empty and no map files set), the world uses Better Continents, but its ground is the game's own. Without a heightmap or height layers the terrain is vanilla, and `Continent Size` and `Sea Level Adjustment` in `[01 BetterContinents.Global]` have no effect.

For procedural land, start from a preset, or build height layers in a world with the `bc hl` commands or the settings window, and save them as a preset of your own with `bc savepreset`. See Worlds from noise.

Check that Better Continents made the world

The line in the Esc menu

Press Esc in the world. At the bottom of the screen, Better Continents shows one of:

```
Better Continents v<version>: ENABLED for this world
Better Continents v<version>: DISABLED for this world
```

The same line shows while you wait to respawn. **ENABLED** means the world uses Better Continents. **DISABLED** means it does not. The usual reasons:

- the dropdown was on **Disabled** when you created the world;
- `Enabled` in `[00 BetterContinents.Debug]` was off when you created the world From Config;
- the preset you picked could not be read (the log says so);
- the world's folder has lost its `BetterContinents` file, for example in a copy or an upload, or the world was created by a dedicated server.

A world made without Better Continents stays that way. Make a new one with the right choice.

bc info

`bc info` prints the settings of the world you are in. It is a cheat command, so turn cheats on first:

> **Console**
> Command: `devcommands`, then `bc info`
>
> Prints the world's settings, starting with its settings version. Run it in single player or on the host. With `Debug Mode` in `[00 BetterContinents.Debug]` on, cheats are already on when you spawn.

For a world made from your own maps at Amount 2, it starts like this (the `...` marks lines left out):

```
Version 12
...
Heightmap file (4096) C:\Users\<you>\Maps\MyWorld\heightmap.png
Heightmap amount 2, blend 1, add 0, mask 0
Heightmap overrides ALL
...
Biomemap file (4096) C:\Users\<you>\Maps\MyWorld\biomemap.png
```

Each map the world uses is listed with its size in pixels, and a map it doesn't use says `disabled`. In a world without Better Continents, `bc info` prints `DISABLED`.

The log

Better Continents writes to the BepInEx log, `BepInEx/LogOutput.log` in the game folder or in your mod manager's profile. Its lines contain `[BetterContinents]`, so search for that. The log is written again at each start of the game. When you create a world, look for lines like these, in this order:

```
[Saving][MyWorld] bWorldBeingCreated flag set, first time save of MyWorld, applying selected
  preset 'From Config'
Time to load C:\Users\<you>\Maps\MyWorld\heightmap.png: <time> ms
Version 12
```

The first line names the dropdown choice the world was made with. Each map that was read has a `Time to load <file>: <time> ms` line, and the world's settings follow, starting with `Version`. The settings are printed again each time the world loads.

Lines that point to a problem:

Line	What it means
No `Time to load <file>` line for a map	With `Directory` set, a map whose file is not in the folder under its standard name is left out without a word. Check the folder and the file name.
`Cannot find image <file>: ...`	A map's file setting names a file that isn't there.
`Cannot use texture <file>: it is W x H pixels, and the largest map Better Continents reads is ...`	The picture is bigger than 16,384 px, or bigger than `Max Map Size` in `[07 BetterContinents.Misc]` allows. Scale it down.
`Cannot use texture <file>: its width (W) does not match its height (H)`	The picture is not square.
`Cannot load texture <file>: ...`	The picture could not be decoded. Save it again as a plain PNG.
`... was made by a world export for Heightmap Amount ... but this new world reads it at ...`	The heightmap comes from an export made for another Amount or Sea Level. Use the export's `export.cfg`, or set the values it names.
`Compact Maps is Auto and <map> <file> is W x H pixels (more than 8192 across), so this world is compact. ...`	A map is more than 8,192 px across, so the world keeps its maps compact. Not a problem.
`Compact Maps is on for this world: its heightmap has fine heights ...`	The heightmap has a `heightmap-fine.png`, so the world keeps its maps compact. Not a problem.
`Compact Maps is Off, so this world keeps its maps as pictures although ...`	`Compact Maps` in `[07 BetterContinents.Misc]` is Off and a map is more than 8,192 px across: the world file, and what every joining player is sent, come to hundreds of megabytes. Set it to Auto and make the world again.
`Fine heights: ...`	What the world made of a `heightmap-fine.png` beside the heightmap, or why it did not use it. See Fine heights.

Troubleshooting and FAQ covers more.

Next

- How a world is made: what the world keeps, and why the config no longer changes it.
- Image maps: every kind of map, and how to draw them.
- Changing an existing world: changing a world after it is made.

4 How a world is made

This chapter explains what Better Continents does with your settings and maps when you create a world, what the world keeps, and why editing the config or a picture afterwards does not change it. Read it once before you build a world you mean to keep.

When you create a world

You create a world in the New World panel, with the Better Continents dropdown on one of three kinds of choice. The world takes its settings once, at that moment:

Choice	Where the new world's settings come from
Disabled	Nowhere: the world does not use Better Continents.
A preset	The preset file: its settings and maps, at the settings version the preset was saved with. A preset saved before alt biomes could be planted takes the config's alt-biome settings.
From Config	`BetterContinents.cfg` as it is at that moment, and the maps it names. When `Directory` in `[00 BetterContinents.Debug]` points at a world export (a folder with an `export.cfg`), the settings in `export.cfg` win over the config's.

Better Continents reads the maps, puts together the world's settings and writes them into the world's folder, in the `BetterContinents` file, when the game first saves the new world, which it does as you create it. From then on the world reads its settings and maps from that file, and never again from the config or the pictures.

Three kinds of setting

Each setting in `BetterContinents.cfg` has one of three scopes, which decide when it is read:

Scope	When it is read	Examples
World	Once, when a world is created (From Config, an import, or a `Directory` with an `export.cfg`). The world stores the value, and the config's value is never read again for that world.	`World Size` and `Sea Level Adjustment` in `[01 BetterContinents.Global]`, `Heightmap Amount` and `High Terrain` in `[02 BetterContinents.Heightmap]`, the map files, the forest, start-position and alt-biome settings, `Directory` and `Enabled` in `[00 BetterContinents.Debug]`, `Compact Maps` in `[07 BetterContinents.Misc]`
Live	While the game runs. A change applies at once, or the next time the setting is needed.	`Settings Transfer Rate`, `Map Memory`, `Max Map Size` and `Wide Sectors` in `[07 BetterContinents.Misc]`, and every setting in `[09 BetterContinents.Export]`. `Wide Sectors` is read by the machine that makes a world, and by the machine that runs it when it loads.
Local	This machine's own switches. Never part of a world, an export or an import.	`Debug Mode` and `Debug Reset Command` in `[00 BetterContinents.Debug]`

Live and Local settings belong to the machine, not to a world: an `export.cfg` line for one of them is ignored when it is imported. `Override version` in `[00 BetterContinents.Debug]` is a World setting with one exception: it is also read at every save, so it can change a world you have. Leave it empty; see Changing an existing world.

Every setting gives the scope of each setting.

The BetterContinents file

A Better Continents world has a file named `BetterContinents`, with no extension, in its folder, `worlds_local/<World>/`. It holds:

- the world's settings version;
- every setting the world was made with, and any change made since with the `bc` commands;
- the world's maps, as pictures or, for a compact world, as compressed tiles;
- which file each map came from, so that `bc reload` can read it again.

It is a binary file, not text. To read it, load the world and use `bc info`.

Better Continents writes the file each time the game saves the world, and keeps the copy before as `BetterContinents.old`. Because the maps are inside it, a world needs nothing else: to share a world or move it to a server, copy the whole world folder. A world folder without the file loads as a world without Better Continents.

A world's settings and maps can take up to 1.5 GB in all. With `Compact Maps` on Auto, the default, a world with a map more than 8,192 px across is made compact, which keeps the file small: one world file with five 16,384 px maps came to 52.6 MB compact, against 352.5 MB as pictures. See Image maps.

Settings versions 11 and 12

Each world stores the settings version it was made with, and Better Continents follows that version's rules for as long as the world exists. This is why updating Better Continents doesn't change the worlds you already have.

- **Version 12** is a world made by Better Continents 0.10 From Config, or from a preset or export of a version 12 world. It gets the new-world behaviour of 0.10: maps that span its World Size + Edge Size, the rest of the world laid out to that size, Heightmap Alpha that blends, Biome precision up to 31, location pins on their exact pixels, and Expand World Data's rules over the heightmap.
- **Version 11** is a world made by Better Continents 0.9.x, or a world made in 0.10 from an older preset (the four presets that ship with the mod are version 7 files), from an export of a world resized under 0.9.x, or with `Override version` set to 11. It generates as under 0.9.4: for one, its maps span 21,000 m whatever its World Size says.
- A world of an older settings version (1 to 10), made by a much older Better Continents, is saved as version 11.

Nothing converts an existing world from version 11 to 12. Coming from 0.9.x or older sets out every difference between the two, and which changes reach every world.

To see a world's version, use `bc info`, whose first line is `Version 11` or `Version 12`, or read the settings that the log prints when the world loads.

How the maps cover the world

- **The square.** On a version 12 world, every map covers a square from −(World Size + Edge Size) to +(World Size + Edge Size) on each axis, centred on the world's centre: 21,000 m across at the default World Size of 10,000 m and Edge Size of 500 m. On a version 11 world the square is 21,000 m across whatever World Size says. With Expand World Size installed, its size decides instead (see Other mods).
- **North is the top** of the picture, and west the left.
- **Pixel size.** One pixel covers 2 × (World Size + Edge Size) / the picture's width: 5.13 m for a 4,096 px map at the default sizes, and 2 m for a 16,384 px map with World Size 15,850 and Edge Size 500.

- **Between pixels.** Better Continents reads a map between the centres of its pixels. At a point between them, a grey map such as the heightmap gives a blend of the four nearest pixels, so the ground runs smoothly from one pixel to the next; the biome map gives the biome that weighs most among those four pixels.

Image maps covers sizes and formats, and Formulas the arithmetic.

In multiplayer

The machine that runs the world, a dedicated server or the player who hosts, sends the world's settings and maps to each player who joins, before the join finishes. The player's game keeps them in its cache, so a later join needs no download as long as the world's settings have not changed. A joining player's own `BetterContinents.cfg` has no say in the world: everyone plays the world's own settings. The server and every player must run the same version of Better Continents. See Multiplayer and dedicated servers.

What can change after the world is made

- Editing `BetterContinents.cfg` or a picture changes only the worlds you create afterwards.
- Live settings, such as `Settings Transfer Rate`, apply while the game runs.
- The `bc` commands and the settings window, in Debug Mode on the machine that runs the world, change the world's own settings, and the world saves them. `bc reload` reads a picture into the world again.
- `Wide Sectors` set to On converts an existing big world when it loads. That is one way: see World size and big worlds.

See Changing an existing world for how, and what it costs.

5 Image maps

Better Continents builds a world from pictures. Each kind of map decides one thing about the world: how high the ground is, which biome is where, where forests grow, and so on. You draw them in an image editor such as GIMP or Krita, or make them with a script. This chapter covers what all maps share: the kinds, where they go, their size, their format and the memory they take. The chapters after it cover each kind.

The kinds of map

Map	Picture	What it decides	Chapter
Heightmap	grey, 16-bit	How high the ground is	Heights and the heightmap
Roughmap	grey	Where the biomes' own rough ground shows through the heightmap	Heights and the heightmap
Biome map	colour, with a legend	Which biome is where	Biomes and the biome map
Alt-biome map	colour, with a legend	Where Valheim 1.0's alt biomes are planted	Alt biomes
Terrain map	colour, with a legend	Which biome's ground texture shows, where it should differ from the biome	Ground colour, heat, lava and moss
Paint map	colour, with a legend	Paint over the ground: dirt, cultivated, paved, moss and lava	Ground colour, heat, lava and moss
Heat map	grey	Where the ground and water are hot, as in the Ashlands	Ground colour, heat, lava and moss
Lava map	grey	Where lava can appear in the Ashlands	Ground colour, heat, lava and moss
Moss map	grey	Where moss can appear in the Mistlands	Ground colour, heat, lava and moss
Forest map	grey	How much forest grows where	Forests and vegetation
Vegetation map	colour, with a legend	Which plants may grow where	Forests and vegetation
Location map	colour, with a legend	Where chosen locations are placed: bosses, traders, the start temple	Locations, creatures and the start
Spawn map	colour, with a legend	Which creatures may spawn where	Locations, creatures and the start

You don't need all of them. A world can use one map, several, or none (see Worlds from noise).

A heightmap can also have a second picture beside it, `heightmap-fine.png`, which refines its heights for tall worlds (see Fine heights).

Where maps go

There are two ways to give Better Continents its maps.

A folder. Set `Directory` in `[00 BetterContinents.Debug]` to a folder, and Better Continents uses the files in it that have these names: `heightmap.png`, `roughmap.png`, `biomemap.png`, `altbiomemap.png`, `terrainmap.png`, `paintmap.png`, `heatmap.png`, `lavamap.png`, `mossmap.png`, `forestmap.png`, `vegetationmap.png`, `locationmap.png` and `spawnmap.png`. A folder that holds an `export.cfg` is a world export, and its settings are used too (see Export and import).

One file at a time. Leave `Directory` empty and set each map's own file setting:

Maps	Settings	Section
Heightmap, roughmap	`Heightmap File`, `Roughmap File`	`[02 BetterContinents.Heightmap]`
Biome map, terrain map	`Biomemap File`, `Terrainmap file`	`[03 BetterContinents.Biomemap]`
Forest map	`Forestmap File`	`[04 BetterContinents.Forest]`
Location, spawn, vegetation, paint, heat, lava and moss maps	`Locationmap File`, `Spawnmap File`, `Vegetationmap File`, `Paintmap File`, `Heatmap File`, `Lavamap File`, `Mossmap File`	`[06 BetterContinents.Maps]`
Alt-biome map	`Altbiomemap File`	`[08 BetterContinents.AltBiomes]`

When `Directory` is set, the file settings are not used.

A heightmap's fine picture is looked for beside the heightmap's file, under its name with `-fine` before the extension: `heightmap-fine.png` in a `Directory`, and `iceland-fine.png` beside a `Heightmap File` named `iceland.png`.

Legends

A colour map has a legend: a text file with the picture's name and `.txt` beside it (`biomemap.txt` beside `biomemap.png`), saying what each colour means. Each kind of map has its own legend format, explained in its chapter. If a biome, location, paint or terrain legend is missing, Better Continents writes the default one; a missing spawn or vegetation legend is written empty.

Size

A map is square. On a new world every map covers the same square, from −(World Size + Edge Size) to +(World Size + Edge Size) on each axis: 21 km across at the default World Size of 10,000 m and Edge Size of 500 m. North is the top of the picture. A bigger picture gives smaller pixels:

```
metres per pixel = 2 × (World Size + Edge Size) / pixels across
```

At the default sizes a 4,096 px map has pixels of 5.13 m, and an 8,192 px map 2.56 m. A 16,384 px map at 2 m a pixel covers 32.7 km (World Size 15,850 + Edge Size 500), and at 4 m a pixel 65.5 km.

A map can be up to 16,384 × 16,384 pixels. A bigger picture is refused before it is read, with a line in the log that says so. `Max Map Size` in `[07 BetterContinents.Misc]` (4096, 8192 or 16384; default 16384) lowers that limit for map files, for example on a machine with little memory; a map refused by it names the setting. Maps a world has already stored are always read.

Format

Save every map as a PNG.

Maps	Picture
Heightmap	16-bit grey. One 16-bit step is 0.003 m at Heightmap Amount 1; an 8-bit step is 0.78 m, which shows as terraces on gentle ground.
Roughmap, forest, heat, lava and moss maps	Grey, 8- or 16-bit.
Biome, alt-biome, terrain, paint, location, spawn and vegetation maps	Colour: RGB, or RGBA where the chapter says so.

Save PNGs plainly: not interlaced, in 8- or 16-bit grey, grey with alpha, RGB or RGBA at 8 bits a channel, or a palette. Better Continents reads such a picture a few rows at a time, with little memory. Any other

picture is read whole, which takes about 1 GB for a 16,384 px colour picture; the log says so for pictures of 4,096 px or more.

Memory and compact maps

Better Continents keeps a map in memory as tiles of 128 × 128 pixels, and a tile of one value, such as open sea, as that one value. Every pixel reads exactly as it was drawn.

`Compact Maps` in `[07 BetterContinents.Misc]` decides whether a new world keeps its maps compressed, in memory, in its world file and in what it sends to joining players. A compact world keeps each tile compressed and decodes it only when the world reads it:

Value	A new world is made compact…
Auto (default)	when any of its maps is more than 8,192 px across, or its heightmap has fine heights. The location map and the alt-biome map don't count.
On	always, when it is made in the newest settings version (12)
Off	never, not even with 16,384 px maps (the log warns), and a `heightmap-fine.png` is not read

The setting is read only when a world is made: From Config, by `bc_import`, or from a `Directory` with an `export.cfg`. A world keeps the way it was made: a compact world stays compact. An older config's `true` reads as On and `false` as Auto, and BepInEx writes the name into the file at the next start.

Compact maps make a big difference to big worlds. Saved as pictures, each 16,384 px biome, spawn or vegetation map takes 268 MB in the world file and in every join; one world file with five 16,384 px maps came to 52.6 MB compact, against 352.5 MB as pictures.

The log says why a new world is compact, or warns when it isn't:

```
Compact Maps is Auto and Heightmap <file> is 16383 x 16383 pixels (more than 8192 across), so
  this world is compact. ...
Compact Maps is on for this world: its heightmap has fine heights (heightmap-fine.png), which
  a world keeps as compressed tiles, and so it does its maps.
Compact Maps is Off, so this world keeps its maps as pictures although <map> <file> is W x H
  pixels (more than 8192 across). ...
```

In a compact world, `bc info` prints a line that starts `Compact Maps: the maps are held and saved as compressed tiles`.

`Map Memory` in `[07 BetterContinents.Misc]` (default 512 MB) is how much decoded map data a machine keeps for compact worlds. Past it, the tiles read least recently are dropped and decoded again when they are needed. One 16-bit 16,384 px map takes 537 MB decoded, so for a 16k world on a server with memory to spare, use 1024 to 2048.

> **Warning:** Better Continents 0.9 cannot read a compact world: it switches Better Continents off for it, and the world gets vanilla terrain. Versions 0.10.0 and later read compact worlds, except a world with fine heights, which needs 0.10.3 or later.

Changing a map after the world is made

A world stores its maps when it is made. Editing the picture afterwards does not change the world. To use the new picture, reload it in Debug Mode:

> **Console**
>
> Command: `bc reload <map>`
>
> Reads the map's picture again and uses it for the loaded world. `<map>` is `hm` (heightmap), `rm` (roughmap), `bm` (biome map), `ab` (alt-biome map), `terrain`, `paint`, `heat`, `lava`, `moss`, `fom` (forest

> map), `vegetation`, `lm` (location map), `spawn`, or `all` for every map.

Better Continents then regenerates the zones that are already made, except the ones players have worked on. See Changing an existing world for what is kept and how.

Tools

GIMP and Krita are free and both save 16-bit PNGs; turn interlacing off when you export. Better Continents ships a palette with every biome and alt-biome colour (`BetterContinents.gpl`) and a printable colour chart (`altbiome-palette.png`) in its package.

For a heightmap, soft brushes work better than hard ones: sharp edges show as cliffs in the game.

Next

- Heights and the heightmap: the most important map.
- Biomes and the biome map.

6 Heights and the heightmap

A heightmap is a greyscale image where white is high and black is low. It gives you direct control over terrain elevation across your whole world.

The height formula and the waterline

With a heightmap loaded and Heightmap Override All on (the default), Better Continents calculates height like this:

```
finalHeight        = heightmapValue × Amount − 0.15 + SeaLevelAdjustment
SeaLevelAdjustment = Lerp(1, −1, Sea Level)
world height (m)   = finalHeight × 200
```

At the default Sea Level Adjustment of 0.5 (which sets SeaLevelAdjustment to 0), this simplifies to:

```
metres above water = (value × Amount − 0.30) × 200
waterline pixel    = 0.30 / Amount
```

The **sea is at 30 m** on the game's height scale (finalHeight 0.15). Everything below 30 m is under water.

Choosing an Amount

`Heightmap Amount` in `[02 BetterContinents.Heightmap]` (default 1, range 0 to 81) is the multiplier that sets how high your terrain can be. Choose it before you draw any height, and keep it consistent: a map drawn for one Amount gives the wrong heights at another.

At the default Sea Level Adjustment of 0.5, black is −30 m on the game's height scale (60 m below the water) at every Amount, and white is as high as the table says. One 16-bit step is 200 × Amount / 65,535 m.

Amount	White, on the game's scale (sea at 30 m)	White, above the water	One 16-bit step	Use when
1	170 m	140 m	0.003 m	Lowland and island worlds
2	370 m	340 m	0.006 m	Worlds that keep every vanilla mountain location
5	970 m	940 m	0.015 m	Ground up to about 900 m
10	1,970 m	1,940 m	0.03 m	Ground up to about 2 km
20	3,970 m	3,940 m	0.06 m	Ground up to about 4 km
40	7,970 m	7,940 m	0.12 m	Ground up to about 8 km
81	16,170 m	16,140 m	0.25 m	Ground up to about 16 km, the most

Use the smallest Amount that holds your highest ground: the steps grow with the Amount, and at high Amounts they show as stairs on gentle slopes (see Fine heights).

Working with heights at Amount 1 (default)

At Amount 1 with default Sea Level Adjustment:

Heightmap value	8-bit grey	Altitude above water	What it is
0.00	0	−60 m	Deepest seabed
0.15	38	−30 m	Below the waterline
0.30	77	0 m	Waterline: the shore
0.40	102	20 m	Low ground

Heightmap value	8-bit grey	Altitude above water	What it is
0.50	128	40 m	Meadows and swamp
0.75	191	90 m	Hills
1.00	255	140 m	Peak (the maximum)

> **Mistake to avoid:** A waterline of 0.15 puts every lowland biome 30 m under water. The correct waterline is 0.30 for Amount 1.

Working with heights at Amount 2

Use Amount 2 when your world needs ground higher than 140 m above the water. Some of the game's mountain locations do: Hildir's Howling Cavern wants ground 200 m above the water, and Moder's altar and the two mountain tower ruins 150 m. At Amount 1 the game cannot place them (on one test map Hildir's cavern placed 0 of 3, so her mountain quest could not be done), and the other mountain sites, such as the frost caves, drake nests and mountain runestones (100 m), crowd into the top fifth of the range. At Amount 2 every vanilla mountain location fits.

At Amount 2 with default Sea Level Adjustment:

Heightmap value	8-bit grey	Altitude above water	What it is
0.00	0	−60 m	Deepest seabed
0.15	38	0 m	Waterline: the shore
0.30	77	60 m	Low ground
0.50	128	140 m	High ground
0.75	191	240 m	High mountains
1.00	255	340 m	Peak (the maximum)

> **In 16-bit precision:** One 16-bit step of a heightmap is `200 × Amount / 65,535 = 0.003 m` at Amount 1 and `0.006 m` at Amount 2. Save maps as 16-bit PNG. An 8-bit heightmap gives steps of about 0.78 m or 1.57 m and will show terraces on gentle ground.

Amounts above 5: high-world rules

The game's own height rules were made for land under about 400 m. When a heightmap is read at an Amount above 5, the most older versions allowed, Better Continents adjusts them for terrain that is much higher (with `High Terrain` in `[02 BetterContinents.Heightmap]` on Auto, the default):

- **Inside a dungeon** now means more than 3,000 m over the ground below, not just higher than 3,000 m. Terrain, weather and building keep working on a mountain, and dungeons, which stand 5,000 m above their entrance, still count as inside.
- **The ground is found at any height:** logging in at the logout point, items, tombstones, a zone's plants, creature spawn points and grass. Before, the grass found no ground above 500 m, and the rest none above 2,000 to 6,000 m.
- **Plants, creatures, locations and grass may stand at any height.** Their upper altitude limits of 1,000 m or more are lifted; lower ones, such as the tree line at 280 m, stay.
- **Parts of locations stay on** at any height (things that switched off over 10,000 m before).
- **The AI can walk** at any height.
- **Weather follows the player** on the ground: random events and the Deep North weather.
- **A new player's Valkyrie starts** over the start point.

`High Terrain` decides which worlds get these rules:

Value	Behaviour
Auto (default)	The rules are on for a world whose heightmap is read at an Amount above 5, and off for every other world.
On	The rules are on for every Better Continents world, whatever its heights. Without a heightmap, the limits are lifted to 1,908 m, a bound on the game's own terrain.
Off	The rules are off for every world, whatever its Amount.

Worlds read at an Amount of 5 or less keep the game's own rules under Auto, so older worlds play exactly as before.

Off costs a world with high ground: no grass above 500 m, no plants, creatures or locations above about 1,000 m, no terrain, weather or building above 3,000 m (the game takes it for the inside of a dungeon), and no ground found above 6,000 m.

A new world takes `High Terrain` when it is made (From Config, `bc_import`, or a `Directory` with an `export.cfg`) and keeps it. The world stores it only when it is not Auto, and a world export writes `High Terrain = On` or `High Terrain = Off` into `export.cfg` only then.

> **Console**
>
> Command: `bc h ht [auto|on|off]`
>
> Changes the loaded world's High Terrain, in Debug Mode on the machine that runs the world. The game's rules switch at once and the zones regenerate; the loaded terrain, the noise and the minimap stay as they are. In a world whose land can pass 9,000 m the alt biomes are placed again too, as for any alt-biome change. Players already connected keep the mode they joined with until they join again. The mode is read by its name, in any case.

The log says what the world gets when it loads:

```
High terrain: this world's heightmap is read at Heightmap Amount 20, above the 5 that older
  versions allowed, so its land can reach 3970 m: ...
High terrain: High Terrain is On for this world, so its land can reach <metres> m at most: ...
High terrain: High Terrain is Off for this world, whose heightmap is read at Heightmap Amount
  20 (its land can reach 3970 m): the game's own height rules stand, ...
```

The last line appears only for a world that Auto would have given the rules. `bc info` prints the same: a line `High terrain: the land can reach ... m` for an Auto world that gets the rules, and `High Terrain On: ...` or `High Terrain Off: ...` for a world that chose a mode.

> **Warning:** An older Better Continents opens a world read at an Amount above 5 with its terrain but without these rules. Better Continents 0.10.2 and older stop at a world whose High Terrain is On or Off (*Unknown feature* in the log) and treat it as a world without Better Continents: it gets vanilla terrain. Back up the world before you change versions.

Things the game keeps unchanged at any height:

- **Fog and distance** stay the game's own. Fog hides terrain beyond roughly 0.6 to 1.4 km on a clear day, and distant terrain is drawn only out to 1.2 km.
- **Clouds stay at 24 to 424 m** on the game's height scale, with the horizon cloud band up to 2,000 m. From a high summit you look down on them.
- **Munin does not land** on ground over 2,000 m.
- **The map shading** comes in steps of 8 m over 8 km and 16 m over 16 km.

- **The Deep North shader** is a fixed circle at (0, −4,000) with a radius of about 12 km. Outside it, farmland is not drawn as soil, mountain ground is raised as deep snow (2.1 m) and the shore band sits 2 m lower.

Other heightmap settings

Heightmap Blend and Heightmap Add

`Heightmap Blend` in `[02 BetterContinents.Heightmap]` (default 1, range 0 to 1) blends the heightmap against flat ground 60 m below the water. At 1 (the default), the heightmap's full range shows. At 0, you have flat ground; at 0.5, the heightmap is half as high.

`Heightmap Add` (default 0) is how strongly the heightmap is added to the result instead of blended into it. You usually want Blend, so leave Add at 0.

Heightmap Mask

`Heightmap Mask` (default 0, range 0 to 1) limits the maximum height to the heightmap's height. Used with Heightmap Override All off to keep ground features from the game's biomes under the heightmap's ceiling.

Heightmap Override All

`Heightmap Override All` (default on) makes the world conform exactly to your heightmap. When it is on, the game's own biome terrain (mountain peaks, swamp flats, rivers) is bypassed.

When it is off, the game builds its own biome terrain on top of your heightmap. The Mountain biome doubles whatever is over 80 m: a 1,000 m peak in your heightmap becomes about 1,900 m high, and the game's own terrain adds 0 to 40 m of noise plus about 4 m for each unit of slope. This is for mixing worlds: keeping the game's feel while your heightmap sets the general shape.

Heightmap Alpha

`Heightmap Alpha` in `[02 BetterContinents.Heightmap]` (default off) lets you blend your heightmap with the game's own terrain, using the PNG's alpha channel as a blend mask.

Save your heightmap as a 16-bit greyscale PNG with an alpha channel (16-bit alpha). Where a pixel is opaque, the world uses your heightmap's height; where it is transparent, the game's terrain shows; in between, they blend. So your heightmap can shape one region and leave the rest to the game.

Terrain records in maps

A world export records the `Heightmap Amount` and `Sea Level Adjustment` its heights are encoded for, as a PNG text chunk named `BetterContinents` inside the heightmap file:

```
Heightmap Amount = 2; Sea Level Adjustment = 0.5
```

Image editors display this text and usually preserve it. When a new world reads such a heightmap at different values, the log shows a warning naming both pairs, and `bc info` shows the record.

A heightmap you draw yourself has no record. It is read at the config's Amount and Sea Level exactly as before.

Fine heights

When terrain gets tall enough that 16-bit steps show as visible stairs (0.25 m at Amount 81), use fine heights to refine them further.

> **Warning:** A world made with fine heights needs Better Continents 0.10.3 or later. Versions 0.10.0 to 0.10.2 cannot read it. They log *Failed to load settings from ...: a map saved in format 2, which this version of Better Continents cannot read*, switch Better Continents off for the world, which then gets vanilla terrain, and leave its settings file alone. Keep such a world away from older versions, and back up before you change versions.

Put a second picture beside your heightmap, named by adding `-fine` before the extension: `heightmap.png` → `heightmap-fine.png`. It is 8-bit grey, the same size as the heightmap, and refines each height to 1/256 of a 16-bit step, 24 bits in all. Each byte is read as a signed number from −128 to 127 and added to the heightmap's value at a 256th of a step (Formulas has the exact rule). A byte of 0 changes nothing, and `heightmap.png` itself reads as it always did.

The fine picture can carry a record, a PNG text chunk named `BetterContinents`, that ties it to its heightmap:

```
Fine Format = 1; Fine Bits = 4; Heightmap CRC-32 = 2723AFF1
```

The CRC-32 is that of the `heightmap.png` file. If it doesn't match, the fine picture belongs to another heightmap and is not used. A fine picture whose record names no CRC is trusted. A world export writes this record for you (see Export and import): after you edit an exported `heightmap.png`, delete its `heightmap-fine.png` or export again.

When fine heights are read

The fine picture is read with the heightmap from its file:

- when a new world is made From Config, from a `Directory` (a world export's folder included) or from `Heightmap File`;
- by `bc_import`, whose preset then holds them;
- in Debug Mode, by `bc h fn`, `bc reload hm`, `bc reload all` and `bc h alpha`.

A joining player gets the world's own heights from the server, and a world that loads from its save uses what it stored: neither reads a fine picture.

The world must be one of the newest settings version (12), read its heightmap at full precision (not `Heightmap Alpha` as it was before 0.10), and have `Compact Maps` in `[07 BetterContinents.Misc]` on Auto or On. A world with fine heights keeps its maps as compressed tiles, so it is always compact (see Image maps). A 16-bit heightmap with an alpha channel, with `Heightmap Alpha` on in a version 12 world, takes fine heights too.

When the fine picture is used, the log says so, for example:

```
Fine heights: heightmap-fine.png refines heightmap.png with 4 bits a pixel: ground in steps of
  1.5 cm at Heightmap Amount 81 (heightmap.png alone: 24.7 cm), 5.7 MB of tiles, 0.17 bits a
  pixel. A Better Continents older than this one cannot read a world made with fine heights.
```

`bc info` shows a `Fine heights: ...` line under the heightmap. Measured on one machine, a dedicated server, with a 16,383 px heightmap at Amount 81: the heights came within 0.9 mm of the reference where the fine picture matters (up to 109 mm off with the heightmap alone), the world file grew by 5.7 MB, and making the world took 1.8 s longer.

A world saved with an `Override version` below 12 has no room for fine heights: it saves its heightmap without them, with one warning in the log.

Making a fine picture

The world export makes one for you: see Export and import. To make one yourself, 4 bits is usually enough: the bytes are then multiples of 16. Good practice is to keep the fine value at 0 under water and on steep ground, where a neighbouring pixel is a whole step or more away.

Why fine heights are not used

When a world reads a heightmap that has a `heightmap-fine.png` beside it and does not use it, the log says why, in a line that starts `Fine heights:`. The reasons, in the order they are checked:

1. There is no `heightmap-fine.png` beside the heightmap. Nothing is logged.
2. The world is of an older settings version, 11: *is not read: the world is saved in settings version 11, which has no fine heights*.
3. `Heightmap Alpha` as it was before 0.10 (Legacy) reads `heightmap.png` as 8-bit grey.
4. `Compact Maps` is Off: *is not read: Compact Maps is Off, and a world with fine heights keeps its maps as compressed tiles (Auto or On reads it)*.
5. `heightmap.png` cannot be read.
6. `heightmap.png` is not 16-bit grey (grey, or grey and alpha).
7. `heightmap-fine.png` cannot be read.
8. `heightmap-fine.png` is bigger than `Max Map Size` allows.
9. It is not 8-bit grey (no alpha, no palette).
10. It is a different size from `heightmap.png`.
11. Its record's Fine Format is not 1.
12. Its record names another CRC-32: *was made for another heightmap.png (its record names CRC-32 …; this one is …). If heightmap.png was edited, make heightmap-fine.png again or delete it*.
13. Its image data cannot be read.
14. Every byte is 0: *every byte of heightmap-fine.png is 0, so it changes nothing: the world is the one heightmap.png makes alone*.

In each case the world is made from `heightmap.png` alone.

The roughmap

The roughmap blends the heightmap with the game's own biome terrain when Heightmap Override All is off. It is a greyscale image the same size as your heightmap: white shows the game's biome terrain, black shows your heightmap smooth, and grey blends between them.

`Roughmap Blend` in `[02 BetterContinents.Heightmap]` (default 1, range 0 to 1) scales the roughmap's effect. At 0, only your heightmap shows.

Height layers

Height layers add procedural noise on top of your heightmap. See Worlds from noise for how to use them.

Editing and checking your heightmap

Before you play:

1. Check the waterline against your Amount: value 0.30 at Amount 1, 0.15 at Amount 2. Black is the deepest seabed, 60 m under the water.
2. Check that your lowland biomes (Meadows, Swamp, Plains) are drawn above the waterline. If almost none of a biome is, the heightmap was drawn for another Amount, and the world will come out drowned.
3. Unless `Mountains Allowed At Center` is on, keep high ground away from the world centre: the mod caps terrain within about 600 m of the centre at 46 m above water, blending out to 1,000 m.
4. With Heightmap Override All on (the default), rivers and lakes must be carved into your heightmap: the game's rivers are bypassed.

Save your heightmap as a plain, non-interlaced 16-bit grey PNG. Better Continents reads such a picture a few rows at a time; other formats are read whole, which takes far more memory for a big map (about 1 GB for a 16,384 px colour picture).

7 Biomes and the biome map

A biome map is a colour picture that says which biome is where: you paint each biome in its colour, and the world follows your layout instead of the game's rings. This chapter gives the colours and the legend, explains how pixels become biomes, and covers `Biome precision`, which sets how closely the ground follows your borders. It ends with the biomes that the Expand World Data mod adds.

The biome colours

Each biome has a colour, listed in the map's legend. These are the default colours, which Better Continents writes into a new legend:

Biome	Name in the legend	Hex	R, G, B
None (the game decides)	`None`	`000000`	0, 0, 0
Meadows	`Meadows`	`00FF00`	0, 255, 0
Black Forest	`BlackForest`	`007F00`	0, 127, 0
Swamp	`Swamp`	`7F7F00`	127, 127, 0
Mountain	`Mountain`	`FFFFFF`	255, 255, 255
Plains	`Plains`	`FFFF00`	255, 255, 0
Mistlands	`Mistlands`	`7F7F7F`	127, 127, 127
Ashlands	`AshLands`	`FF0000`	255, 0, 0
Deep North	`DeepNorth`	`00FFFF`	0, 255, 255
Ocean	`Ocean`	`0000FF`	0, 0, 255

Where a pixel is `None`, the game's own biome generation decides. Better Continents ships these colours, and those of the alt biomes, in a palette for GIMP and Krita, `BetterContinents.gpl` (see Image maps).

The legend

The legend is a text file beside the picture with the same name: `biomemap.txt` beside `biomemap.png`. If it is missing, Better Continents writes it with the default colours above. Each line is a name and a colour:

```
Meadows: 00FF00
BlackForest: 0,127,0
# a comment
```

- A colour is hex (`RRGGBB`) or `r,g,b` in decimal.
- Names are not case-sensitive, and spaces, hyphens and underscores in them are ignored: Black Forest, `black_forest` and `Deep-North` work too.
- Blank lines and lines that start with `#` are skipped.
- A line that is not `name: colour` is logged and skipped. A biome given a colour that doesn't read keeps its default colour; a name that is not a biome is logged, and its colour reads as None.
- A biome listed twice keeps its last colour, with a warning.

You can change any colour, for example to match a palette you already use. With Expand World Data installed, the legend also takes the names of the biomes it adds (below).

How pixels become biomes

Every pixel takes the biome of the legend colour nearest to it, measured over red, green and blue (alpha is ignored). When two legend colours are equally near, the one listed first wins. So a slightly-off colour still reads as the biome you meant.

The catch is soft edges. An anti-aliased edge between two biomes holds colours between them, and those can be nearer a third biome. The half-way blend of Meadows and Ocean, (0, 128, 128), is nearest to Mistlands, so a soft coastline between Meadows and Ocean comes out as a thin fringe of Mistlands.

> **Warning:** Paint biome borders with hard edges: no anti-aliasing, no soft brushes, no smoothing when you scale the picture.

A point in the world falls between pixels. Better Continents weighs the four pixels around it by how close each is, and the biome with the most weight wins.

Leaving a biome out

A biome map doesn't need every biome. When it leaves one out, the game cannot place the locations that need that biome, and the log says so once:

```
The biome map has no Mistlands, so no ... exist for it. Locations that require Mistlands will
  be skipped.
```

That includes boss altars: a world with no Mistlands has no Mistlands boss. Check a new world with `bc locs`, which warns when a boss is missing (see Locations, creatures and the start).

The Ashlands and the Deep North

- **Deep North.** With a biome map, the Deep North's weather and how cultivated ground is painted there follow your Deep North pixels, not the game's fixed area in the north. Its calm sea stays in the game's area, where the water is drawn calm (see World size and big worlds).
- **Ashlands.** What counts as Ashlands (where lava burns and the Ashlands weather applies) follows the heat map when the world has one. With a biome map and no heat map, or `Heatmap Scale` 0, your Ashlands pixels decide instead. See Ground colour, heat, lava and moss.

Biome precision

The game builds the ground in zones of 64 m, and a zone takes its biomes from its 4 corners. So the ground follows your biome borders only on a 64 m grid, and a small patch of a biome can fall between the corners and vanish from the ground. `Biome precision` in `[03 BetterContinents.Biomemap]` fixes that: a value N splits each zone into (N + 1) × (N + 1) cells, each with its own corners.

Biome precision	Cells per zone	One cell
0 (the game's own, the default)	1 × 1	64 m
1	2 × 2	32 m
2	3 × 3	21 m
3	4 × 4	16 m
4	5 × 5	13 m
5	6 × 6	11 m
7	8 × 8	8 m
15	16 × 16	4 m
31	32 × 32	2 m

It moves the ground textures, the grass, the vegetation and the spawn points to follow your borders. It never changes the terrain's height, nor where the game places locations and alt biomes. It works with or without a biome map.

A new world takes 0 to 31. A world made by an older Better Continents (settings version 11) uses at most 5: a higher value is stored but read as 5.

A zone samples (N + 2) × (N + 2) points each time its ground is built: 1,089 at 31, against the 4 corners at 0. Raise it step by step and watch how the game runs. The log states the setting when a world loads, for example:

```
Biome precision 5: each 64 m terrain zone follows the biomes on 6 x 6 cells (10.7 m)
```

> **Console**
> Command: `bc b p <value>`
>
> Sets the loaded world's Biome precision, 0 to 31, in Debug Mode. It rebuilds the loaded ground and grass, and regenerates the zones. The world saves the new value.

Drawing a biome map

The map covers the same square as every other map, with north at the top (see Image maps). Save it as an RGB PNG.

- Keep the biome colours as swatches or a palette, so you pick exact colours every time.
- Put each biome on its own layer, filled with its colour, and paint the layer's mask instead of the colour. Then you can use Threshold on the mask directly, and reorder the layers.
- Put your heightmap on a layer underneath and set the biome layers to 50 % opacity, so you can see both.
- Make masks from the heightmap with Threshold: a land mask, a mountain mask, and a rough swamp mask.
- Take the water out of the swamp mask with the land mask. The swamp looks cleaner, and fewer creatures spawn in the water.
- To grow or shrink a mask, blur it and apply Threshold again.
- To roughen biome borders, use the smudge tool, then apply Threshold again to keep the edges hard.
- Keep the Meadows near the centre and the harder biomes outward: the game places many locations by their distance from the world centre.

The size and shape of each biome region also decide which alt biomes the game gives it: see Alt biomes.

Expand World Data

Better Continents works with the Expand World Data mod's custom biomes:

- **The legend takes its biome names**, as its `expand_biomes` yaml spells them, for example `DeadWastes: 8B4513`. Give each such biome a `terrain` in the yaml. A world export writes the names back.
- **Install Expand World Data wherever the world is played, with the same yaml**, and add new biomes at its end.

A new world (settings version 12) with Expand World Data installed also gets these:

- Its altitude rules (a biome's or territory's altitude multiplier and delta, water depth, height limits and lava dip) apply over the heights of your heightmap, as they do over the game's own terrain. Leave a biome's altitude settings at their defaults if you want your heightmap's heights exactly.
- With a biome map and no heat map (or `Heatmap Scale` 0), the biomes it marks `lava: true` are hot where the biome map puts them.
- A line of `terrainmap.txt` can name one of its biomes for a ground colour (see Ground colour, heat, lava and moss).
- With `Biome precision` above 0, its territories keep their ground colours.
- The world saves the names of the biomes its map holds, and follows them if Expand World Data numbers its biomes differently later. A biome it no longer has reads as None until it has it again.

See Other mods.

Settings and commands

Setting	Section	Default	Range	What it does
`Biomemap File`	`[03 BetterContinents.Biomemap]`	empty	a file	The biome map. Not used when `Directory` is set: its `biomemap.png` is
`Biome precision`	`[03 BetterContinents.Biomemap]`	0	0 to 31	How closely the ground follows the biome borders in each 64 m zone

Command	What it does
`bc b fn <file>`	Sets the loaded world's biome map, in Debug Mode
`bc b p <value>`	Sets its Biome precision
`bc reload bm`	Reads the biome map and its legend again from their files

A reloaded biome map replaces the world's only when both the picture and the legend read without errors; otherwise the world keeps its map, and the log says why.

What to watch for

- **Hard edges.** A soft edge reads as a third biome along the border.
- **The map is stored in the world.** Editing `biomemap.png` or `biomemap.txt` changes nothing until you make a new world, or reload the map with `bc reload bm` in Debug Mode.
- **Zones keep what they got.** Vegetation is placed when a zone is first made. After a `bc` change, Better Continents regenerates the zones, except the ones around what players built or worked on (see Changing an existing world).

Next

- Alt biomes: plant the variants of each biome with a second colour map.
- Heights and the heightmap: the heights your biomes stand on.
- Image maps: sizes, formats and tools.

8 Alt biomes

Valheim 1.0 gives parts of each biome an *alt biome*: Dark Meadows, Troll Black Forest, Fortress Mountain and 29 more, each changing the creatures, vegetation, locations or weather of a region. The game picks them at random every time a world loads. With Better Continents you can plant the alt biomes you want with colours, the way the biome map plants biomes, and steer the game's random placement on the rest of the world. This chapter explains how the game places them, how to paint them, and the settings and commands that control them.

What alt biomes are

There are 32 alt biomes. An alt biome can add or remove creatures, add or remove vegetation, add or remove locations, remove weather types, and raise the chance that creatures spawn at a higher level. None of them changes the biome name shown in the game: a Dark Meadows region still says Meadows. A region can carry several alt biomes, as long as they are not incompatible with each other.

How many each biome can have, counting the ones shared between biomes: Meadows 9, Black Forest 9, Mistlands 7, Swamp 6, Plains 6, Mountain 4 and Deep North 1 (Lantern). The Ashlands and the Ocean have none.

How the game places them

Every time a world loads, on the server and on every player's machine, the game does this:

1. **The grid.** It samples the world on a grid of 2,048 × 2,048 points, 12 m apart, centred on the world centre. Each point records its biome and its height. Points more than 10,500 m from the centre count as Ocean. (A world laid out to its World Size has a grid of its own size: see the end of this chapter.)
2. **Regions.** Every 4-connected group of points of one biome (neighbours left, right, up and down) becomes a region. The Ashlands, the Deep North and the Ocean are different: each is **one** region for the whole world, however many separate patches it has.
3. **Measuring.** For each region the game records its edge count (the points on its border; about 0.47 × the radius in metres for a round region), its centre (the mean of its edge points), its distance from the world centre, and its height: the midpoint of the lowest and highest heights on its edge.
4. **The rules.** A region can get an alt biome only if its distance from the centre is at least the alt biome's minimum, its edge count is inside the alt biome's window, its height is at least 30 m (the water's surface), and it carries no alt biome that is incompatible with this one.
5. **Random placement.** For each alt biome, the game shuffles its biome's regions with a seed made from the world seed and goes through them. While fewer than the alt biome's minimum have it, every region that passes the rules gets it. After that, each passing region gets it with the alt biome's chance, until the maximum is reached. An alt biome of several biomes has one count for all of them.

The same maps and seed always give the same result.

In Valheim 1.0 this grid decides more than alt biomes: every terrain zone takes the biomes at its corners from it, so it also decides the biomes that the ground, vegetation and spawns follow. Better Continents rebuilds it whenever a bc change alters the maps or the settings.

What this means for map makers

- **Islands and mid-sized patches get alt biomes.** Most alt biomes want an edge count of about 66 to 200: a round region of about 150 to 425 m radius, about 30 to 80 px across on a 2,048 px map of the default world. The Mountain ones want larger regions, about 425 to 640 m. Kalhygge Black Forest wants small ones, about 45 to 175 m.
- **Huge regions get none.** A continent-sized Meadows region has an edge count far above every Meadows window. Holes count too: the rim of a lake painted as Ocean, or of another biome inside a region, adds to its edge count.
- **Regions around the centre get none.** The distance is measured from the mean of the edge points. A ring of Meadows around the spawn has its mean near the centre and fails every minimum distance (500 to 2,000 m).
- **Edges must stand above the water.** The biome map, not the coastline, decides where a region ends. A land biome painted out into the sea has edge points under water, which pull its height below 30 m. `Mean Sector Height` helps here (see Steering the random placement).
- **Ragged outlines and stray pixels cost you.** Noisy borders add edge points and can push a region past its maximum. Stray pixels of another biome make tiny regions of their own.

When the game cannot place an alt biome's minimum, its log says so. `bc ab info` shows why each alt biome did or did not place.

The 32 alt biomes

Count is the game's minimum and maximum number of regions per world. *Edge count* is the window a region's edge count must fall in (both ends included). *Incompatible with* works both ways.

#	Alt biome	Plants on	Count	Chance	Min. distance (m)	Edge count	Incompatible with
0	Mushroom	Meadows, Black Forest	1–3	0.2	500	66–198	
1	Lantern	Meadows, Black Forest, Swamp, Mountain, Plains, Mistlands, Deep North	3–5	0.2	500	66–198	
2	Bones	Swamp, Plains, Mistlands	1–3	0.2	2,000	66–198	Peaceful Meadows
3	Menhir	Meadows, Plains	1–3	0.2	750	33–198	Birch Meadows
4	Dark Meadows	Meadows	2–5	0.5	1,900	66–298	Peaceful Meadows
5	Peaceful Meadows	Meadows	1–2	0.2	1,000	99–298	Bones, Dark Meadows
6	Dandelion Meadows	Meadows	1–3	0.2	500	66–198	
7	Raspberry Meadows	Meadows	1–3	0.2	500	66–198	
8	Smalltree Meadows	Meadows	1–3	0.2	500	33–198	
9	Birch Meadows	Meadows	1–3	0.2	500	66–198	Menhir
10	Troll Black Forest	Black Forest	1–3	0.2	750	66–198	
11	Root Black Forest	Black Forest	1–4	0.2	2,000	99–298	Rock Black Forest, Kalhygge Black Forest
12	Ruin Black Forest	Black Forest	1–5	0.2	500	33–198	
13	Rock Black Forest	Black Forest	1–5	0.2	750	33–198	Root Black Forest, Pinetree Black Forest
14	Pinetree Black Forest	Black Forest	1–5	0.2	750	33–198	Rock Black Forest
15	Blueberry Black Forest	Black Forest	1–5	0.2	500	33–198	
16	Hut Swamp	Swamp	1–3	0.2	1,500	33–198	Bog Swamp

#	Alt biome	Plants on	Count	Chance	Min. distance (m)	Edge count	Incompatible with
17	Bog Swamp	Swamp	1–5	0.2	1,500	33–144	Hut Swamp
18	Bat Swamp	Swamp	1–3	0.2	2,000	33–144	
19	Abomination Swamp	Swamp	1–3	0.2	2,000	66–144	
20	Wolf Mountain	Mountain	1–5	0.2	2,000	199–298	Drake Mountain, Fortress Mountain
21	Drake Mountain	Mountain	1–3	0.2	2,000	199–298	Wolf Mountain, Fortress Mountain
22	Fortress Mountain	Mountain	1–2	0.2	2,000	199–298	Wolf Mountain, Drake Mountain
23	Lox Plains	Plains	0–1	0.2	2,000	99–298	Goblin Plains, Death Plains
24	Goblin Plains	Plains	0–2	0.2	2,000	66–198	Lox Plains, Death Plains
25	Death Plains	Plains	0–1	0.2	1,000	99–298	Lox Plains, Goblin Plains
26	Kalhygge Black Forest	Black Forest	1–2	0.2	500	20–79	Root Black Forest
27	Rockless Mistlands	Mistlands	2–5	0.2	2,000	99–598	
28	Trees Mistlands	Mistlands	1–5	0.2	2,000	59–298	
29	Swords Mistlands	Mistlands	1–3	0.2	2,000	59–198	
30	Hare Mistlands	Mistlands	1–3	0.2	2,000	59–198	
31	BroodSwarm Mistlands	Mistlands	1–3	0.2	2,000	59–198	

All 32 need a height of 30 m or more and are enabled in the game.

What each one changes

Creatures and added vegetation are listed by prefab name; removed vegetation by the game's vegetation entry name. The number after an added location is its quantity.

#	Alt biome	What it changes
0	Mushroom	Adds vegetation Pickable_Mushroom
1	Lantern	Adds vegetation deepnorth_lantern_standing
2	Bones	Adds creatures Skeleton_Poison; adds vegetation LargeBone, LargeBone_half01, LargeBone_half02
3	Menhir	Adds creatures Skeleton, Skeleton_Mountains; adds vegetation Rock_7_meadows
4	Dark Meadows	Adds creatures Draugr; adds locations WoodVillage2 (15); removes locations WoodFarm1; removes weather Clear, LightRain; creature level-up chance × 2
5	Peaceful Meadows	Adds creatures Boar, Deer; removes creatures Greyling, Greydwarf, Skeleton, Draugr, Troll; adds vegetation FireFlies
6	Dandelion Meadows	Adds vegetation Pickable_Dandelion
7	Raspberry Meadows	Adds vegetation RaspberryBush
8	Smalltree Meadows	Adds creatures Neck; adds vegetation Beech_small1, Beech_small2, stubbe, Bush01, Beech1; removes vegetation beech1_meadows, birch1_meadows, birch2_meadows
9	Birch Meadows	Adds creatures Deer; adds vegetation Birch1, Birch2, Beech1, Oak1; removes vegetation beech1_meadows
10	Troll Black Forest	Adds creatures Troll
11	Root Black Forest	Adds creatures TentaRoot_wild, Greydwarf_Elite; removes creatures Skeleton; adds vegetation Greydwarf_Root, Greydwarf_Surprise

#	Alt biome	What it changes
12	Ruin Black Forest	Adds vegetation stone_wall_2x1_ruin, stone_wall_1x1_ruin, stone_wall_ruin, stone_wall_ruin_2, GhostSkull; adds locations StoneTowerRuins07_sunk, StoneTowerRuins08_sunk, StoneTowerRuins09_sunk, StoneTowerRuins10_sunk (10 each)
13	Rock Black Forest	Adds vegetation rock4_forest, Rock_3, Rock_4, FirTree_small, stubbe, FirTree; removes vegetation firtree_many_blackforest, Pinetree_01_blackforest, firtree_blackforest; removes locations TrollCave02; creature level-up chance × 2
14	Pinetree Black Forest	Adds vegetation Pinetree_01; removes vegetation firtree_blackforest, firtree_many_blackforest, firtree_small_blackforest; creature level-up chance × 2
15	Blueberry Black Forest	Adds vegetation BlueberryBush
16	Hut Swamp	Adds vegetation SwampTree1, SwampTree2, wooden_path, wood_pole_log_4_worn; removes vegetation swamptree1, swamptree2; adds locations SwampHut1_1, SwampHut2_1, SwampHut3_1 (50 each)
17	Bog Swamp	Adds creatures Abomination; adds vegetation StatueEvil, SwampTree2_log, FirTree_small_dead, SwampTree1, SwampTree2; removes vegetation swamptree1, swamptree2; removes locations InfestedTree01
18	Bat Swamp	Adds creatures Bat_Swamp
19	Abomination Swamp	Adds creatures Abomination
20	Wolf Mountain	Adds creatures Wolf; removes creatures Skeleton, Hatchling; adds vegetation FirTree; removes vegetation rock1_mountain, rock2_mountain; creature level-up chance × 2
21	Drake Mountain	Adds creatures Hatchling; removes creatures Wolf, Skeleton; adds vegetation rock3_mountain_1; removes vegetation firtree_mountain; creature level-up chance × 2
22	Fortress Mountain	Adds creatures Skeleton_Mountains, StoneGolem; adds vegetation stone_wall_ruin, stone_wall_ruin_2, rock3_mountain_1; removes vegetation firtree_mountain; adds locations StoneTowerRuins05_leet (10); removes locations StoneTowerRuins05; creature level-up chance × 3
23	Lox Plains	Adds creatures Lox; removes creatures Goblin; removes vegetation birch2_aut_heath; removes locations GoblinCamp2, StoneTower1, StoneTower3
24	Goblin Plains	Adds creatures Goblin; removes creatures Lox, Deathsquito; adds locations GoblinHut01 (30), GoblinHut02 (30), GoblinHut03 (20); creature level-up chance × 2
25	Death Plains	Adds creatures Deathsquito; removes creatures Lox, Goblin; removes vegetation birch2_aut_heath, heathrockpillar; adds locations TarPit1 (50), TarPit2 (20), TarPit3 (100); removes locations GoblinCamp2, StoneTower1, StoneTower3, StoneHenge1, StoneHenge2, StoneHenge3; creature level-up chance × 2
26	Kalhygge Black Forest	Adds vegetation stubbe, Pinetree_01_Stub; removes vegetation firtree_many_blackforest, firtree_blackforest, Pinetree_01_blackforest, firtree_small_blackforest
27	Rockless Mistlands	Replaces the game's cliff_mistlands1, cliff_mistlands2 and rock_mistlands1 entries with its own at another density; adds vegetation YggaShoot_small1
28	Trees Mistlands	Adds vegetation YggaShoot1, YggaShoot_small1, YggdrasilRoot
29	Swords Mistlands	Adds vegetation giant_sword1, giant_sword2, giant_helmet1, giant_helmet2
30	Hare Mistlands	Adds creatures Hare
31	BroodSwarm Mistlands	Adds creatures SeekerBrood

Added locations are placed when the world's locations are placed, when it is made. Vegetation is placed when a zone is first generated. Creature, weather and level-up changes apply at once. With `Skip Default Locations` on, the added locations are skipped along with the other default locations.

Planting alt biomes with colours

A second, optional colour map, the **alt-biome map** (`altbiomemap.png`), plants alt biomes where you paint them. Its **legend**, `altbiomemap.txt` beside it, says which alt biomes each colour plants. Black and transparent pixels are unplanted: the game's random placement decides there. Every planted area becomes a region of its own and gets the alt biomes its colour names.

Setting up the map

- Set `Altbiomemap File` in `[08 BetterContinents.AltBiomes]`, or put `altbiomemap.png` in your `Directory` folder.
- The picture is square, of any size up to 16,384 px, and covers the same square as every other map, north at the top (see Image maps). Save it as PNG: a lossy format shifts colours.
- If the legend is missing, Better Continents writes it with the default colours below.
- It works with or without a biome map, and doesn't change the biome map.
- Like every map, it is stored in the world when the world is made, and travels with the world, with presets and to players. Its colours are matched once, when it is stored, and every machine reads the same result. Editing the picture or the legend later changes nothing until you read them again with `bc reload ab` or `bc ab fn`.

The legend

One instruction per line:

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

Examples:

```
Dark Meadows: 2D4613                    plant Dark Meadows where the picture is #2D4613
Dark Meadows + Lantern: 0080C0          two alt biomes on one colour
Lantern: 0080C0                         the same colour on another line stacks the same way
*Mistlands: 7F3FBF                      every Mistlands alt biome that fits
!Dark Meadows: 7A8B9C                   force it, even on a Black Forest region
none: C030F0                            keep this area free of alt biomes
Troll Black Forest: at 2750, -1210      the whole region under (2750, -1210)
Bat Swamp: @ 3728, 7088                 short form
```

- **Names** are the game's alt-biome names, as in the tables of this chapter, and are not case-sensitive. A name that isn't one of the 32 is noted when the map is stored (fine if another mod adds it); one the game doesn't have when the world loads is warned about and plants nothing.
- `*` is a wildcard: `*Mistlands` plants every alt biome whose name matches and that fits the land. `*Plains` plants Lox Plains only: Goblin Plains and Death Plains are incompatible with it and are refused.
- `+` stacks alt biomes on one colour. Listing the same colour on several lines does the same.
- `!Name` forces that name past the content rules (below), and makes the colour claim every base biome.
- `none` cannot be combined with names: the `none` is then ignored, with a warning.
- **Colours** are `RRGGBB`, `#RRGGBB`, `RRGGBBAA`, `#RRGGBBAA`, `r,g,b` or `r,g,b,a`. Alpha is accepted and ignored. Black (`000000`) is kept for unplanted land and is refused.
- **Points** are `at x, z` or `@ x, z`, in world metres (x east, z north), with a point as the decimal mark.

- **Comments:** `#` starts a comment at the start of a line, or where it has a space on each side (or ends the line), so `Dark Meadows: #2D4613` is still a colour.
- A colour within 40 RGB units of another legend colour is warned about: pixels between them could go either way.
- A legend holds at most 254 entries: colours plus point lines.

Which pixels are planted

The colours are matched over red, green and blue, by their distance:

A pixel…	is
with alpha below 128	unplanted
within 20 units of a legend colour, and nearer to it than to black	planted by that line (the nearest colour; on a tie, the first in the legend)
as near to black as to the nearest legend colour, or nearer	unplanted
within 20 units of neither a legend colour nor black	unplanted; the log lists the five most common such colours, each with the nearest legend colour and its distance

Then, at every world load, each 12 m grid point takes the result of the nearest pixel. Nothing is planted outside the picture or beyond the grid. A colour **claims** a point only if one of its alt biomes (one the game has and has enabled) belongs to the point's base biome, or a name is forced, or the colour is `none`. Points a colour does not claim stay unplanted and open to random placement. So painting past a biome's edge is harmless: paint Dark Meadows roughly over an island, and only the island's Meadows change.

The default colours

These are the colours Better Continents writes into a new legend, one per alt biome. Every pair of them, and every one and every biome colour of the biome map (black and white included), is more than 40 RGB units apart, so the 20-unit tolerance can never confuse two of them. The palette file `BetterContinents.gpl` holds them all.

Alt biome	Hex	R, G, B	Alt biome	Hex	R, G, B
Mushroom	FF969C	255, 150, 156	Hut Swamp	B1332D	177, 51, 45
Lantern	F2A300	242, 163, 0	Bog Swamp	6E2143	110, 33, 67
Bones	F8E1B6	248, 225, 182	Bat Swamp	382B41	56, 43, 65
Menhir	DDA776	221, 167, 118	Abomination Swamp	95B17E	149, 177, 126
Dark Meadows	2D4613	45, 70, 19	Wolf Mountain	93C7BC	147, 199, 188
Peaceful Meadows	ADFFDA	173, 255, 218	Drake Mountain	5AAAC2	90, 170, 194
Dandelion Meadows	E5C54B	229, 197, 75	Fortress Mountain	B697AF	182, 151, 175
Raspberry Meadows	C2185B	194, 24, 91	Lox Plains	CC605F	204, 96, 95
Smalltree Meadows	6EC03D	110, 192, 61	Goblin Plains	FF7622	255, 118, 34
Birch Meadows	CCFFAC	204, 255, 172	Death Plains	671E07	103, 30, 7
Troll Black Forest	2C7B76	44, 123, 118	Kalhygge Black Forest	A9AE34	169, 174, 52
Root Black Forest	6F5E32	111, 94, 50	Rockless Mistlands	D99FE8	217, 159, 232
Ruin Black Forest	9D6C4B	157, 108, 75	Trees Mistlands	472577	71, 37, 119
Rock Black Forest	5A5264	90, 82, 100	Swords Mistlands	8A7FDB	138, 127, 219
Pinetree Black Forest	004754	0, 71, 84	Hare Mistlands	D56EBB	213, 110, 187
Blueberry Black Forest	4A5CB6	74, 92, 182	BroodSwarm Mistlands	8E2F8E	142, 47, 142

`bc ab names` and `bc_altbiomes names` print the same list in the game.

What planting does

- **Each planted patch becomes a region of its own, never merged.** Each 4-connected patch of one colour inside one of the game's regions becomes a new region. Two patches of one colour in one game region are two regions, and a patch never joins a neighbouring region. On the Ashlands, the Deep North and the Ocean, which are one world-wide region each, all patches of one colour form one region. What is left of a game region stays one region, measured again.
- **Random placement elsewhere is unchanged.** The alt biomes you didn't plant land where they would have without the planting, except in the regions the planting touched and where the quota rule applies.
- **Point lines split nothing.** `Name: at x, z` applies to the whole region under that position.
- **Planting ignores the game's occurrence rules:** region size, distance from the centre, height, world position bounds, neighbour rules and chance. You chose the place.
- **Planting keeps three content rules:** an alt biome the game has disabled is not planted; an alt biome is planted only on the base biomes it belongs to; and an alt biome incompatible with one already on the region is not planted, first come first served in legend order. The disabled and incompatible cases are logged as warnings. `!Name` overrides all three for that name.

The quota rule

Planting happens first, and the game counts every region an alt biome already has when it decides whether to place more. So planted regions count toward the game's minimum and maximum. Planting itself has no cap: plant three Fortress Mountain regions and all three get it, and the game then places none.

Planted	Fortress Mountain (1–2)	Lox Plains (0–1)	Dark Meadows (2–5)
0	at least 1, at most 2 random	at most 1 random	at least 2, at most 5 random
1	at most 1 more random	none	at least 1, at most 4 more random
2	none	none	at most 3 more random
3	none (all 3 planted)	none	at most 2 more random

"At least" holds only where enough regions pass the game's rules. `Amount Multiplier` and the `min` and `max` overrides change these numbers, and planted regions count against the changed ones. An alt biome of several biomes (Lantern, 3–5) counts them all together. On the Ashlands, the Deep North and the Ocean, one colour is one region and counts once.

When planting fails

Zones generated with the wrong alt biomes are permanent, so **a planting error stops the world from loading**, with a clear message, instead of carrying on without the planting. An error is an alt-biome map in the world's settings that cannot be read, or a failure while sampling the map, splitting the regions or applying the planted alt biomes.

- A dedicated server logs the error and quits without saving the world. A host logs out without saving, and a player disconnects.
- In mode Off nothing is planted, so such a world still loads.
- Content problems are not errors: an unknown name, a disabled or incompatible alt biome, or a colour that matches no legend line is logged as a warning and not planted.
- The world holds its own copy of the map, so editing the picture or the legend can't repair a world whose stored map is damaged. Restore the world's `BetterContinents` file from a backup: the copy from the save before ends in `.old`.

- During a live edit with the `bc` commands, a planting error keeps the previous regions and placement instead.

Modes

`Mode` in `[08 BetterContinents.AltBiomes]`, or `bc ab mode` in a loaded world:

Mode	Planted regions	The game's random placement
Random (default)	planted	on unplanted land, steered by the settings below
PlantedOnly	planted	none
Off	none; the map is not even sampled	none: no alt biome anywhere

A `none` region is protected in every mode. Off also clears alt biomes already assigned, and an unreadable alt-biome map doesn't stop the world from loading in that mode. Off means no alt biomes, not no grid: the game builds its grid and regions for every world. Without alt biomes, the locations they add, such as Dark Meadows' wood villages, have nowhere to go.

Steering the random placement

The settings in `[08 BetterContinents.AltBiomes]` are defaults for new worlds: a world stores them when it is made, and the `bc ab` commands change a loaded world's own copy. They steer only the game's random placement; planted regions ignore them. With every one at its default, the random placement is exactly the game's.

Setting	Default	Range	What it does	Console
`Fixed Seed`	empty	a number or any text	The seed of the random placement. Empty, or `world` in the console: the world seed. A whole number is used as it is; other text is turned into a number as a world seed's name is. The same map then gets the same random alt biomes whatever the world seed.	`bc ab seed`, `bc ab reroll`
`Chance Multiplier`	1	0 to 10	Multiplies every alt biome's chance (at most 1 after multiplying).	`bc ab chance`
`Amount Multiplier`	1	0 to 10	Multiplies every alt biome's minimum and maximum count, rounded to whole numbers.	`bc ab amount`
`Region Size Scale`	1	0.1 to 50	Multiplies every alt biome's edge count window. The game gives alt biomes only to regions about 0.1 to 2.4 km across; big hand-drawn regions usually need 2 to 10.	`bc ab regionscale`
`Distance Scale`	1	0 to 10	Multiplies every alt biome's minimum distance from the centre and its world position bounds.	`bc ab distancescale`
`Min Sector Thickness`	0	0 to 20	Regions thinner than this (area / edge count, in 12 m cells) never get a random alt biome. 0 is the game's own. 2 filters the slivers an anti-aliased biome map leaves along its borders.	`bc ab thickness`
`Mean Sector Height`	false	true, false	Measures a region's height as the mean over the whole region, instead of the midpoint of its lowest and highest edge heights. Helps maps whose biome paint runs out into the sea.	`bc ab meanheight`

Setting	Default	Range	What it does	Console
`Fix Neighbour Check`	false	true, false	Uses a corrected version of the game's required and forbidden neighbour test, which is broken. No alt biome of the game's uses it; ones from other mods may.	`bc ab neighbourfix`
`Overrides`	empty	see below	Per-alt-biome overrides of the random placement.	`bc ab set`, `bc ab clear`, `bc ab overrides`

Overrides

`Overrides` takes `Name: key=value, key=value; Other Name: key=value`. In a loaded world, `bc ab set <name> <field> <value>` sets one field (`default` clears it), `bc ab clear [name]` clears one entry's overrides or, without a name, all of them, and `bc ab overrides` prints them.

Key	Value	Replaces the alt biome's
`enabled`	true, false, 1, 0	enabled flag (random placement only)
`chance`	0 to 1	chance
`min`, `max`	0 to 10,000	minimum and maximum count
`mindist`	0 to 1,000,000 (m)	minimum distance from the centre
`minedge`, `maxedge`	0 to 100,000,000 (cells)	edge count window
`minheight`, `maxheight`	−100,000 to 100,000 (m)	height window
`ignorebounds`	true, false	world position bounds: true switches them off

- A value out of range is clamped. An empty value or `default` clears the field.
- A name may hold `*` wildcards, and `*` alone means every alt biome. Names are not case-sensitive.
- Field by field, the exact name wins over a pattern, a pattern with more literal characters over one with fewer, and a pattern over `*`.
- An override replaces the multiplied value: `chance=0.5` is 0.5 whatever `Chance Multiplier` says.
- If the maximum ends up below the minimum, it is raised to the minimum. A disabled alt biome gets a minimum of 0, so the game doesn't warn that it placed too few.
- Overrides never touch planted regions: `Fortress Mountain: enabled=false` stops the game placing Fortress Mountain at random, and planted Fortress Mountain still works.

Examples: `Fortress Mountain: enabled=false`, `*Mistlands: chance=0.5; Hare Mistlands: max=1`, `*: minedge=0, maxedge=100000`.

Grid

`Grid` (default WorldEdge, console `bc ab grid`) changes the grid itself, so it applies to planted and random alt biomes alike. With WorldEdge, when the edge of the world (World Size + Edge Size) is inside 10,500 m and `Map Edge Drop-off` is on, grid points beyond the edge count as outside-the-world ocean, so alt biomes, and the location candidates the game takes from the grid, cannot land beyond the edge. Vanilla always samples the 10,500 m disc. `Grid` has no effect when Expand World Size lays out the grid, nor on a world laid out to its World Size.

Commands

The `bc ab` commands are part of `bc`, a cheat command for the machine that runs the world (see Console commands). A change saves with the world and redoes only as much of the grid, regions and placement as it affects, then regenerates the zones. Connected players keep their placement until they reconnect. A value command without a value prints the current one.

Command	What it does
`bc ab info`	A summary of this world's alt biomes: why each did or did not place, what each legend line planted, and up to 40 planted regions.
`bc ab list [filter]`	Up to 60 regions with alt biomes. The filter is `planted`, `random`, `all`, a biome name, or part of an alt-biome name.
`bc ab here`	The region you stand in, what was planted on it, and how each alt biome of its biome fared.
`bc ab show [filter]`, `bc ab hide`	Pins every region with an alt biome (optionally only matching alt biomes), and removes those pins.
`bc ab export`	Writes `altbiomes-<time>.png` and a `.txt` report to `<save data>/BetterContinents/<world name>/`: the region grid, one pixel per grid point, north up, alt-biome regions tinted in their alt biome's colour, planted ones striped magenta, borders darkened.
`bc ab hash`	This machine's grid and placement hashes.
`bc ab names [filter]`	The game's alt biomes, with their default colours and random-placement rules.
`bc ab rebuild`	Rebuilds the grid, the regions and the placement from the current settings.
`bc ab fn [file]`	Sets the alt-biome map and stores it with its legend. To remove it, clear its field in the settings window.
`bc reload ab`	Reads the alt-biome map and its legend again from their files. `bc reload all` includes it.
`bc ab mode`, `grid`, `seed`, `chance`, `amount`, `regionscale`, `distancescale`, `thickness`, `meanheight`, `neighbourfix`	The settings in the tables above.
`bc ab set`, `bc ab clear`, `bc ab overrides`, `bc ab reroll [seed]`	The overrides, and a new fixed placement seed (random, or the one given).

Any player, on a client or a host, can use `bc_altbiomes`:

Command	What it does
`bc_altbiomes`	The hashes and whether they agree with the server's, and the region you stand in.
`bc_altbiomes here`	The region you stand in, in full.
`bc_altbiomes names [filter]`	The alt-biome names and their default colours. Works in the main menu too.
`bc_altbiomes hash`	The hashes and whether they agree with the server's.
`bc_altbiomes list [filter]`	Up to 200 regions with alt biomes. It reveals where they are, so on a client it needs devcommands, like the game's own `altbioms`.

Multiplayer

The alt-biome options and the stored map travel to players with the world's Better Continents settings. Each machine builds its own grid and regions; the server then sends its placement, and the player's game applies it, so every region both machines agree on carries the server's alt biomes, random ones included. Both sides compare two hashes, one of the grid and one of the placement, and log whether they match. A difference is a warning, never a kick: on a machine whose grid differs, the regions that still match get the server's alt biomes and the rest get none.

Log lines

When a world loads, Better Continents logs a summary like this (from a test world):

```
[BetterContinents] Alt biomes: split 7 planted region(s) out of 6 region(s) (0 fully planted
  and reused), 29175 of 4194304 grid points planted.
[BetterContinents] Alt biomes (world load): mode Random, placement seed world seed, grid 2048
  x 12 m to 10500 m (Vanilla), 1687 sectors, 14 with alt biomes (7 planted, 7 random), 8
  planted region(s)
[BetterContinents] Alt biomes: planted: 8 region(s), 7 alt biome(s) applied, 1 protected
  (none), 0 skipped for the base biome, 0 refused (disabled or incompatible)
[BetterContinents] Alt biomes:   #2D4613 [Dark Meadows] -> Dark Meadows: 4401 grid points in 1
  region(s)
[BetterContinents] Alt biomes:   point at 3728, 7088 [Bat Swamp] -> Bat Swamp: 1 region(s)
```

It goes on with a line per base biome, the biomes the map leaves out, and, for each alt biome, why it did or did not place ("+N planted (counted toward the maximum)" where planting counts). In multiplayer the server logs whether each player's hashes match, and so does the player.

A world laid out to its World Size

A new world (settings version 12) whose World Size or Edge Size is not the game's own has a grid of its own size, unless Expand World Size is installed, which lays out the grid itself. The points stay 12 m apart, and there are as many as cover the world:

```
points a side = 2 × ceil(1,024 × (World Size + Edge Size) / 10,500)
```

That is 3,190 for World Size + Edge Size of 16,350 m, 4,000 for 20,500 m, and 6,392 for 32,764 m. From 42,000 m the grid stops at 8,192 points a side, and the points grow apart instead. The grid reaches the edge of the world, so the alt biomes and the game's locations reach the whole world, and `Grid` has no effect.

The alt biomes' own rules are not scaled: their counts, minimum distances and edge count windows stay the game's. `Amount Multiplier`, `Distance Scale` and `Region Size Scale` change them.

A bigger grid costs memory and time each time the world loads, on the server and on every player's machine as it joins, and mode Off doesn't spare it. It takes about 0.2 GB at 16,350 m. At 32,764 m it took about 0.8 GB and 24 to 27 s, measured on one machine. See World size and big worlds.

What to watch for

- **The map is stored in the world.** Read it again with `bc reload ab` after editing the picture or the legend.
- **Zones keep what they got.** Vegetation is placed when a zone is first generated, and an alt biome's locations when the world is made. After a `bc` change Better Continents regenerates the zones, except those around what players built or worked on (see Changing an existing world). Plant before the world is explored.

- **Paint patches a few cells wide.** The grid is 12 m: a patch narrower than a cell or two can vanish or break into several regions.
- **One colour on the Ashlands, Deep North or Ocean is one region.** Two Deep North areas painted with the same colour are one region and count once toward the maximum. Use two colours to make two.
- **Planting chooses alt biomes, not biomes or terrain.** Paint biomes on the biome map.

Next

- Biomes and the biome map: lay out the base biomes first.
- Console commands: every bc ab command.
- Export and import: an export writes the alt-biome map of a world.

9 Ground colour, heat, lava and moss

This chapter covers the maps that change how the ground looks and behaves, without changing its height or its biome: the terrain map (which ground texture shows), the paint map (paint over the ground, such as paving and cultivated soil), the heat map (where the Ashlands' heat is), the lava map and the moss map. You need none of them for a working world; each one gives you control over one thing the game otherwise decides from the biome.

The terrain map

The ground texture normally follows the biome: grass in the Meadows, mud in the Swamp. The terrain map overrides it wherever it has a colour, so you can, for example, give a stretch of Plains the Meadows' grass.

Set it with `Terrainmap file` in `[03 BetterContinents.Biomemap]` (note the lower-case f), or put `terrainmap.png` in your `Directory`. Its legend is `terrainmap.txt` beside it. Each line names a ground and the colour that paints it:

Ground	Default colour	Ground	Default colour
`Default` (the biome's own)	`000000`	`Plains`	`FFFF00`
`Meadows`	`00FF00`	`Mistlands`	`7F7F7F`
`BlackForest`	`007F00`	`AshLands`	`FF0000`
`Swamp`	`7F7F00`	`DeepNorth`	`00FFFF`
`Mountain`	`FFFFFF`	`Ocean`	`0000FF`

If the legend is missing, or empty, these defaults are used, and a missing one is written out.

- The ground names are not case-sensitive. The colour is hex (`RRGGBB`) or `r,g,b`.
- A colour in the picture must match a legend colour exactly. **A colour that is in no line is used as it is**, as a raw ground colour (below). So keep the picture's colours exact.
- Where the picture is `Default` (black, with the default legend), the biome's own ground shows.

Raw ground colours

The game paints the ground by a hidden colour, and each ground has its own. A legend line can name such a colour instead of a ground, and a picture colour that is in no line is used as one. These are the game's grounds:

Ground	Raw colour (RGBA)
Meadows, Ocean	`00000000`
Black Forest	`0000FF00`
Swamp	`FF000000`
Mountain, Deep North	`00FF0000`
Plains	`000000FF`
Mistlands	`0000FFFF`
Ashlands	`FF0000FF`

Because a raw colour can need its alpha, save a terrain map that holds raw colours as an RGBA PNG.

Expand World Data grounds

On a new world (settings version 12) with Expand World Data installed, a line can name one of its biomes for the ground, for example `DeadWastes: FF8000`. Pixels of that colour get the ground colour Expand World Data gives that biome, and the log says:

```
Terrain map legend: DeadWastes is Expand World Data's biome, its ground colour ...
```

The world keeps the colour, not the name, so players and machines without Expand World Data read a plain colour. See Biomes and the biome map.

> **Console**
> Command: `bc terrain fn <file>`
>
> Sets the loaded world's terrain map, in Debug Mode. `bc reload terrain` reads the picture and legend again.

The paint map

Paint goes on top of the ground texture. It is what players change when they pave or cultivate. The Mistlands use the paint's alpha for their moss, and the Ashlands for their lava. The paint map sets the paint wherever you want.

Set it with `Paintmap File` in `[06 BetterContinents.Maps]`, or put `paintmap.png` in your `Directory`. Save it as an RGBA PNG.

Paint	Colour (RGBA)
Grass	`000000FF`
Dirt	`FF0000FF`
Cultivated	`00FF00FF`
Paved	`0000FFFF`
Patches	`00BF00FF`
Grass, dark	`996600FF`
Paved, moss	`000080FF`
Paved, dirt	`FF0080FF`
Paved, dark	`00FF80FF`
Moss (Mistlands)	depends on the alpha
Lava (Ashlands)	where the alpha is about 0.6 or more (hex 99)

Paints can be mixed, but the results are hard to predict: try them on a small area first.

Alpha, moss and lava

Each pixel's red, green and blue become the paint's. Its alpha works like this, in order:

1. In the Mistlands, if there is a moss map, the alpha comes from the moss map.
2. In the Ashlands, if there is a lava map, the alpha comes from the lava map.
3. Otherwise, a fully opaque pixel keeps the game's own alpha, so the game's own moss and lava show.
4. Otherwise, the alpha is the pixel's.

Players can't change the paint's alpha, so a paint map that sets alpha in other biomes affects how their terrain edits look there.

The paint legend

Some image editors make alpha hard to paint, so the legend, `paintmap.txt` beside the picture, can turn opaque picture colours into any paint. It starts empty. Each line is the paint, a colon, and the picture colour that paints it:

```
0000FFFF: 808080
```

That line turns mid-grey pixels into paved ground. Colours are hex or `r,g,b[,a]`. A picture colour in no line is used as it is. The first line for a picture colour wins. There are no comments in this legend, and a number that doesn't read loses the whole legend, with a line in the log.

> **Console**
> Command: `bc paint fn <file>`
>
> Sets the loaded world's paint map, in Debug Mode. `bc reload paint` reads the picture and legend again.

The heat map

In the Ashlands, heat makes the water boil, damages ships and sets players burning. In the game, heat rises as you go deeper into the Ashlands: most effects reach their full strength, a heat of 1, about 300 m in, and the effects for standing partly in water reach theirs, 10, about 3,000 m in. The heat map lets you put the heat where you want it.

Set it with `Heatmap File` in `[06 BetterContinents.Maps]`, or put `heatmap.png` in your `Directory`. Save it as 16-bit grey. `Heatmap Scale` in the same section (default 10, range 0 to 100) multiplies it: white is 10 by default. Since most effects stop growing at 1, a pixel of 0.1 (grey 26 of 255) already has the full effect at the default scale. A scale of 0 switches the heat map off.

> **Warning:** With a heat map, a spot counts as Ashlands wherever its heat is above 0. The heat map must be exactly 0 (pure black) everywhere that is not Ashlands: one stray grey pixel turns that ground into Ashlands, with its lava, weather and creatures.

Heat should still reach into the sea around the Ashlands, rising towards the coast: that is what burns ships on the way in, as in the game. Draw the heat as a shape around your Ashlands, fading out to sea, rather than tracing the biome's edge. The sea is drawn red wherever its heat is above 0, so the red water shows how far the heat reaches (see World size and big worlds).

Without a heat map, on a world with a biome map (or with `Heatmap Scale` 0), the biome map's Ashlands pixels decide what counts as Ashlands, so lava still burns there. On a new world with Expand World Data installed, its `lava: true` biomes count too.

> **Console**
> Command: `bc heat fn <file>`, `bc heat sc <value>`
>
> Set the loaded world's heat map and Heatmap Scale, in Debug Mode. `bc reload heat` reads the picture again.

The lava map

The game decides where the Ashlands ground is lava from a mask, and the lava map replaces that mask. It changes only the Ashlands, so you don't need to keep it black elsewhere.

Set it with `Lavamap File` in `[06 BetterContinents.Maps]`, or put `lavamap.png` in your `Directory`. Save it as 16-bit grey. Lava shows where the value is about 0.6 or more (grey 153 of 255).

Some Ashlands locations read the same mask when the game places them: the lava leviathans want 0.6 to 1, the charred fortress and the Ashlands boss's location at most 0.1, the Morgen holes at most 0.3, and the fortress ruins at most 0.8. So a lava map with lava everywhere leaves no room for the fortresses, and one with no lava leaves none for the lava leviathans.

> **Console**
> Command: `bc lava fn <file>`
>
> Sets the loaded world's lava map, in Debug Mode. `bc reload lava` reads the picture again.

The moss map

The moss map sets the Mistlands' ground cover. It changes only the Mistlands, so you don't need to keep it black elsewhere. Its grey value becomes the alpha of the ground paint there, which the Mistlands use for moss. Try it on a small area first, and check the result in the game.

Set it with `Mossmap File` in `[06 BetterContinents.Maps]`, or put `mossmap.png` in your `Directory`. Save it as 16-bit grey.

> **Console**
> Command: `bc moss fn <file>`
>
> Sets the loaded world's moss map, in Debug Mode. `bc reload moss` reads the picture again.

Settings

Setting	Section	Default	Range	What it does
`Terrainmap file`	`[03 BetterContinents.Biomemap]`	empty	a file	The terrain map: which ground texture shows
`Paintmap File`	`[06 BetterContinents.Maps]`	empty	a file	The paint map: paint over the ground
`Heatmap File`	`[06 BetterContinents.Maps]`	empty	a file	The heat map: the Ashlands' heat
`Heatmap Scale`	`[06 BetterContinents.Maps]`	10	0 to 100	What white in the heat map is worth; 0 switches it off
`Lavamap File`	`[06 BetterContinents.Maps]`	empty	a file	The lava map: where the Ashlands ground is lava
`Mossmap File`	`[06 BetterContinents.Maps]`	empty	a file	The moss map: the Mistlands' ground cover

The file settings are not used when `Directory` is set: its maps of the standard names are.

What to watch for

- **Exact colours.** The terrain and paint maps match colours exactly, and use an unknown colour as it is. Paint with hard edges, with no anti-aliasing or smoothing to shift a colour.
- **A stray pixel of heat is Ashlands.** Check the heat map is pure black away from the Ashlands.
- **The maps are stored in the world.** Editing a picture changes nothing until you make a new world, or reload it with `bc reload` in Debug Mode. See Image maps.

Next

- Forests and vegetation: forests, plants and the vegetation map.
- Biomes and the biome map: the biomes these maps sit on.

10 Forests and vegetation

This chapter covers where trees grow and what else grows where. Three settings shape the game's own forest; a forest map paints forests and clearings where you want them; and a vegetation map switches plants on or off by area. The end of the chapter covers doubled trees and rocks, and the `bc_twins` command that removes them.

How the game decides where forest grows

The game works out a *forest factor* from noise at every spot, and the vegetation it ties to that factor grows only where the factor says forest. That gives the pattern of forest patches and clearings you see in a vanilla world. Better Continents can change the size of the patches, how much of the land is forest, and which trees follow the factor, and a forest map can take over the pattern.

Forest Scale

`Forest Scale` in `[04 BetterContinents.Forest]` sets the size of the game's own forest and clearing patches. It doesn't change how much of the land is forest, and it doesn't scale a forest map.

Forest Scale	Size of the patches
0	about a third of the game's
0.25	about 0.64 × the game's
0.5 (the default)	the game's own
0.75	about 1.8 × the game's
1	about five times the game's

The setting takes 0 to 10, but keep it within 0 to 1. Near 1.15 the game's own forest is the same everywhere, and above that the patches shrink again.

> **Note:** A `BetterContinents.cfg` written by an older Better Continents may still hold `Forest Scale = 1`, which was the default up to 0.10.1. Better Continents sets it to 0.5 once, the first time the game starts with a newer version, and says so in the log. Write 1 there again if you want the larger patches. Worlds already made keep their forests.

Forest Amount

`Forest Amount` in `[04 BetterContinents.Forest]` (default 0.5, range 0 to 1) sets how much forest there is against clearings. 0.5 changes nothing; higher values give more forest, lower values more clearings.

Forest Factor Overrides All Trees

Some trees the game places whatever the forest factor says, so neither the factor nor a forest map moves them. `Forest Factor Overrides All Trees` in `[04 BetterContinents.Forest]` (default false) makes the forest factor, the game's own or a forest map's, decide these too:

FirTree, FirTree_small, FirTree_small_dead, FirTree_oldLog, Pinetree_01, SwampTree1, SwampTree2, SwampTree2_log, SwampTree2_darkland, HugeRoot1, and the spider webs vertical_web, horizontal_web and tunnel_web.

With it on, you can paint clearings into the Black Forest, the Swamp or the Mistlands.

The forest map

The forest map is a grey picture, `forestmap.png`, set with `Forestmap File` in `[04 BetterContinents.Forest]` or found in your `Directory`. Save it as 8- or 16-bit grey. It covers the same square as every other map (see Image maps).

Better Continents turns the game's forest factor into a 0-to-1 *forestness* (0 bare, 1 dense), combines it with the map's value, and turns the result back:

```
result = Lerp(forestness, forestness × map, Forestmap Multiply) + map × Forestmap Add
```

A spot is forest when the result is above about 0.41. What black means depends on the two settings:

Forestmap Multiply, Forestmap Add	The result	Black means	White means
1, 1 (the defaults)	map × (forestness + 1)	a clearing	forest
0, 1	forestness + map	the game's own forest	forest

- **With the defaults (1 and 1)**, black and anything below about 0.2 (grey 52 of 255) is a clearing, and 0.41 (grey 105) or more is forest everywhere. In between, the game's own pattern decides where the trees stand. Use this when black should mean open ground: heath, farmland, the slopes above a tree line.
- **With Multiply 0 and Add 1**, black leaves the game's own forest as it is, and a higher value adds trees, up to forest everywhere from about 0.41. Use this when the map only adds woods to the game's pattern.

In both cases, leave `Forest Amount` at 0.5 and `Forest Scale` at 0.5 unless you want to change the game's own part of the pattern.

> **Tip:** Don't hold the forest back to keep the minimap readable. The game draws its forest shading on the map only in the Meadows, the Plains and the Black Forest; the Swamp, the Mountains and the Deep North get none, and the Mistlands have their own mist shading. The shading follows the trees.

The vegetation map

The vegetation map switches plants on and off by area: trees, bushes, rocks and the other objects the game scatters. It is a colour picture, `vegetationmap.png`, set with `Vegetationmap File` in `[06 BetterContinents.Maps]` or found in your `Directory`, with a legend beside it, `vegetationmap.txt`. A missing legend is written empty.

- **Black** is the game's own vegetation: the map changes nothing there.
- **White** always means none: no vegetation.
- Any other colour must be in the legend exactly. A colour that isn't is logged once and treated as black.
- Save it as an RGB PNG of at least 329 × 329 px, which gives a pixel per 64 m zone on the default world. A larger picture gives more detail where you take plants away.

Each line of the legend is a colour, a colon, and entries separated by commas:

```
# Red: keep only beech trees and this rock
FF0000: none, Beech1, Rock4_forest
# Blue: let beeches grow here in any biome
0000FF: +Beech1
# Green: no plants whose names start with Rock
00FF00: -Rock*
```

Entry	Means
none	Switches off every plant here, except those named on the same line

Entry	Means
`+name`	Lets this plant grow here whatever the biome: its biome rule becomes "any biome", for the whole 64 m zone. Its other rules still apply
`-name`	Switches this plant off here, precisely, at each spot it would be placed
`name`	Exempts this plant from `none` on the same line, so it keeps its normal rules. On its own it changes nothing

- The colour is `RRGGBB` or `r,g,b` (a fourth number is alpha). Don't put `#` before it: a line that starts with `#` is a comment.
- A line needs exactly one colon; other lines are skipped. A colour number that doesn't read ends the legend at that line, with a line in the log.
- Names are the game's prefab names, not case-sensitive, and `*` is a wildcard: `Beech*`, `*_small`, `Rock*forest`. The fir is `FirTree`; the pine is `Pinetree_01`.

> **Console**
> Command: `bc vegetation fn <file>`
>
> Sets the loaded world's vegetation map, in Debug Mode. `bc reload vegetation` reads the picture and legend again.

Doubled vegetation and bc_twins

The game seeds each vegetation entry's random numbers with the world seed, the zone and the plant's name. Two entries for one plant therefore pick the same spots, and a zone filled a second time over its own objects picks every spot again: trees and rocks then stand in pairs, a copy within a metre of each one. A zone gets filled twice by a save put back together from mismatched files, or by a world tool's zone command. A `+name` entry on the vegetation map, or an alt biome's added vegetation, can make two entries for one plant meet.

The twin guard. In every Better Continents world, a plant is not placed within 1 m of the same kind of object that already stands in its zone, or that another vegetation entry placed there in the same pass. It changes vegetation only, never terrain, and only in zones generated from now on. An entry never blocks its own placements, so a plant with one entry keeps the game's density, except beside an object of its kind that already stands in its zone. Where two entries place the same plant, the second plant within 1 m of the first is dropped, so such a zone can end up with fewer plants.

`bc_twins` finds and removes the twins a world already has. It is not a cheat command: any player can count with it. Removal runs where the world is: in single player, on the host, or on a dedicated server through `bc_twins server ...` (admins only).

Command	What it does
`bc_twins`	Counts the twins by plant and zone: exact ones (two copies on one spot, the mark of a zone filled twice) and near ones (the same plant within 1 m, but apart). A client counts only the area around it; `bc_twins server` counts the whole world.
`bc_twins <radius>`	The same, with a radius from 0.1 to 5 m.
`bc_twins remove [radius]`	Shows what a cleanup would delete, and deletes nothing: exact twins only, or near ones too when you give a radius.
`bc_twins remove confirm [radius]`	Deletes the extra copies, 1,000 a frame. The copy nearest the ground stays. Nothing a player built is deleted, and crops that players grow are removed only when two stand on one spot.
`bc_twins server ...`	The same, run by a dedicated server.
`bc_twins help`	Shows the usage.

> **Warning:** `bc_twins remove confirm` cannot be undone. Back up the world first. The world saves as usual afterwards; `save` saves at once.

Settings and commands

Setting	Section	Default	Range	What it does	Console
`Forest Scale`	`[04 BetterContinents.Forest]`	0.5	0 to 10	Size of the game's own forest patches	`bc fo sc`
`Forest Amount`	`[04 BetterContinents.Forest]`	0.5	0 to 1	How much forest, against clearings	`bc fo am`
`Forest Factor Overrides All Trees`	`[04 BetterContinents.Forest]`	false	true, false	The forest factor decides 13 more kinds of tree, root and web	`bc fo ffo`
`Forestmap File`	`[04 BetterContinents.Forest]`	empty	a file	The forest map	`bc fo fn`
`Forestmap Multiply`	`[04 BetterContinents.Forest]`	1	0 to 1	How strongly the map scales the game's forest	`bc fo mu`
`Forestmap Add`	`[04 BetterContinents.Forest]`	1	0 to 1	How strongly the map is added to the game's forest	`bc fo add`
`Vegetationmap File`	`[06 BetterContinents.Maps]`	empty	a file	The vegetation map	`bc vegetation fn`

The file settings are not used when `Directory` is set. In Debug Mode the `bc fo` values change the loaded world; `bc fo ffo` takes effect the next time the world is loaded.

What to watch for

- **Settings are stored when a world is made.** Changing them in the config changes only new worlds. See Changing an existing world.
- **Zones keep their plants.** Vegetation is placed when a zone is first generated. After a `bc` change Better Continents regenerates the zones, except those around what players built or worked on.
- **Forest Scale doesn't scale the forest map.** It sizes only the game's own pattern, which the map is combined with.

Next

- Locations, creatures and the start: the location and spawn maps.
- Image maps: sizes, formats and tools.

11 Locations, creatures and the start

This chapter covers three maps and settings that decide where things happen in a world: the location map pins boss altars, traders and dungeons where you want them, the spawn map decides which creatures may spawn where, and the start position sets where new players arrive. It ends with `Skip Default Locations`, which leaves every location to you.

The location map

The location map is a black picture with a dot of colour, a *pin*, for each location you want placed. Its legend, `locationmap.txt` beside it, says which location each colour is. Set it with `Locationmap File` in `[06 BetterContinents.Maps]`, or put `locationmap.png` in your `Directory`.

The pins are placed when the world is made. Then the game goes on with its own placement of locations, unless `Skip Default Locations` is on (see below).

The legend

Each line is a location's name, a colon and a colour:

```
StartTemple: 255,0,0
Eikthyrnir: FF9900
```

- The name is the game's location prefab name. Everything before the last colon is the name, so Expand World Data's `Name:Alias` form works.
- The colour is hex (`RRGGBB`) or `r,g,b`.
- Blank lines and lines that start with `#` are skipped. A line that doesn't read is logged and skipped. A name listed twice keeps its last colour.
- Several names may share a colour. Each pin of that colour then gets one of them.

If the legend is missing, Better Continents writes the default one: 88 locations in 48 colours. These are the locations most maps pin:

Location	Prefab name	Default colour
The start temple	`StartTemple`	255, 0, 0 (`FF0000`)
Eikthyr's altar	`Eikthyrnir`	255, 153, 0 (`FF9900`)
The Elder's altar	`GDKing`	0, 255, 0 (`00FF00`)
Bonemass's altar	`Bonemass`	0, 255, 255 (`00FFFF`)
Moder's altar	`Dragonqueen`	74, 134, 232 (`4A86E8`)
Yagluth's altar	`GoblinKing`	255, 255, 0 (`FFFF00`)
The Queen's entrance	`Mistlands_DvergrBossEntrance1`	153, 0, 255 (`9900FF`)
Haldor, the trader	`Vendor_BlackForest`	0, 0, 255 (`0000FF`)
Hildir	`Hildir_camp`	255, 105, 180 (`FF69B4`)

The rest of the default legend lists the game's crypts, dolmens, ruins, towers, runestones, villages, shipwrecks and others. You can add any location the game has by its prefab name.

How pins become locations

- **A pin** is a patch of one colour whose pixels touch left, right, up or down. Each pin is one location. To put two of the same location close together, keep at least one black pixel between their pins.
- **Colours must match the legend exactly** (alpha is ignored). The log names a colour that no line uses.
- **Where it lands, on a new world:** exactly on the pin's pixel nearest its middle, where the heightmap and the biome map put that pixel. Pixel i of a map N pixels across is at i / (N − 1) across the map, so on a 10,501 px map of the default world the pixels are 2 m apart, as the heights are. Pins that share a colour are dealt out the same way every time, so the same maps give the same world.
- **Where it lands, on an older world** (settings version 11): at the corner of a pixel picked at random from the pin, up to a pixel south-west of where the other maps put it.
- **The maps of a world export** are read the older way, because the export writes each pin for it: a world built from an export finds its pins where the export wrote them.
- **One location per zone.** The game keeps one location in each 64 m zone. When a pin's zone already holds another location, the pin isn't placed, and the log says so.

Drawing the location map

- Draw pins with a pencil tool, not a brush: hard edges and no colour variation. Make them big enough to see without zooming in.
- Keep the legend's colours as swatches, or as a layer you can pick colours from with the eyedropper.
- Draw on a layer over your heightmap and biome map, so you can see where each pin lands.
- Use a layer for each kind of pin (bosses, traders, ruins), so you can change them all at once.

Save it as an RGB PNG of at least 329 × 329 px. A bigger picture places pins more precisely.

Checking the result

Command	What it does
`bc locs`	Counts every location in the world, and warns when the start temple, a boss altar, Haldor or the Queen's entrance is missing
`bc bosses`	Pins those on the map
`bc show <prefix>`	Pins every location whose name starts with the prefix
`bc hide [prefix]`	Removes those pins, or with no prefix every pin `bc show` and `bc bosses` placed. Your own pins stay.

`bc` is a cheat command for the machine that runs the world: see Console commands.

The spawn map

The spawn map decides which creatures may spawn where. It is a colour picture, `spawnmap.png`, set with `Spawnmap File` in `[06 BetterContinents.Maps]` or found in your `Directory`, with a legend beside it, `spawnmap.txt`. A missing legend is written empty.

- **Black** is the game's own spawning: the map changes nothing there.
- **White** always means none: no creature spawns there.
- Any other colour must be in the legend exactly. A colour that isn't is logged once and treated as black.
- The map is read where each 64 m zone's spawner stands, so it works per zone. A picture of 329 × 329 px gives one pixel per zone on the default world; a bigger one doesn't add detail.
- Event spawns (raids) don't follow the map.

Each line of the legend is a colour, a colon, and entries separated by commas:

```
# Red: let wolves and skeletons spawn here, whatever the biome
FF0000: +Wolf, +Skeleton
# Blue: no spawns, except wolves and skeletons, which keep their normal rules
0000FF: none, Wolf, Skeleton
# Green: no wolves and no skeletons
00FF00: -Wolf, -Skeleton
```

Entry	Means
`none`	Stops every creature here, except those named on the same line
`+name`	Lets this creature spawn here whatever the biome: its biome rule becomes "any biome". Its other rules still apply
`-name`	Stops this creature here
`name`	Exempts this creature from `none` on the same line, so it keeps its normal rules. On its own it changes nothing

- The colour is `RRGGBB` or `r,g,b`. Don't put `#` before it: a line that starts with `#` is a comment.
- A line needs exactly one colon; other lines are skipped. A colour number that doesn't read ends the legend at that line, with a line in the log.
- Names are the game's prefab names, not case-sensitive, and `*` is a wildcard: `Troll*`, `*_Elite`, `*camp*`.

The start position

A new player arrives at the start temple. To put the start somewhere you choose, set these in `[05 BetterContinents.StartPosition]`:

Setting	Default	Range	What it does
`Override Start Position`	false	true, false	Places the start temple at Start Position X and Y
`Start Position X`	0	−1,000,000 to 1,000,000	The east–west position, in metres from the world's centre
`Start Position Y`	0	−1,000,000 to 1,000,000	The north–south position, in metres from the world's centre; north is positive

The start temple is placed when the world's locations are placed, so these settings matter only when a world is made.

> **Warning:** Better Continents doesn't check the position. A start under water or on a cliff is where new players arrive. Pick the spot on your own heightmap.

On a new world, the start is placed before the location map's pins, so a pin in the same 64 m zone can't take its place. On an older world (settings version 11) the pins come first, and a pin in the start's zone wins; the log then shows an error, and the game places the start itself.

> **Console**
>
> Command: `bc st os <true|false>`, `bc st x <metres>`, `bc st y <metres>`
>
> Change the loaded world's start settings in Debug Mode. They are used only when the world's locations are placed, so they don't move the start of a world that exists.

Skip Default Locations

After the location map's pins, the game places its own locations: the altars, the traders, the dungeons and the rest. `Skip Default Locations` in `[01 BetterContinents.Global]` (default false) skips that. The start temple and the location map's pins are still placed, so every other location in the world is one you pinned. That includes the locations the alt biomes add, such as the Dark Meadows villages.

> **Warning:** With `Skip Default Locations` on, a location you don't pin isn't in the world at all. Pin every boss altar and trader the game needs, and check with `bc locs`.

Settings and commands

Setting	Section	Default	What it does	Console
`Locationmap File`	`[06 BetterContinents.Maps]`	empty	The location map	`bc l fn`
`Spawnmap File`	`[06 BetterContinents.Maps]`	empty	The spawn map	`bc spawn fn`
`Override Start Position`	`[05 BetterContinents.StartPosition]`	false	Places the start at X and Y	`bc st os`
`Start Position X`	`[05 BetterContinents.StartPosition]`	0	East–west, in metres	`bc st x`
`Start Position Y`	`[05 BetterContinents.StartPosition]`	0	North–south, in metres	`bc st y`
`Skip Default Locations`	`[01 BetterContinents.Global]`	false	Leaves out the game's own location placement	`bc g skipdefaultlocations`

The file settings are not used when `Directory` is set. `bc reload lm` and `bc reload spawn` read a map and its legend again in Debug Mode.

What to watch for

- **Locations are fixed when the world is made.** Regenerating a zone builds its location again on the same spot. For a new location layout, make a new world. See Changing an existing world.
- **A biome map that leaves a biome out** leaves out the game's locations that need it. See Biomes and the biome map.

Next

- Forests and vegetation: the vegetation map, which works like the spawn map.
- Alt biomes: the alt biomes add locations of their own.

12 Worlds from noise

You don't need maps to make a Better Continents world. This chapter covers worlds whose terrain comes from noise: the height layers, the Global settings that shape them, and the presets that ship with the mod. The same height layers can also add detail on top of a heightmap.

Where the terrain comes from

Better Continents makes a world's terrain with its own height function only when the world has a heightmap or at least one height layer. A world with neither gets the game's own terrain, and then `Continent Size`, `Sea Level Adjustment` and `Mountains Allowed At Center` have no effect.

A world made From Config has no height layers: the config has no settings for them. A world gets its layers from a preset, or you build them in a loaded world with the `bc hl` commands or the settings window (Debug Mode). They are saved with the world, and `bc savepreset <name>` saves them, with everything else, as a preset.

The presets that ship with the mod

The New World screen's Better Continents box offers four presets: Continents, Pangaea, Swirls and Vanillaish. Each is a stack of three height layers and holds no maps. Pick one, and the world's terrain comes from its layers.

They are old settings-version 7 files, so a world made from one is a settings version 11 world. At the game's own world size, which they use, version 11 and version 12 worlds generate alike. See How a world is made.

To make your own preset, build a world you like, then type `bc savepreset <name>` in it. The preset is saved in `<save data>/BetterContinents/presets/`, beside the game's `worlds_local` folder, and appears in the New World screen. A preset keeps the settings version it was saved with, so it makes the same world every time it is used with the same seed.

Height layers

Height layers are stacked noise. Each layer has its own noise (OpenSimplex2, Perlin, cellular and others), fractal settings, an optional warp, an optional mask and a blend mode. Better Continents applies them in order, on top of the heightmap if there is one.

How layers combine

For each point, Better Continents starts from the heightmap's value (value × `Heightmap Amount`, with the default heightmap settings), or from 0 without a heightmap. Then each layer in turn:

```
layer value = Blend(value so far, the layer's noise, its blend mode)
value       = the layer value, or with a mask: Lerp(value so far, layer value, mask)
```

The result goes through the same last step as a heightmap: −0.15, plus the Sea Level Adjustment offset, then × 200 for metres. So at the default Sea Level Adjustment of 0.5, a stacked value of 0.30 is the waterline. See Formulas.

- Each layer's noise is scaled to 0 to 1 over the world, then passes through its filters in this order: invert, smooth threshold, range, threshold, opacity.
- **A new layer's blend mode is Normal, which replaces everything below it**, your heightmap included, unless the layer has a mask. To add to the terrain below instead, choose another blend mode, such as LinearDodge (which adds), with a small range, or give the layer a mask.

- The blend modes are Normal, Darken, Multiply, ColorBurn, LinearBurn, Lighten, Screen, ColorDodge, LinearDodge, Overlay, HardLight, SoftLight, VividLight, LinearLight, PinLight, HardMix, Difference, Exclusion, Subtract and Divide.
- A frequency is in cycles per metre: the default, 0.0005, is one cycle every 2 km. `Continent Size` divides it (see below).
- Masks start from the second layer (layer 1).
- Each layer's seed comes from the world seed, so a preset makes a different world with a different seed.

Building layers

Layers are numbered from 0. Every change rebuilds the terrain, like any other `bc` change.

Command	What it does
`bc hl add`	Adds a layer at the end: OpenSimplex2 noise, fractal type FBm, 7 octaves, frequency 0.0005, blend mode Normal.
`bc hl <n> npreset def`	Sets layer n's noise to that default noise.
`bc hl <n> npreset ri`	Sets layer n's noise to ridged noise (fractal type Ridged), for ridge-like terrain.
`bc hl <n> n <name> [value]`	Layer n's noise settings.
`bc hl <n> nw on true`	Adds a warp to the noise, set under `bc hl <n> nw`. `on false` removes it.
`bc hl <n> m on true`	From layer 1: gives the layer a mask, set under `bc hl <n> m`. `on false` removes it.
`bc hl <n> mpreset def` or `25%`	From layer 1: a default mask with a smooth threshold of 0.6 to 0.75. It doesn't switch the smooth threshold on: follow it with `bc hl <n> m ust true`.
`bc hl <n> mpreset ri`	As above, with a warp on the mask.
`bc hl <n> mw on true`	For a layer with a mask: adds a warp to the mask, set under `bc hl <n> mw`.
`bc hl <n> delete`	Deletes layer n.
`bc hl <n> down`	Swaps layer n with the next one (not for the first or last layer).

Console commands lists the noise settings (`nt`, `fq`, `ft`, `fo` and the rest). The console reads only text, numbers and `true` or `false`, so the settings chosen from a list (noise type, fractal type, the cellular and warp types, and the blend mode) are set in the settings window. The window also shows a preview of each layer and of the final result.

> **Tip:** For ridged mountain ranges, give a layer the ridged noise preset (`bc hl <n> npreset ri`) and a warp on its noise (`bc hl <n> nw on true`).

The Global settings

These settings are in `[01 BetterContinents.Global]`. Like every World setting, they are stored in a world when it is made. In Debug Mode, `bc g` changes them in the loaded world.

Setting	Default	Range	What it does	Console
`Continent Size`	0.5	0 to 1	Scales the height layers' noise features	`bc g cs`
`Sea Level Adjustment`	0.5	0 to 1	Moves the land against the sea	`bc g sl`
`Mountains Allowed At Center`	false	true, false	Off: holds the terrain near the centre down	`bc g mc`
`Rivers`	true	true, false	Whether the game makes its rivers	`bc g r`
`Ashlands Gap`	false	true, false	Whether to add the game's ocean gap around the Ashlands	`bc g ag`

Setting	Default	Range	What it does	Console
`Deep North Gap`	false	true, false	Whether to add the game's ocean gap around the Deep North	`bc g ng`
`Map Edge Drop-off`	true	true, false	Whether the land drops away at the edge of the world	`bc g me`

`World Size` and `Edge Size` are in the same section: see World size and big worlds. `Skip Default Locations` is too: see Locations, creatures and the start.

Continent Size

`Continent Size` scales the features of the height layers' noise. It turns into a scale of 3 at 0, 1 at 0.5 and 0.2 at 1, which divides each layer's frequency. So at 0 the features are about three times larger, at 1 about five times smaller, and 0.5 leaves them as they are. It does not scale a heightmap or the game's own terrain.

Sea Level Adjustment

`Sea Level Adjustment` moves the land up or down against the sea, which stays at 30 m. It adds `Lerp(1, -1, value)` before the × 200, which is 0 at the default 0.5. Each 0.05 above 0.5 lowers the land by 20 m, and each 0.05 below raises it by 20 m: `bc g sl 0.55` puts the land 20 m lower against the water.

With a heightmap at `Heightmap Amount` 1, the waterline is at heightmap value 2 × Sea Level Adjustment − 0.7:

Sea Level Adjustment	Waterline (heightmap value at Amount 1)	Result
0.35	0	No sea: black is at the water's surface
0.5 (default)	0.30	The default
0.85	1.0	Everything is under water

At other Amounts, see Formulas.

Mountains Allowed At Center

While `Mountains Allowed At Center` is off (the default), Better Continents holds the terrain within about 600 m of the world centre to at most 46 m above the water, and blends back to the full height by 1,000 m from the centre. This applies to heightmaps too. Turn it on if your map puts high ground at the centre.

Rivers, the gaps and the edge

- `Rivers` switches the game's rivers on or off. With a heightmap and `Heightmap Override All` on (the default) they never appear whatever it says: carve rivers into the heightmap instead.
- `Ashlands Gap` and `Deep North Gap` add the ocean gaps the game puts around the Ashlands and the Deep North. Both are off by default, and custom maps usually don't need them.
- `Map Edge Drop-off` lets the land drop away over `Edge Size` past `World Size`. With it off, the land doesn't drop away and the world has no edge.

Settings older worlds still use

Mountains Amount, Ocean Channels and the ridge settings are not in the config and have no console commands. Only worlds stored in the oldest settings formats (1 to 6) still use the values they saved, and `bc info` prints them. For ridged terrain in a new world, use a height layer with the ridged noise preset.

What to watch for

- **Without a heightmap or a height layer, the terrain is the game's own.** A world made From Config with no map looks like a vanilla world, whatever `Continent Size` or `Sea Level Adjustment` say.
- **A Normal layer hides what is below it.** Mask it, or use another blend mode, to keep your heightmap.
- **Layers are tied to the world, not the config.** Editing `BetterContinents.cfg` does not change a world's layers. Save them in a preset to use them again.

Next

- Heights and the heightmap: the height formula, which height layers share.
- Console commands: every height-layer setting.
- World size and big worlds: World Size and Edge Size.

13 World size and big worlds

This chapter is about making a world bigger than the game's own, and about what changes once it is very big. Past about 16 km from the centre, the game's own way of keeping track of objects stops working well, and Better Continents fixes that with Wide Sectors. Read the Wide Sectors part before you make or convert a big world: part of it can't be undone.

World Size and Edge Size

`World Size` in `[01 BetterContinents.Global]` is the world's radius in metres (vanilla: 10,000). `Edge Size` is how far the world's edge reaches past it (vanilla: 500). On a new world:

- the maps span 2 × (World Size + Edge Size) from side to side: 21,000 m with the defaults, 41,000 m with World Size 20,000 and Edge Size 500;
- the land drops away over the Edge Size band (with `Map Edge Drop-off` on, the default);
- the game's edge, where ships are pushed back, follows World Size.

A world made by an older Better Continents (settings version 11) keeps the game's 21,000 m, whatever its World Size says; there World Size only moves the edge. See How a world is made.

What follows the world's size

On a new world, these are laid out to its size:

- every image map: one pixel covers 2 × (World Size + Edge Size) / the picture's width;
- the alt-biome grid, so locations can be placed across the whole world;
- where the game searches for places to put locations;
- the game's own biome bands, and the Ashlands and Deep North rings;
- lakes and stream sources;
- the minimap.

Terrain detail keeps its size: hills and noise are as big as in a vanilla world.

Wide Sectors: worlds past 16 km

> **Warning:** Read this before you make or convert a world whose World Size + Edge Size is more than 16,350 m. A world with Wide Sectors needs Better Continents 0.10.3 or later from then on.

Why big worlds need it

Valheim files its objects (trees, rocks, creatures, buildings, dropped items) by 64 m zone, and its filing stops about 16.4 km from the centre to the east, west, north and south. Past that, the game's own code goes wrong in three ways, each starting at its own distance:

World Size + Edge Size	What goes wrong without Wide Sectors
up to 16,350 m	Nothing. Wide Sectors never applies, and the world is saved exactly as before.
past 16,350 m	Everything more than about 16.4 km out shares one list. A player anywhere out there is sent the whole list, not only what is near them: on a test world, 5,742 objects when 965 were nearby. The list grows with every area explored or built on out there, and it is saved as one file.

World Size + Edge Size	What goes wrong without Wide Sectors
past about 22,100 m	A patch of 8 × 8 zones about 22 km south-west of the centre is saved under the same file name as every portal in the world. Once anything is out there, the two overwrite each other: either that patch's objects are lost, or all the world's portals are gone after the next restart.
past 32,767 m	Each zone has one hidden spawner that spawns its creatures, placed once, when the zone is first made. The game saves a spawner exactly while both its coordinates are within 32,767 m of the centre: a zone at 25,600 m by 25,600 m is saved where it is. Farther out, the save moves the spawner to the 20 km line (20,000 m on that axis). After a restart those zones spawn no creatures of their own, and their spawners pile up along the 20 km line.

A world of World Size 32,264 and Edge Size 500 (32,764 m) meets the first two; the spawner problem starts just past it.

What Wide Sectors does

With Wide Sectors, every zone out to 65.5 km from the centre gets its own place in memory and in the save, on the server and on every player's machine. Spawners are saved where they are, at any distance, and the south-west patch is kept apart from the portals. It costs about 32 MB of memory on each machine while a wide world is loaded, and planning a save takes 21 to 31 ms instead of 3 to 4 ms (measured on one machine).

The setting

`Wide Sectors` in `[07 BetterContinents.Misc]` is read by the machine that makes a world, in its New World screen, and by the machine that runs the world (single player, the host or a dedicated server) when it loads. Joining players follow the world, whatever their own setting says. It has no console command, and a world export never writes it into `export.cfg` (a line for it there is ignored).

A dedicated server never makes a Better Continents world itself: you make the world on your own computer and copy its folder to the server (see Multiplayer and dedicated servers). So your own `Wide Sectors` decides whether a new big world is made wide, and the server's decides only what happens when it loads one: On converts a big world that isn't wide yet.

Value	A new world past 16,350 m	An existing world past 16,350 m
Auto (default)	is made wide	keeps the sectors it has
On	is made wide	is converted to wide when it loads (one way)
Off	is made with the game's sectors	keeps the sectors it has; a wide world stays wide

With the default, a new big world is wide from the start, and you don't need to do anything. A world of 16,350 m or less is never touched. The log says what happens to a new world:

```
Sectors: this new world reaches 24500 m, so it is made with wide sectors (Wide Sectors is
  Auto): every zone out to 65504 m has a sector and save chunks of its own.
```

The world records that it is wide in its own settings, and `bc info` says *Wide sectors: every zone out to 65504 m has a sector and save chunks of its own*. With Off, a wide world stays wide, and the log says why: *Wide Sectors is Off, but this world stays wide: its settings say it is a wide world*. Under Auto, an existing big world that is not wide gets a line that says it keeps the game's sectors and that On converts it.

If the game's save code has been changed, for example by another mod, so that a wide save cannot be loaded wide, the world is not loaded at all, and nothing is saved over it: the log has an error, and the main menu shows Better Continents' message. A dedicated server stops.

Converting an existing world

A big world made before 0.10.3, or made with Wide Sectors Off, still has the problems above. To convert it:

1. Back up the world folder (`worlds_local/<World>`).
2. Set `Wide Sectors` to On on the machine that runs the world.
3. Load the world. Better Continents converts it as it loads, and before the world's first save the game saves a copy of the old save beside it, named `<world>_backup_<date and time>`, which the game's Manage Saves list shows. The log says so, for example:

```
Sectors: 5653 objects lie beyond the game's sectors, where the save filed them in chunk (0,
  0), and 121 are of the zones the game's save files as its portals' chunk (1, 0) ...
Sectors: this world was saved with the game's sectors and is converted to the wide ones.
  Before its first save the game saves a copy of it as '<world>_backup_<date and time>',
  beside it. ...
```

Once the converted world has been saved, its own settings say it is wide, so it stays wide under Auto too. Set `Wide Sectors` back to Auto then if you don't want other big worlds that this machine runs converted as well.

Converting puts the objects from the shared list and the misfiled patch back where they belong. It can't undo damage already done: objects already doubled stay doubled, spawners already moved to the 20 km line stay there, and whatever the portal file already overwrote stays lost. The longer a big world runs without Wide Sectors, the more of that damage there is.

> **Warning:** Converting is one way. Afterwards the world needs Better Continents 0.10.3 or later. An older Better Continents stops at the world's wide marker (*Unknown feature: 68* in the log) and opens it as a world without Better Continents. If an older version, or the game without Better Continents, saves a wide world, it writes objects twice, and they come back as two of each (two of every chest). No setting can stop an older version from doing this.

In multiplayer this can't happen by accident: a player with another Better Continents version is refused when they join. The risk is the machine that runs the world: don't open a wide world with an older version there.

Memory, start time and joining

The times and memory below were measured on one machine, a Linux dedicated server; the join times follow from the transfer rate.

- **Starting.** On a server with 32 cores, a vanilla-size world starts in about 11 s, a 24.5 km world on the game's own terrain in about 18 s, and a world of 32,264 + 500 m with five 16,384 px maps in about 16 s.
- **The alt-biome grid** is built on every machine that loads the world. At 32 km it has 6,392 × 6,392 points and costs about 0.8 GB. Better Continents builds it on every core: about 5 s on a 32-core machine, where the game's own way takes 29 s on one core. Fewer cores take longer, and a world on the game's own terrain gains less (at 24.5 km, about 10 s instead of 20 s). From 42 km it is capped at 8,192 × 8,192 points.
- **Maps.** A 16-bit 16,384 px map takes 537 MB decoded. `Map Memory` in `[07 BetterContinents.Misc]` (default 512 MB) is how much decoded map data a machine keeps, so at the default not even one such map stays decoded whole. On a server with RAM to spare, use 1024 to 2048 for a 16k world. With `Compact Maps` on Auto, the default, a world with a map more than 8,192 px across is made compact: one world file with five 16,384 px maps came to 52.6 MB, against 352.5 MB as pictures. See Image maps.

- **Joining.** A player who doesn't have the world yet is sent the world file at `Settings Transfer Rate` (default 512 KB/s): 52.6 MB takes about 1.7 minutes, 203 MB about 6.5 minutes. Raise the rate if the server's upload allows it. Better still, give players the world's `.bcworld` file (`bc_cache export`), so they join without a download. See Multiplayer and dedicated servers.

Tall terrain in a big world

High terrain (a Heightmap Amount above 5) works the same in a big world. See Heights and the heightmap.

The sea and the Deep North in a big world

The game's shaders draw three things by fixed circles that fit a world of the game's own size, whatever the world's maps say:

- **The Ashlands sea's red.** The water shader colours the sea red past a circle about 12,000 m from (0, 4,000), the game's Ashlands ring. Better Continents 0.10.3 and later set the colours of each 64 m patch of water so the red follows the world's own hot sea instead: the sea that heats swimmers and damages ships. With a heat map, that is where the heat map makes the sea hot. Without one, it is the Ashlands ring laid out to the world's size. Only the sea counts, so lava on land doesn't colour the water beside it. Where the edge of the hot sea is not the game's ring, it is drawn in steps of 64 m, softened by how much of each patch is hot.
- **The calm sea of the Deep North.** The water shader draws the waves dying down past a circle about 12,000 m from (0, −4,000). Boats and fish float on those same waves, so the calm water stays where the game puts it, in a big world and in a world whose biome map moves the Deep North.
- **The Deep North's ground.** The terrain shader marks the Deep North with the same circle. Outside it, farmland is not drawn as soil, mountain ground is raised as deep snow (2.1 m), and the shore band sits 2 m lower. Better Continents doesn't change this.

In a big world, many places outside these circles are far from the north, and they show the calm sea and the Deep North's ground too.

Limits

- World Size + Edge Size can go up to 65,000 m. Beyond 65.5 km, objects share one list again, as in the game.
- In practice, start time and memory set the limit. Set World Size to how far players will really go. With `Map Edge Drop-off` off the world has no edge, but Wide Sectors still follows World Size + Edge Size.
- Far from the centre, positions are less precise: about 4 mm at 65 km.
- With Expand World Size installed, the larger of its size and World Size + Edge Size decides whether a world needs Wide Sectors. Expand World Size then lays the world out itself, as before. See Other mods.

Settings

Setting	Section	Default	What it does
`World Size`	`[01 BetterContinents.Global]`	10000	The world's radius, in metres
`Edge Size`	`[01 BetterContinents.Global]`	500	How far the edge reaches past World Size, in metres
`Wide Sectors`	`[07 BetterContinents.Misc]`	Auto	Wide sectors for worlds past 16,350 m (above)
`Map Memory`	`[07 BetterContinents.Misc]`	512	Decoded map data a machine keeps, in MB

Setting	Section	Default	What it does
`Compact Maps`	`[07 BetterContinents.Misc]`	Auto	Keeps maps compact; see Image maps
`Max Map Size`	`[07 BetterContinents.Misc]`	16384	The largest map read from a file, in pixels
`Settings Transfer Rate`	`[07 BetterContinents.Misc]`	KB512	How fast a joining player is sent the world

> **Console**
> Command: `bc g worldsize <metres>` and `bc g edgesize <metres>`
>
> Change the loaded world's World Size and Edge Size in Debug Mode. See Changing an existing world before you change a world you play.

Next

- Heights and the heightmap: tall terrain and its rules.
- Coming from 0.9.x or older: every one-way change in one place.
- Other mods: Expand World Size and other mods.

14 Changing an existing world

A world keeps the settings and maps it was made with. This chapter covers the ways to change a world you already have: the `bc` console commands and the settings window, reloading a map after you edit it, and zone regeneration, which brings the parts of the world the game has already made in line with a change. A change removes things from the world, and some changes cannot be undone, so back the world up first (see Backups).

What reaches an existing world

What you change	Does it reach a world you have?
A World setting in `BetterContinents.cfg`: the map files, `World Size`, `Heightmap Amount` and the rest	No. It is a default for the worlds you create afterwards.
A picture	No, until you read it in again with `bc reload`.
A Live setting: `Settings Transfer Rate`, `Map Memory` and `Max Map Size` in `[07 BetterContinents.Misc]`, and `[09 BetterContinents.Export]`	Yes, while the game runs. These belong to the machine, not to the world.
A setting changed with the `bc` commands or the settings window	Yes, at once, and the world saves it.
`Override version` in `[00 BetterContinents.Debug]`	Yes: it is read at every save. It is for testing only (see Override version).
`Wide Sectors` in `[07 BetterContinents.Misc]` set to On	Yes: an existing world past 16,350 m is converted when it loads, one way (see Converting a big world to Wide Sectors).

How a world is made explains the World, Live and Local settings.

Debug Mode and the bc commands

The `bc` commands change the settings of the world you are in. They are cheat commands: they need `devcommands` on, and they run only on the machine that runs the world, in single player or on the host. A dedicated server does not run them for an admin. To change a dedicated server's world, stop the server, copy the world folder to your own computer, make the change there in single player, and copy the folder back.

With the Server Devcommands mod, a player who has joined someone else's world can type `bc` too, but there a change alters only that player's own copy of the settings and their own view of the ground, never the world or the other players.

`Debug Mode` in `[00 BetterContinents.Debug]` (default off) makes the work easier. In a world that uses Better Continents, on the machine that runs it, Debug Mode:

- turns `devcommands` on when you spawn;
- shows *Better Continents Debug Mode Enabled!* at the top left of the screen;
- adds a **Better Continents** button to the Esc menu and to the large map. It opens the settings window, which shows the same settings as the `bc` commands, with their values, and changes them the same way. Alt+F8 opens it too while the Esc menu or the large map is open.

You don't need Debug Mode for the commands: typing `devcommands` in the console turns cheats on as well. Debug Mode itself does not reveal the map, but most changes do, because they redraw the minimap and explore all of it. Leave Debug Mode off for a world you play.

Using the commands

Each setting the console can change has a group and a short name. The groups are `g` (global), `h` (heightmap), `r` (roughmap), `b` (biome map), `fo` (forest), `st` (start position), `heat`, `terrain`, `l` (location map), `paint`, `lava`, `moss`, `vegetation`, `spawn`, `hl` (height layers) and `ab` (alt biomes). Console commands lists every name.

> **Console**
>
> Command: `bc <group> <setting> [value]`, for example `bc h am 2`
>
> With a value, sets the setting for the loaded world: `bc h am 2` sets Heightmap Amount to 2. Without one, prints the current value. `bc h am help` describes the setting and gives its range, and `bc h` or `bc h help` lists the group.

A map's file is changed the same way: `bc <group> fn <file>` takes a full path, a folder (its standard file name is used) or a file name in `Directory`.

Settings that are not in the console, such as `Directory`, `Compact Maps` and the Live and Local settings, change only in `BetterContinents.cfg`. `High Terrain` is in the console, as `bc h ht`.

A change is kept

A change made with `bc` or in the settings window changes the world's own settings. Better Continents writes them into the world's `BetterContinents` file at the next save, so the world keeps the change. It is not a preview: to try something without keeping it, work on a copy of the world.

Players connected from other machines keep the settings they joined with. When they join again, they download the changed settings.

What a change does

After a change (any setting, `bc reset`, an alt-biome change, a map reload or Biome precision), Better Continents:

1. for most changes, redraws the minimap, which reveals all of it, and rebuilds the alt-biome grid. A change of Biome precision does neither, and the `bc ab` commands leave the minimap alone and rebuild only as much of the alt-biome data as they affect;
2. builds the loaded terrain again, the distant terrain and the grass too. That stalls the game for a second or two (1.6 s measured);
3. regenerates the world's zones, as the next section describes.

A change of `High Terrain` (`bc h ht`) changes only which of the game's height rules apply: it switches them at once and regenerates the zones, without redrawing the minimap or rebuilding the loaded terrain. Only in a world whose land can pass 9,000 m does it place the alt biomes again, as an alt-biome change does. See Heights and the heightmap.

Zone regeneration

The game fills a zone, a 64 m square, with its trees, rocks, locations and creature spawners once, the first time somebody comes near, and then keeps what it made. So a zone that is already generated does not follow a change by itself. After every change, and on `bc regen`, Better Continents resets the world's generated zones, and each reset zone generates again with the current settings, its locations included, when somebody comes near.

It runs on the machine that runs the world, single player or the host, in a world that uses Better Continents.

What is left alone

A generated zone is left alone, together with the eight zones around it, when it holds:

- something a player built or planted: anything placed from the build menu or planted with the cultivator, such as buildings, chests, ships, portals, crops and saplings that are still growing;
- a tombstone;
- ground a player has worked with the hoe, the cultivator or the pickaxe: fields, paths, roads, levelled or dug ground;
- you, while you are inside a dungeon.

A player connected from another machine keeps the zones their game has loaded around them: the 5 × 5 zones around them with the game's default settings, and never fewer than 3 × 3. This is checked again as the work goes, so a player who moves keeps the zones around where they are now, and something you build while it runs keeps its zone from then on. The zones around your own character on the surface are regenerated like the rest.

A location that reaches into more than one zone is kept or regenerated whole: if one of its zones is left alone, the others are too.

What is removed

In a reset zone, everything the world made goes: the trees, rocks and buildings the world placed, creatures that are not tamed, and items lying on the ground. A creature that a creature spawner made goes with its spawner, wherever it has walked to, unless it is tamed or stands in a zone that is left alone.

Players, what players built, tombstones and tamed animals are never removed, in any zone. A tree a player grew from a sapling counts as the world's once it has grown, so it goes unless something nearby keeps its zone.

Where locations stand was decided when the world was made: a reset zone builds its location again on the same spot. `Skip Default Locations` in `[01 BetterContinents.Global]` acts when the game places a world's locations, while the world is being made (see Locations, creatures and the start). For a new layout of locations, make a new world.

The ground

The ground of every zone, the zones left alone included, is built from the world's settings, so after a change it follows the new settings everywhere. The edits players made with the hoe, the cultivator and the pickaxe are stored as changes to the ground, and they ride on the new ground. Where a zone that is left alone meets a zone that was reset, Better Continents clears the height edits along the shared edge, on the side that is left alone, so that the two grounds meet. The paint on the ground stays.

> **Warning:** A change that lowers the ground under a base leaves its pieces standing in the air, and the game breaks a piece that has lost its support a few seconds later. In a test, a workbench and a floor left about 3 m up broke about 8 seconds later. A change that raises the ground buries them. Trees and rocks in zones that are left alone float or sink the same way. Make big height changes before you build, or away from what you want to keep.

`bc regen` on its own changes no ground.

How it runs

- The work is spread over frames, the zones nearest the players first, so the game keeps running. With other players connected, it removes at most 100 objects a frame and about 6,000 a second.
- The console shows a line when it starts, a line at every tenth of the way when 20 or more zones are to be reset, and a line when it finishes:

  ```
  Zone regeneration: resetting <n> of <m> generated zones; <k> left alone (near something a
    player built or worked on, a tombstone, or a player).
  Zone regeneration: <done> of <n> zones (<percent>%), <count> objects removed.
  Zone regeneration finished in <seconds> s: <n> zones reset, <count> objects removed. A
    reset zone generates again with the new settings when somebody comes near.
  ```

  The last line can say more: how many objects stayed, how many zones were left alone after all, and at how many left-alone zones the ground was adjusted to meet the reset ones.
- A new change, or `bc regen` again, while it runs starts it over, after the zone it was working on is finished.
- A save while it runs never stores a zone half emptied: the zones it has emptied are saved as not generated, and they generate again when somebody comes near.
- If an error stops the work, the console says `Zone regeneration stopped by an error, see the log.` The zones it had emptied are finished, and the zones it had not reached are as they were: `bc regen` resets them.
- A location whose zone the game generates while the work runs, because somebody came near it, can end up half placed or placed twice. `bc regen` again sets it right.

Where it doesn't run

The console says	Because
`Zone regeneration: it runs on the machine that runs the world: in single player, or on the host.`	You have joined someone else's world. Make the change on the host or in single player.
`Zone regeneration: this world does not use Better Continents.`	The world is the game's own. Better Continents never resets its zones.
`Zone regeneration: load a world first.`	You are in the main menu.

Resetting part of the world

`bc regen` takes words that reset only part of the world. Every rule of the plain command still holds: zones near something a player built or worked on, a tombstone or a player are left alone, a location that reaches into such a zone is kept whole, and players, what players built, tombstones and tamed animals are never removed. A word only ever takes zones out of the reset.

Word	What it means
`dry`	Counts what would be reset and changes nothing: the number of zones, their biomes (at their middle), the nearest zones first, and the whole list in the log. Run it before the real thing.
`biome=Meadows,BlackForest`	The zones where any corner is one of these biomes (a zone half in the biome is in). The names are those of the biome map's legend, in any case, with a comma between; `black_forest` and `Black-Forest` read too.
`min=<m>` and `max=<m>` (`radius=<m>` is `max`)	The zones whose middle is at least, or at most, this many metres from `pos` (the world's centre when there is no `pos`).
`pos=<x>,<z>`	The place `min` and `max` measure from, in metres.
`zone=<x>,<z>`	One zone, by its zone coordinates. A zone is 64 m; a place's zone is its x and z divided by 64, rounded.

Word	What it means
`chance=0.25`	Each zone that is left is reset with this chance. A number past 1 is a percentage: `chance=25`. A zone keeps its draw for as long as the request stands.

Run it with `dry` first, then without:

```
bc regen dry biome=AshLands min=3000
bc regen biome=AshLands min=3000
bc regen zone=12,-40
bc regen max=2000 chance=0.5
```

- A word that is not understood starts nothing, and the console says what it expected.
- A zone the words leave out is treated like a zone near a player's base: it is left alone, and a location that reaches over the border of what you asked for is kept whole, so the area that is reset can be a little smaller than asked.
- While `Debug Reset Command` holds a console command, the words are refused, because a console command takes no filters. Empty the setting to use them.
- It runs on the machine that runs the world (single player or the host), in Debug Mode, like the plain command. A `dry` run is refused while a regeneration is running, and says so.

bc regen and bc reset

> **Console**
> Command: `bc regen`
>
> Regenerates the zones now, as a change does. It leaves the loaded terrain as it is, because nothing has changed, and it does not redraw the minimap or rebuild the alt-biome grid. On its own it resets every generated zone; words after it reset only part of the world (the next section). `bc regen help` says what it does. With a command in `Debug Reset Command`, that command runs instead.

> **Console**
> Command: `bc reset`
>
> Redraws the minimap (which reveals all of it), rebuilds the alt-biome grid and the loaded terrain, and then regenerates the zones. Most changes do all of this for you.

Debug Reset Command

`Debug Reset Command` in `[00 BetterContinents.Debug]` is empty by default, and empty means Better Continents regenerates the zones itself. A console command written there runs instead, after each change and on `bc regen`: for example `zones_reset start`, the command of the Upgrade World mod. Better Continents then does not regenerate the zones itself. See Other mods.

A `BetterContinents.cfg` from 0.9.x that still held that old default, `zones_reset start`, was set to empty once, with a line in the log that says how to go back. A command you write there yourself stays.

Reloading a map

A world stores its maps when it is made, so editing a picture changes nothing until you read it in again:

> **Console**
> Command: `bc reload <map>`, for example `bc reload hm`
>
> Reads the map's picture again and uses it for the loaded world. `<map>` is `hm` (heightmap), `rm` (roughmap), `bm` (biome map), `ab` (alt-biome map), `terrain`, `paint`, `heat`, `lava`, `moss`, `fom` (forest map), `vegetation`, `lm` (location map), `spawn`, or `all` for every map. Only the maps the world uses are offered.

The picture is read from the file the map came from; if that file is gone, from the file the config names now. Then Better Continents rebuilds the terrain and regenerates the zones, as for any change. Reloading the alt-biome map also plants the alt biomes again. A biome map whose new picture or legend has an error is not used: the world keeps its biome map, and the console says so.

Converting a big world to Wide Sectors

A world whose World Size + Edge Size is more than 16,350 m, made before 0.10.3 or with `Wide Sectors` Off, still uses the game's own sectors, with the problems World size and big worlds describes. Set `Wide Sectors` in `[07 BetterContinents.Misc]` to On on the machine that runs the world, and the world is converted when it loads. Before the converted world's first save, the game saves a copy of the old save beside it, named `<world>_backup_<date and time>`.

> **Warning:** Converting is one way. Afterwards the world needs Better Continents 0.10.3 or later, and an older version that saves it writes its objects twice. Converting cannot undo damage already done. Back up the world folder first, and read Converting an existing world before you do it.

Override version

> **Warning:** `Override version` is for testing. While it holds a number, every world is saved in that settings version, and a world saved in an older version than its own loses what its version added. Leave it empty, on a server above all.

`Override version` in `[00 BetterContinents.Debug]` is empty by default.

- **Empty:** every world is saved in its own settings version, an older one than 11 as 11, and a new world made From Config gets the newest, version 12.
- **A whole number of 11 or more:** while it is set, every world is saved in that version, a world made from a preset too, and a new world made From Config gets it. A host also sends it to joining players. 11 saves a version 12 world as version 11, which loses what version 12 added from its next load. 12 stamps a version 11 world as version 12 at its next save: zones generated from then on follow the 0.10 rules, and seams are likely where they meet the old ones.
- **A number below 11** writes an old fixed layout of that number and drops data.

It is read at every save and every join, so an edit changes the next autosave of a loaded world. Every `export.cfg` lists it as empty, so loading an export through the config (`bc_import ... config`, or pasting `export.cfg` into `BetterContinents.cfg`) clears it.

One use outside testing: to make a world from your own maps as Better Continents 0.9.4 would have made it, set it to 11 before you create the world From Config, and empty it again afterwards.

Backups

Zone regeneration removes what the world placed in every reset zone. Before you use the `bc` commands on a world you care about, copy its whole folder somewhere safe:

- Windows: `%AppData%\..\LocalLow\IronGate\Valheim\worlds_local\<World>`
- Linux: `~/.config/unity3d/IronGate/Valheim/worlds_local/<World>`

Better Continents keeps the previous copy of its own file as `BetterContinents.old` in the same folder, but that holds only the save before the last, and none of the game's files. It is not a backup of the world.

Settings

Setting	Section	Default	What it does
`Debug Mode`	`[00 BetterContinents.Debug]`	false	Turns cheats on at spawn and adds the settings window, on the machine that runs the world
`Debug Reset Command`	`[00 BetterContinents.Debug]`	(empty)	A console command that runs instead of Better Continents' own zone regeneration
`Override version`	`[00 BetterContinents.Debug]`	(empty)	Saves every world in the given settings version; for testing
`Wide Sectors`	`[07 BetterContinents.Misc]`	Auto	On also converts an existing big world when it loads

Next

- Console commands: every `bc` command.
- World size and big worlds: Wide Sectors in full.
- Multiplayer and dedicated servers: what players see after a change.

15 Multiplayer and dedicated servers

A Better Continents world can run on a dedicated server or be hosted from the game. The machine that runs the world sends its settings and maps to each player who joins, so players need only the mod, not the pictures or the config. This chapter covers versions, how the settings reach players and how long that takes, how to let players skip the download, and setting up a server.

Everyone on the same version

On a world that uses Better Continents, the server and every player must run exactly the same version of the mod, whatever the world's settings version. A 0.9.x player cannot join a 0.10 server, a 0.10 player cannot join a 0.9.x server, and two different 0.10 releases refuse each other too. A player with another version, or without Better Continents, is refused when they join; a player with another version sees a message such as *Better Continents: local mod version doesn't match the servers*.

A world without Better Continents has no check, so players without the mod can join it.

Update the server and every player together, and back up the world first. Coming from 0.9.x or older says what an older version does to a newer world.

How the settings reach players

When a player joins a Better Continents world:

1. Their game tells the server its Better Continents version, and which world settings it already holds in its cache.
2. If the cache holds this world's settings exactly as the server has them, the player's game loads them from there, and nothing is downloaded.
3. Otherwise the server sends the world's settings and maps, before the join finishes. The player sees a bar, *Better Continents: downloading world settings from server …*, and their game keeps what it received in its cache.

The cache is the `BetterContinents/cache` folder beside the game's `worlds_local` folder. Each world's settings are filed under an id worked out from their content, so any change to them, such as a `bc` change on the host, gives a new id, and the next join downloads the world's settings again. Players who are connected during a change keep the settings they joined with until they join again.

The player's game then plays the world's own settings, whatever the player's config says. The server also sends the world's alt-biome placement, which the player's game uses; both sides write to their logs whether the two agree, and a disagreement is a warning, never a refusal.

How long a join takes

`Settings Transfer Rate` in `[07 BetterContinents.Misc]` (default KB512) is how fast the machine that runs the world sends a joining player the world's settings over their Steam connection. Valheim limits every Steam connection to about 150 KB/s. This setting raises the rate of that one connection for the transfer, then puts the game's own back.

Value	Rate
Vanilla	The game's own, about 150 KB/s
KB256, KB384, KB512 (default), KB768	256, 384, 512 or 768 KB/s
MB1, MB1_5, MB3	1, 1.5 or 3 MB/s

Value	Rate
Unlimited	100 MB/s is set; the transfer itself moves about 4 MB/s at most (3.85 MB/s measured)

- **Only the sending machine's value counts:** the dedicated server's, or the host's. A joining player's own value has no effect.
- **Steam sends at exactly the rate set,** with no congestion control, so pick a rate the server's upload can carry.
- **A new value applies to the next player who joins,** without a restart, because Better Continents reads the config again when it changes.
- **Crossplay connections** (PlayFab) cannot be raised, and are sent at the game's rate. The log says so once.

How long a download takes:

Settings sent	Vanilla	KB512 (default)	MB3	Unlimited
11.8 MB, measured on a local dedicated server	78 s	23 s		3 s
52.6 MB		about 1.7 minutes	about 17 s	
139 MB		about 4.4 minutes	about 44 s	
203 MB		about 6.5 minutes	about 65 s	

The rows from 52.6 MB down follow from the transfer rate.

The server's log says what each download should take, and what it took:

```
Settings transfer rate: 512 KB/s (server setting)
The <size> MB of settings take about <time> s to send at that rate.
Settings sent: <bytes> bytes in <time> s (<speed> KB/s)
```

Each join that needs a download also costs the server one long frame: about 0.3 s for 52.6 MB and 1.3 s for 203 MB, measured on one machine. A big world costs the joining player's machine time and memory as well, whatever the download: see World size and big worlds.

For a big world, raise the rate if the server's upload allows it. Better still, give players the world's `.bcworld` file, so that they join without a download.

Letting players skip the download

`bc_cache export` writes a world's settings as a file that players can import before they join. It is the same data a joining player would download, so a player who has imported it joins without a download.

1. **Make the file.** On the host, in the world, run `bc_cache export`. For a dedicated server's world, join it once yourself and run `bc_cache export` on your own machine: it writes your cached copy of the world, which is the same file. The file, `<World>-<id>.bcworld`, and a `.txt` description beside it are written to `BetterContinents/<World>/` beside `worlds_local`, and the console names the path.
2. **Give players the file.**
3. **Each player imports it.** Either drop it into `BepInEx/config/BetterContinents/seed/` (in the game folder, or in a mod manager's profile), and it is imported the next time the game starts, or run `bc_cache import <file>` once, in the main menu or in a world.
4. **They join.** The server finds the world's settings in their cache, and they skip the download.

> **Console**
> Command: `bc_cache [export | import <file or folder> | list | help]`
>
> `export` writes the world's settings as a `.bcworld` file and a `.txt` description: on the host, the loaded world's; on a player's machine, its cached copy of the world it is in. `import` adds one `.bcworld` file, or every one under a folder, to this machine's cache. `list` shows the cache: id, size and date. Any player can run it, and `import` and `list` work in the main menu too.

Things to know:

- **A mod pack can carry it.** When Better Continents starts, it imports every `.bcworld` file it finds under `BepInEx/plugins/` and in `BepInEx/config/BetterContinents/seed/`. A mod pack that holds the file pre-seeds every player who installs it, with nothing to type.
- **An import never overwrites** a cache entry, and a file's id is worked out from its content, never from its name.
- **After a change, make a new file.** Any change to the world's settings, such as a `bc` change on the host, gives them a new id, and the old file no longer matches.
- **Versions.** The file of a version 12 world is accepted by Better Continents 0.10 only: 0.9.x refuses it as a bad header.
- **Size.** A `.bcworld` file can be up to 1.5 GB.

Running a dedicated server

1. Install BepInEx and the same version of Better Continents on the server: see Install Better Continents.
2. Make the world on your own computer, in the New World screen, and check it there. A dedicated server cannot create a Better Continents world: a world it creates comes out vanilla. For a world whose World Size + Edge Size is past 16,350 m, your own `Wide Sectors` decides whether it is made wide (see World size and big worlds).
3. Copy the whole world folder, `worlds_local/<World>/`, to the server's `worlds_local`, including the file named `BetterContinents`.
4. Set `Settings Transfer Rate` in the server's own `BetterContinents.cfg`.
5. Start the server. When it loads the world, its log prints the world's settings, starting with `Version`.
6. For a big world, give players the world's `.bcworld` file.

The `bc` commands don't run on a dedicated server. To change its world, stop the server, change a copy of the world in single player, and put the folder back: see Changing an existing world. An admin can ask a dedicated server to export its own world with `bc_export server`, or to remove doubled vegetation with `bc_twins server`, and the answer comes back to the admin's own console: see Export and import and Forests and vegetation.

A joining player's own config

Setting	On a joining player's machine
The world's settings: its maps, `World Size` in `[01 BetterContinents.Global]`, `Heightmap Amount` in `[02 BetterContinents.Heightmap]` and the rest	The world's own, as the server sent them. The player's config has no say.
`Settings Transfer Rate`	Only the server's value counts for the join.
`Hud` and `Allow Export` in `[09 BetterContinents.Export]`	The server's values, while the player is connected to a server that runs Better Continents 0.9.0 or later.

Setting	On a joining player's machine
`Map Memory` and `Max Map Size` in `[07 BetterContinents.Misc]`, and the other Export settings	The player's own.
`Wide Sectors` in `[07 BetterContinents.Misc]`	No effect: the player's game follows the world, wide or not.
`Debug Mode` in `[00 BetterContinents.Debug]`	No effect: it acts only on the machine that runs the world.

With the Server Devcommands mod, a joined player can type `bc` commands, but a change there alters only that player's own copy of the settings and their own view of the ground, never the world or the other players, and zone regeneration refuses to run. `bc info` on a joined player's machine shows the settings the server sent.

Settings

Setting	Section	Default	What it does
`Settings Transfer Rate`	`[07 BetterContinents.Misc]`	KB512	How fast the machine that runs the world sends a joining player the world's settings
`Allow Export`	`[09 BetterContinents.Export]`	true	Whether players connected to this server may export the world
`Hud`	`[09 BetterContinents.Export]`	false	Shows the export HUD; on a player connected to a Better Continents server, the server's value is used

Next

- World size and big worlds: what a big world costs to load and to join.
- Export and import: exporting a server's world.
- Coming from 0.9.x or older: updating a server and its players.

16 Coming from 0.9.x or older

Updating Better Continents does not change your worlds: each world keeps generating as it was made. This chapter says which worlds get the new behaviour of 0.10, what changes for every world, what happens to your config, and what to watch for when versions are mixed. Some new worlds cannot go back to an older version: read Going back to an older version before you update a server.

Before you update

- Back up every world you care about: copy its whole folder, `worlds_local/<World>`.
- Update the server and every player together. On a Better Continents world they must all run the same version.
- A world made with Wide Sectors or with fine heights needs Better Continents 0.10.3 or later from then on, and so does a world whose `High Terrain` is On or Off.
- A world whose heightmap is read at an Amount above 5, with `High Terrain` on Auto, opens in an older version without the rules for tall worlds.
- A world made with compact maps cannot be read by Better Continents 0.9.

Which worlds get the new behaviour

Each world stores a settings version, and Better Continents follows that version's rules. A version 12 world gets the new-world behaviour of 0.10. A version 11 world keeps the rules of 0.9.4 and is saved as 0.9.4 saved it.

How the world was made	Its settings version
By Better Continents 0.9.x	11
By 0.10, From Config	12
By 0.10, from one of the four presets that ship with the mod (Continents, Pangaea, Swirls, Vanillaish), which are old version 7 files	11
By 0.10, from a preset saved by 0.9.x	11
By 0.10, from a preset saved by 0.10 from a version 12 world (`bc savepreset`)	12
From a world export: its preset, `bc_import`, or a `Directory` that holds its `export.cfg`	12 when the export's maps span what its own World Size + Edge Size give, as an export of a version 12 world or of a vanilla-size 0.9.x world does. Otherwise 11: an export of a world resized under 0.9.x, or of one made under Expand World Size at another size
From a folder of your own maps with no `export.cfg`	12
With `Override version` in `[00 BetterContinents.Debug]` set to 11	11

A folder of your own maps gives a slightly different world in 0.10 from the one 0.9.4 made from the same files: the pins land exactly, the start is placed before the pins, Heightmap Alpha blends, Biome precision goes finer, and Expand World Data's rules apply. To get the 0.9.4 result, set `Override version` to 11 before you create the world, and empty it again afterwards.

Most differences matter only when you use the feature. At the default World Size of 10,000 m and Edge Size of 500 m, with no Expand World Data, no Heightmap Alpha, Biome precision of 5 or less and no location map, a version 12 world generates the same as a version 11 one.

To see a world's version, use `bc info`: its first line is `Version 11` or `Version 12`. The log prints the same settings when the world loads. A world whose folder has no `BetterContinents` file is a vanilla world and has no version. Nothing converts an existing world from version 11 to 12.

What a version 12 world does differently

	Version 11 world	Version 12 world
What the maps span	21,000 m, whatever World Size says	2 × (World Size + Edge Size), the land dropping away over Edge Size; 21,000 m at the defaults
The world's layout: alt-biome grid, where locations go, the game's biome bands, Ashlands and Deep North rings, lakes, streams, minimap	Vanilla's; World Size only moves the game's edge	Laid out to the world's size; `Grid` in `[08 BetterContinents.AltBiomes]` has no effect
`Heightmap Alpha` in `[02 BetterContinents.Heightmap]` on	The heightmap is read as 8-bit grey, and its alpha is not used	16-bit grey with a 16-bit alpha, blended with the game's own terrain
`Biome precision` in `[03 BetterContinents.Biomemap]`	Up to 5 (cells of about 11 m); a higher value is stored but used as 5	Up to 31 (cells of 2 m)
Location map pins	At the corner of one of the pin's pixels, picked at random, up to a pixel south-west of where the other maps put it; locations that share a colour are dealt out at random	On the pixel nearest the pin's middle; locations that share a colour are dealt out the same way every time. A world export's pins are still read at the corner
The start override and a pin in the same 64 m zone	The pin wins, and the start is dropped	The start is placed first, and the pin is dropped
Expand World Data's altitude rules	The maps' heights replace them, except where the rough map is white	They apply over the maps' heights
Expand World Data's `lava: true` biomes, with a biome map and no heat map	Only the Ashlands are hot	They are hot too
Expand World Data's grounds	A terrain legend that names one of its biomes reads as a bad colour (a warning, and Meadows ground); Biome precision's colour replaces a territory's	The name gives that biome's ground colour; Biome precision keeps a territory's ground colour
Expand World Data's biomes on a biome map	Stored by number, so renumbering its yaml swaps them	Stored with their names, and followed if the numbers move; a biome it no longer has reads as None
Compact maps	Never	With `Compact Maps` on Auto, a new world is compact when a map is more than 8,192 px across or its heightmap has fine heights; On makes every new world compact; see `Compact Maps` below
A `heightmap-fine.png` beside the heightmap	Not read (the log says why)	Read: fine heights, unless `Compact Maps` is Off. See Fine heights
The world file	Saved as 0.9.4 saves it	Version 12. Better Continents 0.9.x reads it as version 11, and cannot read a compact one
A `.bcworld` file of the world	Accepted by 0.9.x and 0.10	Accepted by 0.10 only

Changes for every world

These reach every world, old or new. None of them changes what a world's settings generate, except the vegetation twin guard, which changes only the plants of zones that are not generated yet.

- **Maps take less memory.** Better Continents holds its maps in tiles of 128 × 128 pixels, and a tile of one value, such as open sea, as that one value. Every value reads as before. A real set of 10,501 px maps (a heightmap, a forest map and a biome map) took about 99 MB instead of about 1.32 GB.
- **The game's edge** (the kill zone, the ship's push-back and the water's edge) follows each world's own World Size, for every world loaded in a session. Before, a world loaded after another one could keep the first world's edge.
- **Water follows its ground.** When a zone's ground is rebuilt in place, its water takes the depth of the new ground.
- **The vegetation twin guard.** In zones generated from now on, a plant is not placed within 1 m of the same kind of object already there. Zones already generated keep what they have; `bc_twins` finds and removes the doubles a world already has. See Forests and vegetation.
- **Zone regeneration** follows every `bc` change and `bc regen`, in single player or on the host, and uses the world's own rules, so a version 11 world edited with `bc` keeps its 0.9.x rules. See Changing an existing world.
- **Exports.** `Default Heightmap Amount` in `[09 BetterContinents.Export]` is 1 (it was 2), and an exported `heightmap.png` records the Amount and Sea Level its heights are encoded for.

Your BetterContinents.cfg

0.10 reads your existing file as it is: every key keeps its name and section, and the new keys appear in it. Most settings are defaults for the worlds you create afterwards: an existing world uses what it stored.

Changed once

Each of these is changed once, the first time the game starts with the version that brings it, with a line in the log. The hidden `Config Version` in `[07 BetterContinents.Misc]` records it, so a value you set again afterwards stays.

Setting	Changed once	Why
`Default Heightmap Amount` in `[09 BetterContinents.Export]`	2 to 1	1 is Better Continents' default everywhere. Set 2 again to export at the old encoding, −30 m to 370 m.
`Debug Reset Command` in `[00 BetterContinents.Debug]`	`zones_reset start` to empty	Better Continents regenerates the zones itself. Write `zones_reset start` again to run Upgrade World's command instead.
`Forest Scale` in `[04 BetterContinents.Forest]`	1 to 0.5	0.5 is the game's own size of forest and clearing patches; 1 makes them about five times larger. Worlds already made keep their forests.

Gone

Fix Water Color and Ocean Channels in `[01 BetterContinents.Global]` are gone from the config, with the `bc g fixwatercolor` command. The first never did anything, and the second matters only to settings formats 1 to 6. A world that saved them still loads them.

New

Setting	Section	Values (default)	Since	What it does
`Compact Maps`	`[07 BetterContinents.Misc]`	Auto, On, Off (Auto)	0.10.0	Whether a new world keeps its maps compact. In 0.10.0 to 0.10.2 it was an on/off switch, off by default: an old value `true` reads as On, and `false` as Auto. See Image maps
`Map Memory`	`[07 BetterContinents.Misc]`	64 to 65536 MB (512)	0.10.0	How much decoded map data a compact world may keep on this machine
`Max Map Size`	`[07 BetterContinents.Misc]`	4096, 8192, 16384 (16384)	0.10.3	The largest map this machine reads from a picture
`Wide Sectors`	`[07 BetterContinents.Misc]`	Auto, On, Off (Auto)	0.10.3	Wide sectors for worlds past 16,350 m. See World size and big worlds
`High Terrain`	`[02 BetterContinents.Heightmap]`	Auto, On, Off (Auto)	0.10.3	The rules for tall worlds. See Heights and the heightmap
`Largest Size`	`[09 BetterContinents.Export]`	4096, 8192, 16384 (16384)	0.10.3	The largest export this machine makes

Coming from an older version still: the `[08 BetterContinents.AltBiomes]` section came with 0.8.1, and `Settings Transfer Rate` in `[07 BetterContinents.Misc]` and the `[09 BetterContinents.Export]` section with 0.9.0.

The world export's fine heights are not a setting: each export chooses them with `bc_export fine=auto|off|4|8` or the export window's *Fine heights* choice. See Export and import.

Changed

- `Heightmap Amount` in `[02 BetterContinents.Heightmap]` takes 0 to 81 (it took 0 to 5), and `Default Heightmap Amount` takes 0.01 to 81.
- `Default Size` in `[09 BetterContinents.Export]` takes 128 to 16384 (it stopped at 8192).
- `Start Position X` and `Start Position Y` in `[05 BetterContinents.StartPosition]` take −1,000,000 to 1,000,000 m (the config stopped at 10,500).
- `Biome precision` in `[03 BetterContinents.Biomemap]` takes 0 to 31. A version 11 world uses at most 5.
- `Spawnmap File` in `[06 BetterContinents.Maps]` is only the creature spawn map. It is no longer moved into `Locationmap File`, and no `spawnmap.png` is renamed to `locationmap.png`.
- A new file has the New World panel's dropdown on **Disabled**. An existing file that holds the old choice "Vanilla" keeps it, and still makes a world without Better Continents, with an error in the log.
- `Override version` is read at every save, so an edit changes the next autosave of a loaded world. Leave it empty.

Mixing versions

In multiplayer

On a world that uses Better Continents, the server and every player must run exactly the same version, whatever the world's settings version. Players play the settings the server sends, whatever their own config says. A player's cached copy of a version 11 world stays valid after the update, so it is not downloaded again.

A version 12 world opened in 0.9.x

> **Warning:** This is not supported. Copy the world folder first if you try it.

Better Continents 0.9.x reads a version 12 world as version 11: its maps span 21,000 m whatever World Size says, the world is not laid out to its size, the heightmap's alpha is read as 8 bits, and so on. Zones generated from then on follow the 0.9.x rules, so a world of another size than the default can show seams, while the pins already in the world stay. Its next save writes the world as version 11, and from then on 0.10 treats it as made by 0.9.x.

A compact world cannot be read by 0.9.x at all: it switches Better Continents off for the world, generates vanilla terrain that stays in the world, and leaves the `BetterContinents` file alone. A preset saved by 0.10 loads in 0.9.x as a version 11 world, and a compact one makes a world without Better Continents.

A 0.9.x world in 0.10, and back

With `Override version` empty, 0.10 writes a version 11 world's file exactly as 0.9.4 does, so the world opens in 0.9.x as before. `BetterContinents.old` beside the file holds only the save before the last, so back the folder up before you change versions.

Going back to an older version

> **Warning:** These worlds cannot go back to an older Better Continents. Back up the world folder before you update, and never open such a world with an older version, in single player or on a server.

A world…	In an older Better Continents
with Wide Sectors: made with 0.10.3 or later past 16,350 m, or converted	An older Better Continents stops at the world's wide marker (*Unknown feature: 68*) and opens it as a world without Better Continents. Its saves write objects twice, and they come back as two of each (two of every chest). This happens with 0.10.2 or older, and with the game without Better Continents. No setting can stop it.
made with fine heights	Better Continents 0.10.0 to 0.10.2 cannot read it (*a map saved in format 2, which this version of Better Continents cannot read*): they switch Better Continents off for the world, which gets vanilla terrain, and leave its settings file alone.
with `High Terrain` On or Off	Better Continents 0.10.2 and older stop at it (*Unknown feature*) and treat it as a world without Better Continents: it gets vanilla terrain.
whose heightmap is read at an Amount above 5, with `High Terrain` on Auto	It opens with its terrain, but without the rules for tall worlds. Among others, a player higher than 3,000 m counts as inside a dungeon, the grass finds no ground above 500 m, items, tombstones, plants and creature spawns find none above 2,000 to 6,000 m, and plants, creatures and locations keep their upper altitude limits.
made with compact maps	Better Continents 0.9 cannot read it, and it gets vanilla terrain. Versions 0.10.0 to 0.10.2 read it.

In multiplayer this can't happen by accident: a player with another version is refused when they join. The risk is the machine that runs the world.

Worlds from before Valheim 1.0

A Better Continents world from before Valheim 1.0, with its `.fwl`, `.db` and `.BetterContinents` files side by side, still loads. On its first save the game converts it to a world folder, and Better Continents moves its settings into the folder. The old file is kept, renamed `<world>.BetterContinents.pre10`, because it holds the only other copy of the world's maps.

Next

- How a world is made: settings versions and what a world keeps.
- World size and big worlds: Wide Sectors, before you make a big world.
- Multiplayer and dedicated servers: updating a server and its players.

17 Other mods

Better Continents works alongside other mods that touch the world. This chapter gives the facts for the ones players ask about: Expand World Size, Expand World Data, map mods and no-map worlds, Server Devcommands, Upgrade World, and mods that change how the game places plants.

Expand World Size

When Expand World Size is installed, it sizes the world and lays it out itself, as it always has: its size, its alt-biome grid, the minimap and where locations go. Better Continents leaves the layout to it, and Better Continents' maps span Expand World Size's size, in every world. The log says so when the game starts:

```
"Expand World Size" detected: it sizes and lays out the world.
```

When a world made by 0.10 has its own World Size and Edge Size, and they differ from Expand World Size's, the log also names both sizes.

- **Install it on every machine that plays the world, or on none.**
- **Don't add it to a world that Better Continents laid out** (a world made by 0.10 without it). Its size and layout would take the place of the ones the world was made with.
- **Keep its World Stretch at 1.** Any other value magnifies Better Continents' maps, so that only their centre shows. The log warns once, in any Better Continents world.
- **Exports.** An export made under Expand World Size at another size is rebuilt as it was only with Expand World Size installed at that size.
- **Wide Sectors.** With Expand World Size installed, the larger of its size and World Size + Edge Size decides whether a world needs Wide Sectors. See World size and big worlds.
- **Load times.** The faster loading of big worlds in 0.10.3 applies to worlds that Better Continents lays out, not to a world laid out by Expand World Size.
- **Its save grid (Expand World Size 1.44 and later).** For a world bigger than 16,384 m, Expand World Size widens the game's save grid itself, out to 65,536 m. Wide Sectors does the same job with a different chunk layout in the save, so the two never both run. While Expand World Size's grid is wider than the game's, a new world is not made with wide sectors (the log says that Expand World Size's save grid files its objects), `Wide Sectors` stays out of the way in every mode, and nothing is patched twice. A save that already has wide-sector chunks, made by Better Continents' Wide Sectors, is not loaded while Expand World Size's grid is in use: the game's load error is set, nothing is saved over the save, and a message names both mods, because the two layouts cannot read each other's saves. To keep such a world, remove Expand World Size, or set its save grid to the game's (Save system size 0, with a world no bigger than 16,384 m). With no Expand World Size, or a release before 1.44, nothing changes.

Expand World Data

Expand World Data adds biomes of its own, and rules for how high each biome's ground is. Better Continents works with both.

Its biomes on a biome map

- Name the added biomes in the biome map's legend, `biomemap.txt`, as Expand World Data's `expand_biomes` yaml does, for example `DeadWastes: 8B4513`. A name can be written in any case, with spaces, underscores or hyphens.
- Give each added biome a `terrain` in the yaml, for example `terrain: Plains`. Without one, Expand World Data gives it the game's default ground, flat and 30 m under water, with or without Better Continents.
- Install Expand World Data wherever the world is played, with the same yaml, and add new biomes at its end.
- A legend that Better Continents writes because it was missing names only the ten vanilla biomes: add the Expand World Data names yourself. A world export does it for you: it gives each added biome a colour of its own and names it in `biomemap.txt`.
- Where a biome map or a heat map says what a place is, it wins over Expand World Data's world yaml.
- Where a world's biome map holds a biome that Expand World Data does not define now (it is not installed, or its yaml changed), the game's default generation decides there, and the log says so. The map keeps the biome for when it is defined again.

In a version 12 world

In a world made by 0.10 (settings version 12), more of Expand World Data's rules reach the maps:

Rule	Version 12 world	Version 11 world
Altitude rules: a biome's or territory's altitude multiplier and delta, water depth, height limits and lava dip	Apply over the heightmap's heights, as over the game's own	The heightmap's heights replace them, except where the rough map is white
`lava: true` biomes, with a biome map and no heat map (or `Heatmap Scale` in `[06 BetterContinents.Maps]` at 0)	Hot where the biome map puts them: lava burns there, and the Ashlands weather applies	Only the Ashlands are hot
A terrain legend (`terrainmap.txt`) that names one of its biomes	Gives that biome's ground colour, as in `DeadWastes: 8B4513`	Reads as a bad colour: a warning, and Meadows ground
Biome precision	Keeps its territories' ground colours	Replaces them
Its biomes on a biome map	Saved with their names, and followed if Expand World Data numbers them differently later; a biome it no longer has reads as None	Saved by number, so renumbering the yaml swaps biomes

To keep a heightmap's heights exactly in a version 12 world, leave each biome's altitude settings at their defaults in Expand World Data. See Biomes and the biome map.

Current releases

Expand World Data 1.79 moved two of its classes, `WorldInfo` and `BiomeManager`, into namespaces. Better Continents finds them under the new and the old names, and, if a later release moves them again, by name and member. A Better Continents from before that change found neither with 1.79: Expand World Data then never learned the world's size, its lava biomes and terrain were not read, and a new world stored its biomes as numbers instead of names, with only warnings in the log. Once at each start, the log says what Better Continents reached:

```
"Expand World Data" <version> detected. Applying compatibility: size yes, biome names yes,
  lava and terrain yes, minimap height yes.
```

A *no* names a part that does not work with the release you run. Better Continents is checked against the real files of Expand World Data 1.73 and 1.79.

Map mods and no-map worlds

- **The minimap is finished before other mods see it.** Better Continents draws it on several CPU cores, up to 16, and the loading screen waits for it, as it does for the game's own map.
- **ZenMap:** cartography tables with hidden biomes, and the admin map, show the real land and sea, as crafted parchment maps do.
- **No-map worlds:** mods that show players a map only at a table or on a parchment, or that give admins their map back, get the finished map on every join.
- **Other map mods:** a mod that reads, copies or restyles the minimap when it loads sees the finished map, and mods that hook the game's map drawing run once, after it is done.

Server Devcommands

- It gives admin tools for testing a world, such as flying, god mode and a free camera.
- With Server Devcommands 1.51 or later installed, Better Continents adds tab completion for its `bc` command. With an older version, the log asks you to update it.
- It lets any player with `devcommands` on type `bc` on their own machine. On a player's machine that has joined someone else's world, a `bc` change alters only that player's copy of the settings and their own view of the ground, never the world or the other players, and zone regeneration refuses there: `Zone regeneration: it runs on the machine that runs the world: in single player, or on the host.` Make `bc` changes on the host or in single player.
- `bc info` on a joined player's machine shows the settings the server sent.

Upgrade World

Upgrade World has commands for resetting zones, placing locations (`location_register`) and regenerating parts of a world you are building. Better Continents answers the questions Upgrade World asks the game about biomes, heights, forests, locations and vegetation, so its commands follow this world's maps. A zone it resets generates again through the game, so the vegetation map, the twin guard and the baked layer apply as always. The notes below were checked offline, against Upgrade World 1.87's own files (its names, its config and the merge of `locations_restore`); they have not been run in a game yet. Upgrade World must be on the server and on the admin's client, as it needs anyway.

Better Continents regenerates the zones itself after a `bc` change and on `bc regen`. If you write a console command in `Debug Reset Command` in `[00 BetterContinents.Debug]`, for example Upgrade World's `zones_reset start`, Better Continents runs that command instead, after each change and on `bc regen`, and does not regenerate the zones itself. With the setting empty, the default, Better Continents' own regeneration keeps what players built and the zones around it, and adjusts the ground where kept and reset zones meet: see Changing an existing world.

A `BetterContinents.cfg` from 0.9.x held `zones_reset start` there as its default. Better Continents 0.10 empties it once, with a line in the log; a command you write there afterwards stays.

World radius

Upgrade World's own World radius (`[1. General]` in `upgrade_world.cfg`, default 10,500 m, the game's own world) is the reach of its commands: past it, they do not list the zones that are not generated yet (generated ones are still included). `Upgrade World Radius` in `[07 BetterContinents.Misc]`, Off by default, raises it when a world is loaded: with Upgrade World installed, its World radius is set to cover this world's World Size + Edge Size, rounded up to a whole zone (64 m), so that `zones_reset`, `zones_generate`, `locations_add`, `biomes_count`, `world_reset` and `upgrade ...` reach the outer ring of a world past 10,500 m. It is never lowered, and Upgrade World writes the new value to `upgrade_world.cfg` itself. With it Off, Upgrade World's setting is left as it is, and the log says once, on a world it does not cover, that this setting exists.

Locations

- `locations_restore`**.** Upgrade World builds a fresh location registry and merges it with the old one, zone by zone. Better Continents puts the location map's pins and the start position into the fresh registry, as the game's own `genloc` does, so a restore finds each pin where it is and keeps it. If a pin is ever missing after a restore, `bc genloc pins` puts it back (see Console commands).
- `locations_add`**, and** `upgrade tarpits`**,** `mountain_caves`**,** `bogwitch`**,** `bearcave`**,** `combatruins` **and** `hildir`**.** Upgrade World runs its own copy of the game's location search. Every question it asks the game is answered by Better Continents' biomes, heights and forest, and a point in the location exclusion map is refused. Two differences: an excluded point counts as one tried, so an excluded area can end with fewer locations than asked; and on a world with `Skip Default Locations` nothing is added, because only the pins exist.

Resetting by biome

`upgrade ashlands` and `upgrade deepnorth` choose their zones by distance from a point of the game's own layout (more than 11,600 m from a point 4,000 m north or south of the centre), not by biome. On a world with its own biome map or size that is not necessarily the Ashlands or the Deep North, and Better Continents says so once in the log when the command is given on such a world. To reset by biome, use `bc regen dry biome=AshLands`, then `bc regen biome=AshLands` (see Changing an existing world). `upgrade mistlands` selects by biome (the four corners of a zone) and so follows the biome map.

Other commands

- `world_gen` and `upgrade *_worldgen` change the game's world-generation version in memory. What Better Continents' maps decide does not change; the rest of the world does.
- `objects_remove`, `clean_*` and `prune_savesystem` delete objects directly, including a baked layer's live pieces in the zones they empty. This has not been tried with a baked world.
- `ChunkSaveFix` (Upgrade World 1.85 and later) assumes the game's own save grid of 64 chunks a side. With Wide Sectors on, its clean-up of emptied chunks has not been tried.

Mods that change how plants are placed

Better Continents patches the game's vegetation placement for the vegetation map and the twin guard. If another mod has changed that code so that the patch cannot be installed, the log says so, the vegetation map and the twin guard are off, and the rest of Better Continents still works.

Next

- World size and big worlds: World Size, Edge Size and Wide Sectors.
- Biomes and the biome map: drawing biome maps, with Expand World Data's biomes.
- Changing an existing world: zone regeneration in full.

18 Export and import

Better Continents can turn the world you are standing in, vanilla or Better Continents, into a set of maps that rebuild it, ready to edit in any image editor. It can also turn a folder of maps, an export or one you painted yourself, into a choice in the New World screen, without editing your config by hand. This chapter is the short version: the Export & Import guide in the package (`BetterContinents-Export-Guide.pdf`) walks through every step.

Exporting a world

Type `bc_export` in the console (F5), or use the export window. The export runs in the background while you play, and writes into its own folder:

```
<save data>/BetterContinents/<world name>/export-<date and time>/
```

That is beside the game's `worlds_local` folder, never inside the world's own folder. The export holds:

- the maps: `heightmap.png`, `biomemap.png`, `locationmap.png`, `forestmap.png`, `heatmap.png`, `coldmap.png` (where the Deep North is), `altbiomemap.png`, `lavamap.png`, `mossmap.png` and `paintmap.png`, with their legends, and `heightmap-fine.png` when the export writes fine heights (below). A world that has a location exclusion mask also gets `no_spawns_mask.png`;
- `export.cfg`: the Better Continents settings that rebuild the world from those maps;
- `world.json`: the world's radius, edge size and layout on one line, which the other mod BetterWorlds reads;
- `README.txt`: the ways to make a world from the export, step by step, and notes about this export;
- `manifest.json`: the export's numbers, for tools;
- `sources/`: a Better Continents world's own maps as it was made;
- a New World preset named after the world and the time, for exports up to 8,192 px (see Sizes, time and memory);
- with the `editor` option, an `editor/` folder: the ground as a 32-bit float picture, and masks, for Gaea and other terrain editors. `bc_import edited` brings the edited ground back as a new export folder. The Export & Import guide's chapter 10 has the steps.

`bc_export status` shows its progress, and `bc_export cancel` stops it and deletes its partial files and its preset. Console commands lists every option.

Who can export

`bc_export` is not a cheat command: every player can use it while the server's `Allow Export` in `[09 BetterContinents.Export]` is on. The machine that runs the world always may. A player's export holds only the locations the server has shown that player on the map, so export from the host, or ask a dedicated server with `bc_export server` (admins only), for a complete location map.

The export window

Set `Hud` in `[09 BetterContinents.Export]` to true. In a world, F9 then shows or hides a status box, and F7 opens the window, which frees the mouse while it is open. Its Export tab lists every map with what it holds, the size, the Heightmap Amount and the Sea Level the heights are encoded for, the Heat Scale, the *Fine heights* choice, and an estimate of the export's size, memory and time. Its Import tab lists your exports, with Make preset and Use as config buttons.

Heights in an export

The heights are encoded for `Default Heightmap Amount` (1) and `Default Sea Level` (0.5) in `[09 BetterContinents.Export]`, unless you choose others in the window or with `bc_export amount=` and `sealevel=`.

Amount	Heights stored (sea at 30 m)	Note
1 (default)	−30 m to 170 m	Higher ground is clipped, and the export says how many pixels
2	−30 m to 370 m	Keeps every vanilla mountain
81 (the most)	−30 m to 16,170 m	In steps of a quarter of a metre

`Default Heightmap Amount` takes 0.01 to 81, and `amount=` any Amount above 0 up to 81. Use the smallest Amount that holds the world's highest ground: the steps grow with the Amount (see Choosing an Amount).

`export.cfg`, `manifest.json` and a text record inside `heightmap.png` all carry the Amount and the Sea Level, so an import always reads the heights back right. Don't change `Heightmap Amount` in `export.cfg`: the pixels are encoded for the value written there. See Heights and the heightmap.

An export of a world whose `High Terrain` is On or Off writes that mode into `export.cfg` too. An export of a world on Auto, the default, writes no such line, and the new world takes your config's.

Sizes, time and memory

The window offers 1024, 2048, 4096, 8192 and 16384 pixels a side, and `bc_export` takes any size from 128 to 16384. `Default Size` (4096) is the size when you give none; 2048 and up keep the alt-biome map exact.

`Largest Size` in `[09 BetterContinents.Export]` (4096, 8192 or 16384; default 16384) is the largest export this machine makes. The window hides bigger sizes, and `bc_export` refuses them with a message that names the setting.

An export holds about 100 MB of memory while it runs, at any size, because each map is written as it is sampled: the window's estimate says about 85 MB at 1,024 px and 99 MB from 2,048 px up, and 112 MB there with a fine heights file. Measured on one machine, a dedicated server:

- A 16,384 px export took about 3 minutes on 8 workers, and an 8,192 px export 43 s. Fine heights added about 2% (186 s against 183 s at 16,384 px).
- A 16,384 px export of a plain procedural world (World Size 15,850 + Edge Size 500) was about 131 MB of PNGs. Noisy, hand-edited maps are larger.

The window's time is a rough guess from the size alone, from *well under a minute* up to 2,048 px to *a few minutes* at 16,384 px. A game that is also drawing the world takes longer than a server.

Making the New World preset takes memory over the game's own. Measured on one machine, with all nine maps: about 50 MB at 2,048 px, 200 MB at 4,096 px, 0.8 GB at 8,192 px and 1.4 GB at 16,384 px (19 s); the window's estimate follows these. So the export makes its New World preset itself only up to 8,192 px. Above that it leaves the preset to `bc_import`, and the window, its `README.txt` and `manifest.json` say so: run `bc_import` in the main menu to make it.

Every player who joins a world built from 16,384 px maps is sent those maps. With `Compact Maps` on Auto, the default, such a world is made compact (see Image maps), and Multiplayer and dedicated servers covers the download. To make a world from 16,384 px maps, `Max Map Size` in `[07 BetterContinents.Misc]` must be 16384, its default, on the machine that makes it.

Fine heights in an export

Each export can write fine heights: `heightmap-fine.png` beside `heightmap.png`, 8-bit grey, which refines every height below the 16-bit steps (see Fine heights). It is a choice for each export, not a setting: `bc_export fine=auto|off|4|8`, or the window's *Fine heights* choice of auto, off, 4 bits or 8 bits.

Choice	What the export writes
auto (default)	4 bits where one step of `heightmap.png`, 200 × Amount / 65,535 m, is more than 1.5 cm: from a Heightmap Amount of about 4.92, so an export at Amount 5 gets the file. Below that it writes none, and the folder is the one an export always made.
4, 8	4 or 8 bits at any Amount: a step refined to a sixteenth, or to a 256th. Below an Amount of about 4.92 a note says they add nothing worth their size.
off	No `heightmap-fine.png`.

`heightmap.png` is the same, byte for byte, either way. The fine value is 0 under water (below 30 m) and where the ground rises a whole step or more to a neighbouring pixel, so the file is small: 5.7 MB beside an 86.7 MB `heightmap.png` at 16,384 px (1.1% of the pixels), and 30 KB at 4,096 px. A fine file in which every byte is 0 is not kept, and a note says so.

The file carries its record after the pixels, with the bits it uses and the CRC-32 of the `heightmap.png` the export wrote:

```
Fine Format = 1; Fine Bits = 4; Heightmap CRC-32 = 2723AFF1
```

`manifest.json` gains `fineBits` and `fineEncoding` only when the file is written. A Better Continents world with fine heights also writes `sources/heightmap-fine.png`.

After you edit `heightmap.png`, delete `heightmap-fine.png` or export again: a fine file whose CRC doesn't match is not used, and the log says so.

> **Warning:** A world made from fine heights needs Better Continents 0.10.3 or later. Versions 0.10.0 to 0.10.2 can't read it, and give it vanilla terrain.

Making a world from an export

There are four ways, from the easiest:

1. **Pick the preset.** In the main menu choose Start Game, pick your character, then New World. Pick the export's preset (`<world> <date> <time>`) in the Better Continents box, and create the world. The preset is a copy of the folder as it was exported, so it keeps working if you move or delete the folder.
2. **After editing the pictures,** `bc_import`**.** Type `bc_import` in the console, in the main menu or in a world. It makes the preset again from your newest export as it is now, and selects it.
3. **To keep editing,** `bc_import <number> config`**.** It copies the folder's `export.cfg` into `BetterContinents.cfg` (your old file is kept beside it), with `Directory` set to the folder, and selects From Config. Every new world then reads the pictures as they are at that moment.
4. **Point** `Directory` **at the folder.** Set `Directory` in `[00 BetterContinents.Debug]` to the export folder and make the world From Config. The folder's `export.cfg` comes with the maps, laid over your `BetterContinents.cfg` for that world, without changing either file. Its settings win, so change a setting it lists in `export.cfg`, not in your config. The log lists what it changed.

`bc_import list` numbers your export folders, newest first. `bc_import <number>`, `bc_import <world>` and `bc_import <folder>` pick one, and any folder of Better Continents maps works, with or without an `export.cfg`. The export window's Import tab does the same with buttons.

> **Note:** Importing, like editing any map, changes only what the next new world is built from. A world that exists keeps the maps it was made with.

Which settings version the new world gets

A world made from an export, by any of the four ways, is a settings version 12 world when the export's maps span what its own World Size and Edge Size give: when `totalSize` in `manifest.json` equals 2 × (World Size + Edge Size). Every export of a version 12 world does, and so does an export of a world of the game's own size. Otherwise, as for an export of a world resized under Better Continents 0.9.x or laid out by Expand World Size at another size, the new world is version 11. See How a world is made.

Settings

The `[09 BetterContinents.Export]` settings apply at once. On a player connected to a server, the server's `Hud` and `Allow Export` are used instead of the player's own.

Setting	Default	Range or choices	What it does
`Hud`	false	true, false	Shows the export HUD: the status box and the window
`Allow Export`	true	true, false	Whether players connected to this server may export
`Hud Hotkey`	F9	a key, or None	Shows or hides the status box
`Window Hotkey`	F7	a key, or None	Opens or closes the window
`Default Size`	4096	128 to 16384	The export's size when none is given
`Default Heightmap Amount`	1	0.01 to 81	The Amount the heights are encoded for
`Default Sea Level`	0.5	0 to 1	The Sea Level Adjustment they are encoded for
`Largest Size`	16384	4096, 8192, 16384	The largest export this machine makes

Fine heights have no setting: choose them for each export (above).

Next

- `BetterContinents-Export-Guide.pdf`, in the package: every step, with pictures, including chapter 10 on sculpting the ground in Gaea or another terrain editor and using an export with BetterWorlds.
- Image maps: editing the exported pictures.
- Multiplayer and dedicated servers: sharing a world, and the `.bcworld` file.

19 Every setting

This chapter lists every setting in `BetterContinents.cfg`, section by section, in the order of the file. For each one it gives the scope, the default, the range or choices, what it does, and the chapter that explains it. The console column names the `bc` command that changes the setting in a loaded world, where there is one (see Console commands).

The scope says who reads a setting and when:

- **World**: a default for new worlds. It is stored in a world's settings when the world is made (From Config, a preset, `bc_import`, or a `Directory` with an `export.cfg`), and never read from the config again for that world. Editing it changes only worlds you make afterwards. In Debug Mode, the `bc` commands change the loaded world's own copy.
- **Live**: applies while the game runs, on this machine. Never part of a world, an export or an import. `Wide Sectors` is one too: the machine that makes a world reads it then, and the machine that runs a world reads it when the world loads.
- **Local**: this machine's own switches. Never part of a world, an export or an import.

Better Continents reads `BetterContinents.cfg` again whenever the file changes on disk, so you don't need to restart the game after an edit. A setting marked *hidden* is not shown in the Configuration Manager window.

[00 BetterContinents.Debug]

Setting	Scope	Default	Range or choices	What it does	Console
`Enabled`	World	true	true, false	Whether a new world made From Config uses Better Continents. A world made with it off gets none of the mod's features.	
`Debug Mode`	Local	false	true, false	For editing a world live with the `bc` commands. On the machine that runs the world, it turns devcommands on when you spawn, shows "Better Continents Debug Mode Enabled!" and adds a Better Continents button to the Esc menu and the large map, which opens the settings window (Alt+F8 there does the same). Leave it off for a world you play. See Changing an existing world.	
`Debug Reset Command`	Local	empty	a console command	Empty: Better Continents regenerates the zones itself after a `bc` change and on `bc regen`. A console command written here, such as `zones_reset start`, runs instead. See Changing an existing world.	
`Override version`	World	empty	a whole number	For testing only. Empty: every world is saved in its own settings version, and a new world gets 12. A number saves every world in that version, and 11 makes worlds as Better Continents 0.9 did. A world saved in an older version than its own loses what its version added. See Changing an existing world.	
`Directory`	World	empty	a folder	A folder of maps, each read by its standard name (`heightmap.png`, `biomemap.png` and so on). When it is set, the map file settings are not used. A folder with an `export.cfg` (a world export) also brings that file's settings, which win over the config's. See Image maps.	

[01 BetterContinents.Global]

Setting	Scope	Default	Range or choices	What it does	Console
`Skip Default Locations`	World	false	true, false	Skips the game's default location placement. The start temple and the location map's pins are still placed. See Locations, creatures and the start.	`bc g skipdefaultlocations`
`Continent Size`	World	0.5	0 to 1	Scales the features of the height layers' noise: 0 makes them about three times larger, 1 about five times smaller, and 0.5 leaves them as they are. It does not scale a heightmap. See Worlds from noise.	`bc g cs`
`World Size`	World	10000	no limit in the config; 0 to 1,000,000 in the console	The world's radius in metres. A new world is laid out to it: its maps span World Size + Edge Size each way. A world made by an older Better Continents keeps the game's 21,000 m, and World Size only moves its edge. See World size and big worlds.	`bc g worldsize`
`Edge Size`	World	500	no limit in the config; 0 to 1,000,000 in the console	How far the world's edge reaches past World Size, in metres. On a new world the land drops away over it.	`bc g edgesize`
`Sea Level Adjustment`	World	0.5	0 to 1	Moves the land against the sea: each 0.05 above 0.5 lowers the land by 20 m, and each 0.05 below raises it by 20 m. It applies where a heightmap or height layers make the terrain. See Worlds from noise.	`bc g sl`
`Ashlands Gap`	World	false	true, false	Whether to add the game's ocean gap around the Ashlands. Custom maps usually don't need it.	`bc g ag`
`Deep North Gap`	World	false	true, false	Whether to add the game's ocean gap around the Deep North. Custom maps usually don't need it.	`bc g ng`
`Rivers`	World	true	true, false	Whether the game makes its rivers. With a heightmap and `Heightmap Override All` on, they never appear.	`bc g r`
`Map Edge Drop-off`	World	true	true, false	Whether the land drops away at the edge of the world.	`bc g me`
`Mountains Allowed At Center`	World	false	true, false	Off: terrain within about 600 m of the world centre is held to 46 m above the water, blending to full height by 1,000 m. This applies to heightmaps too.	`bc g mc`

[02 BetterContinents.Heightmap]

Setting	Scope	Default	Range or choices	What it does	Console
`Heightmap File`	World	empty	a file	The heightmap: a square picture, 16-bit grey for smooth slopes, white highest. A `heightmap-fine.png` beside it (8-bit grey, the same size) adds finer steps: see Fine heights. Not used when `Directory` is set. See Heights and the heightmap.	`bc h fn`
`Heightmap Amount`	World	1	0 to 81	Multiplies the heightmap's values. At Sea Level Adjustment 0.5, Amount 1 gives heights of −30 m to 170 m (the sea is at 30 m), 2 gives −30 m to 370 m, and 81 gives −30 m to 16,170 m. With `High Terrain` on Auto, a world whose heightmap is read at an Amount above 5 also gets the high-terrain rules. At 81 one step of a 16-bit heightmap is 25 cm high, which fine heights make finer. See Heights and the heightmap.	`bc h am`
`Heightmap Blend`	World	1	0 to 1	How strongly the heightmap is blended into the result.	`bc h bl`
`Heightmap Add`	World	0	−1 to 1	How strongly the heightmap is added to the result. You usually want Blend instead.	`bc h ad`
`Heightmap Mask`	World	0	0 to 1	How strongly the heightmap acts as a mask on the normal height generation: it limits the height to the heightmap's.	`bc h ma`
`Heightmap Override All`	World	true	true, false	The world follows the heightmap exactly. Off, the game's own biome heights are made over the heightmap's, and the Mountain biome doubles what is over 80 m: a 1,000 m peak stands about 1,900 m.	`bc h ov`
`Heightmap Alpha`	World	false	true, false	A new world reads the heightmap at full precision and blends it with the game's own terrain by its alpha: where a pixel is transparent, the game's terrain shows. A world made by an older Better Continents keeps reading it at 8 bits, without the blend.	`bc h alpha`
`Roughmap File`	World	empty	a file	The roughmap: a grey picture of where the biomes' own rough ground shows through the heightmap (white) and where the heightmap stays smooth (black). Only used with `Heightmap Override All` off.	`bc r fn`
`Roughmap Blend`	World	1	0 to 1	How strongly the roughmap applies.	`bc r bl`
`High Terrain`	World	Auto	Auto, On, Off	Whether Better Continents adjusts the game's own rules for high ground, which assume land under about 400 m. Auto: a world whose heightmap is read at a Heightmap Amount above 5 gets the adjustments, and every other world keeps the game's own rules. On: every Better Continents world gets them. Off: none does. Off costs a world with high ground: no grass above 500 m, no plants, creatures or locations above about 1,000 m, no terrain, weather or building above 3,000 m, and no ground found above 6,000 m. A new world takes it when it is made and keeps it; the world stores it only when it is not Auto. See Heights and the heightmap.	`bc h ht`

[03 BetterContinents.Biomemap]

Setting	Scope	Default	Range or choices	What it does	Console
`Biomemap File`	World	empty	a file	The biome map: a colour per biome, named by the legend beside it (`biomemap.txt`, written with the default colours when missing). Not used when `Directory` is set. See Biomes and the biome map.	`bc b fn`
`Biome precision`	World	0	0 to 31	How closely ground textures, grass, vegetation and spawn points follow the biome borders inside each 64 m zone. 0 is the game's own: a zone takes its biomes from its 4 corners. N splits each zone into (N + 1) × (N + 1) cells, down to 2 m at 31. A world made by an older Better Continents uses at most 5. Heights don't change. See Biomes and the biome map.	`bc b p`
`Terrainmap file`	World	empty	a file	The terrain map: the ground colour, named by its legend (`terrainmap.txt`). Not used when `Directory` is set. See Ground colour, heat, lava and moss.	`bc terrain fn`

[04 BetterContinents.Forest]

Setting	Scope	Default	Range or choices	What it does	Console
`Forest Scale`	World	0.5	0 to 10	Size of the game's own forest and clearing patches: 0.5 is the game's own size, 0 about a third of it, 1 about five times larger. Keep it within 0 to 1. A forest map is not scaled. See Forests and vegetation.	`bc fo sc`
`Forest Amount`	World	0.5	0 to 1	How much forest there is, against clearings. 0.5 changes nothing; higher gives more forest.	`bc fo am`
`Forest Factor Overrides All Trees`	World	false	true, false	The forest factor, the game's or a forest map's, also decides where 13 kinds of tree, root and web grow that the game places without it.	`bc fo ffo`
`Forestmap File`	World	empty	a file	The forest map: a grey picture, white for more forest, combined with the game's own forest by Forestmap Multiply and Forestmap Add. Not used when `Directory` is set.	`bc fo fn`
`Forestmap Multiply`	World	1	0 to 1	How strongly the forest map scales the game's own forest factor.	`bc fo mu`
`Forestmap Add`	World	1	0 to 1	How strongly the forest map is added to the game's own forest factor.	`bc fo add`

[05 BetterContinents.StartPosition]

Setting	Scope	Default	Range or choices	What it does	Console
`Override Start Position`	World	false	true, false	Places the start at Start Position X and Y. The position is not checked. See Locations, creatures and the start.	`bc st os`
`Start Position X`	World	0	−1,000,000 to 1,000,000	The start's east–west position, in metres from the world's centre.	`bc st x`
`Start Position Y`	World	0	−1,000,000 to 1,000,000	The start's north–south position, in metres from the world's centre (north is positive).	`bc st y`

[06 BetterContinents.Maps]

Setting	Scope	Default	Range or choices	What it does	Console
`Locationmap File`	World	empty	a file	The location map: black, with a dot of colour for each location, named by `locationmap.txt`. See Locations, creatures and the start.	`bc l fn`
`Spawnmap File`	World	empty	a file	The spawn map: which creatures may spawn where, named by `spawnmap.txt`. White stops every spawner. See Locations, creatures and the start.	`bc spawn fn`
`Vegetationmap File`	World	empty	a file	The vegetation map: which plants may grow where, named by `vegetationmap.txt`. See Forests and vegetation.	`bc vegetation fn`
`Paintmap File`	World	empty	a file	The paint map: the paint over the ground, each colour as it is or as `paintmap.txt` maps it. See Ground colour, heat, lava and moss.	`bc paint fn`
`Lavamap File`	World	empty	a file	The lava map: a grey picture of where the Ashlands ground is lava (white).	`bc lava fn`
`Mossmap File`	World	empty	a file	The moss map: a grey picture of the Mistlands ground cover.	`bc moss fn`
`Heatmap File`	World	empty	a file	The heat map: a grey picture of the Ashlands heat, which decides where the Ashlands' weather, lava and ship damage are (anything above black).	`bc heat fn`
`Heatmap Scale`	World	10	0 to 100	Multiplies the heat map: white is this value. Most heat effects stop growing at 1. 0 switches the heat map off. See Ground colour, heat, lava and moss.	`bc heat sc`

Every file setting here is not used when `Directory` is set.

[07 BetterContinents.Misc]

Setting	Scope	Default	Range or choices	What it does	Console
`NexusID`	Local	446		Hidden. Not a setting to change.	
`SelectedPreset`	Local	Disabled	Disabled, From Config, or a preset's name	Hidden. The choice in the New World screen's Better Continents box, kept between sessions.	
`Settings Transfer Rate`	Live	KB512	Vanilla, KB256, KB384, KB512, KB768, MB1, MB1_5, MB3, Unlimited	How fast the host or dedicated server sends a joining player the world's Better Continents settings and maps over Steam. Vanilla leaves the game's own rate, about 150 KB/s; Unlimited sets 100 MB/s. Only the server's value counts. No effect on PlayFab (crossplay) connections. See Multiplayer and dedicated servers.	

Setting	Scope	Default	Range or choices	What it does	Console
`Compact Maps`	World	Auto	Auto, On, Off	Whether a new world keeps its maps as compressed tiles of 128 × 128 pixels, in memory, in its world file and in what it sends to joining players. Every map reads exactly as it would without. Auto: a new world is compact when any of its maps (the location and alt-biome maps aside) is more than 8,192 px across, or when its heightmap has fine heights. On: every new world of the newest settings version is compact. Off: no new world is, not even one of 16,384 px maps (the log warns), and a `heightmap-fine.png` is not read. Read only when a world is made; a world keeps it for good. An old value `true` reads as On, and `false` as Auto. Better Continents 0.9 cannot read a compact world. See Image maps.	
`Map Memory`	Live	512	64 to 65,536	For a compact world: how many megabytes of decoded map tiles this machine keeps. Past it, the tiles read least recently are dropped and decoded again when they are read. No effect on any other world. Applies at once.	
`Config Version`	Local	0		Hidden. Records which one-time changes Better Continents has made to this file.	
`Max Map Size`	Live	16384	4096, 8192, 16384	The largest map, in pixels across, that this machine reads from a picture file. A bigger picture is refused before it is read, with a line in the log, and the world is made without that map. A world's own stored maps are always read. Applies the next time a map is read. See Image maps.	

Setting	Scope	Default	Range or choices	What it does	Console
`Wide Sectors`	Live	Auto	Auto, On, Off	Whether a world whose World Size + Edge Size is past 16,350 m gets a sector and save chunks of its own for every zone out to 65.5 km, where the game gives everything beyond 16.35 km one shared sector. Auto: a new world past 16,350 m is made wide, and an existing world keeps the sectors it has. On: as Auto, and an existing world past 16,350 m that still has the game's sectors is converted when it loads, one way (the game saves a copy of the old save first). Off: a new world gets the game's sectors whatever its size; a world already wide stays wide. Read by the machine that makes a world, in the New World screen, and by the machine that runs it (single player, the host or a dedicated server) when it loads; a joining player's game follows the world. A world of 16,350 m or less is never touched. Never written to `export.cfg`. A wide world needs Better Continents 0.10.3 or later. Where Expand World Size's own save grid is wider than the game's (Expand World Size 1.44 and later, for a world past 16,384 m), that grid files the world's objects and Wide Sectors stays out of its way (see Other mods). See World size and big worlds.	
`Upgrade World Radius`	Live	Off	true, false	Whether the Upgrade World mod, when it is installed, is told this world's size: its World radius setting (default 10,500 m, the game's own world) is raised to cover this world's World Size + Edge Size, rounded up to a whole zone (64 m), so that its commands reach the outer ring of a world past 10,500 m. It is never lowered. Off: Upgrade World's setting is left as it is, and the log says once, on a world it does not cover, that this setting exists. Applies when a world loads. See Other mods.	

> **Warning:** `Wide Sectors` On converts an existing world for good, and a wide world must never be opened by an older Better Continents: its saves would write objects twice. Back up the world first. See World size and big worlds.

[08 BetterContinents.AltBiomes]

Every setting here is a default for new worlds. All but `Altbiomemap File`, `Mode` and `Grid` steer only the game's random placement: planted regions ignore them. See Alt biomes.

Setting	Scope	Default	Range or choices	What it does	Console
`Altbiomemap File`	World	empty	a file	The alt-biome map, which plants alt biomes with colours, named by `altbiomemap.txt` (written with a default colour per alt biome when missing). Black and transparent pixels are left to the game.	`bc ab fn`
`Mode`	World	Random	Random, PlantedOnly, Off	Random: the game's random placement on unplanted land, plus every planted region. PlantedOnly: only planted regions. Off: no alt biomes at all, planted ones included.	`bc ab mode`
`Grid`	World	WorldEdge	WorldEdge, Vanilla	WorldEdge: alt biomes, and the location candidates the game takes from the grid, stop at the edge of the world when it is inside 10,500 m. Vanilla: always the 10,500 m disc. No effect on a world laid out to its World Size.	`bc ab grid`
`Fixed Seed`	World	empty	a number or any text	The seed of the game's random placement. Empty: the world seed. A seed here gives the same random alt biomes whatever the world seed is.	`bc ab seed`
`Chance Multiplier`	World	1	0 to 10	Multiplies every alt biome's chance (the game's 0.2, and Dark Meadows' 0.5).	`bc ab chance`
`Amount Multiplier`	World	1	0 to 10	Multiplies every alt biome's minimum and maximum number of regions.	`bc ab amount`
`Region Size Scale`	World	1	0.1 to 50	Multiplies every alt biome's region size window. The game gives alt biomes only to regions about 0.1 to 2.4 km across; hand-drawn maps with big regions usually need 2 to 10.	`bc ab regionscale`
`Distance Scale`	World	1	0 to 10	Multiplies every alt biome's minimum distance from the world centre (500 to 2,000 m in the game) and its world bounds.	`bc ab distancescale`
`Min Sector Thickness`	World	0	0 to 20	Regions thinner than this (area / edge length, in 12 m cells) never get a random alt biome. 0 is the game's own. 2 filters the slivers an anti-aliased biome map leaves along its borders.	`bc ab thickness`

Setting	Scope	Default	Range or choices	What it does	Console
`Mean Sector Height`	World	false	true, false	Measures a region's height as the mean over the whole region instead of the game's (lowest + highest) / 2 of its border. Helps maps whose biome paint runs out into the sea.	`bc ab meanheight`
`Fix Neighbour Check`	World	false	true, false	Uses a corrected version of the game's required and forbidden neighbour test, which is broken. No alt biome of the game's uses it; ones from other mods may.	`bc ab neighbourfix`
`Overrides`	World	empty	`Name: key=value, key=value; Other Name: key=value`	Per-alt-biome overrides of the random placement. A name may hold `*` wildcards, and `*` alone means all. Keys: enabled, chance, min, max, mindist, minedge, maxedge, minheight, maxheight, ignorebounds.	`bc ab set`, `bc ab clear`

[09 BetterContinents.Export]

These settings apply at once. On a client connected to a server, the server's `Hud` and `Allow Export` are used instead of the client's own. See Export and import.

Setting	Scope	Default	Range or choices	What it does	Console
`Hud`	Live	false	true, false	Shows the world export HUD in game: a status box (Hud Hotkey) and a window (Window Hotkey) with an Export tab and an Import tab.	
`Allow Export`	Live	true	true, false	Whether players connected to this server may export the world. The machine that runs the world always may.	
`Hud Hotkey`	Live	F9	a key, or None	Shows or hides the export HUD's status box (with no Shift, Ctrl or Alt held).	
`Window Hotkey`	Live	F7	a key, or None	Opens or closes the export and import window, which frees the mouse while it is open.	
`Default Size`	Live	4096	128 to 16384	Pixels per side of an export when `bc_export` is given no size, and the window's first choice. 2048 and up keep the alt-biome map exact. A value above Largest Size is used as Largest Size.	
`Default Heightmap Amount`	Live	1	0.01 to 81	The Heightmap Amount an export encodes its heights for, unless `bc_export` or the window says otherwise. `export.cfg` records it, so an import reads the heights back right.	
`Default Sea Level`	Live	0.5	0 to 1	The Sea Level Adjustment an export encodes its heights for, unless `bc_export` or the window says otherwise.	
`Largest Size`	Live	16384	4096, 8192, 16384	The largest export this machine makes: the window offers sizes up to it, and `bc_export` refuses bigger ones with a message that names this setting.	

Fine heights in an export are not a setting: each export chooses them, with `bc_export fine=auto|off|4|8` or the window's *Fine heights* choice (auto by default). See Export and import.

Next

- Console commands: every `bc` command.
- Changing an existing world: which settings reach a world that already exists.

20 Console commands

This chapter lists every Better Continents console command: its syntax, what it does and where it works. Open the console with F5.

Better Continents has these commands (`bc_bake`, `bc_water` and `bc_lake`, and the `bc bake`, `bc water` and `bc lake` words under `bc`, are described in the Baking guide and in the Water and lakes guide):

- `bc`, for building and changing worlds. It is a cheat command.
- `bc_altbiomes`, `bc_export`, `bc_import`, `bc_cache` and `bc_twins`, which every player can use, each with its own rules about who may do what.

Where the commands work

Command	Needs devcommands	Where it works
`bc`	yes	On the machine that runs the world: single player, or the host of a multiplayer game. Not in the main menu.
`bc_altbiomes`	only for `list` on a client	In a world, for every player. `bc_altbiomes names` works in the main menu too.
`bc_export`	no	In a world, for every player the server allows (`Allow Export`). The machine that runs the world always may. `bc_export server` asks a dedicated server (admins only).
`bc_import`	no	For every player, in the main menu and in a world.
`bc_cache`	no	For every player. `export` needs a world; `import` and `list` work in the main menu too.
`bc_twins`	no	In a world, for every player. Removal runs where the world is; `bc_twins server` asks a dedicated server (admins only).

`Debug Mode` in `[00 BetterContinents.Debug]` turns devcommands on for you when you spawn, on the machine that runs the world. You can also type `devcommands`. See Changing an existing world.

With the Server Devcommands mod, a player who has joined a server can type `bc` on their own machine. A change made there alters only that player's copy of the settings and their own view of the ground, never the world or the other players, and zone regeneration refuses to run there. Make `bc` changes on the host or in single player. `bc info` on a joined player's machine shows the settings the server sent.

When an admin sends `bc_export server ...` or `bc_twins server ...` to a dedicated server, the answer comes back to the admin's console as well as going to the server's log.

How `bc` works

`bc` is a tree of groups, commands and values. `bc` on its own, or `bc <group> help`, lists what is under it, with each value's range and current value:

```
bc h help
```

A value without an argument prints its current value; with an argument it sets it:

```
bc g sl          prints Sea Level Adjustment
bc g sl 0.55     sets it to 0.55
```

Numbers take a point as the decimal mark. Switches take `true` or `false`. A value chosen from a list takes one of its names, in any case: `bc h ht on`.

A `bc` value changes the loaded world's own settings, which are saved with the world. Each change rebuilds what it affects (the minimap, the alt-biome grid and the loaded terrain), then Better Continents

regenerates the world's zones, except the ones around what players built or worked on. Changing an existing world explains what is kept. Use the `bc` commands on a test world, or back the world up first.

The same tree is shown as a settings window: in Debug Mode, the Better Continents button in the Esc menu or on the large map opens it, and Alt+F8 does the same while either is open.

World information

Command	What it does
`bc info`	Prints the loaded world's Better Continents settings. The first line is the settings version, `Version 11` or `Version 12`. Some printed values are worked out from the settings, so they don't always match the config or the console's names.
`bc locs`	Prints how many of each location the world has, and warns when one of these is missing: StartTemple, Eikthyrnir, GDKing, GoblinKing, Bonemass, Dragonqueen, Vendor_BlackForest and Mistlands_DvergrBossEntrance1.
`bc bosses`	Pins those eight locations on the map.
`bc show [prefix ...]`	Pins every location whose name starts with one of the prefixes (not case-sensitive). With no prefix it pins every location in the world.
`bc hide [prefix ...]`	Removes the pins that `bc show` and `bc bosses` put there: for those prefixes, or all of them with no prefix. Your own pins stay.

Regenerating the world

Console

Command: `bc regen`

Regenerates the zones now, the way every `bc` change does. Every generated zone is reset and generates again with the current settings when somebody comes near, except the zones within one zone of something a player built or worked on (a field, a path, levelled ground), a tombstone, a player connected from another machine, or you inside a dungeon. A location that reaches into one of those zones is kept whole. Players, what players built, tombstones and tamed animals are never removed, and the loaded terrain is left as it is. Words after it reset only part of the world: `bc regen [dry] [biome=<names>] [min=<m>] [max=<m>] [pos=<x>,<z>] [zone=<x>,<z>] [chance=<0..1>]`. Start with `dry`, which counts and changes nothing. See Changing an existing world. If `Debug Reset Command` holds a console command, that command runs instead, and the words are refused.

bc genloc

Console

Command: `bc genloc [alt | pins | help]`

Makes the world's locations again, the way the game's own `genloc` does, and does it also when the game's own run is stuck or the world was loaded with its locations marked as made. It says what it found (how many locations the world has, how many are placed and how many are not), runs the game's location generation, and when the run ends it says what changed: how many locations there were before and after, which types changed in number, how many wait for a player to come near, which locations ended short of their quantity, and anything Better Continents could not place from the location map or the start position. It then sends the new location icons to the players.

Placed locations and every zone that is generated stay as they are. Only the locations that are not placed yet are cleared and rolled again, and only in zones that are not generated yet; the new ones are placed as players reach their zones. The location map's pins and the start position go back in first.

Stay in the world until the console says the run is done, because a save in the middle keeps only the placed locations.

If the game's slot for the run is taken and nothing has run in it for a few seconds (a run that threw an error and was abandoned leaves it so, and the game's own `genloc` then does nothing), the command stops that run, frees the slot and starts a new one. A run that is alive is left alone and no second one starts. The command runs where the world runs: single player, the host, or the dedicated server. On a joined player's machine it says so and does nothing. In the main menu it says that no world is loaded.

Console

Command: `bc genloc alt`

Passes to the game's alt-biome branch, `genloc alt`: the game makes the alt biomes' placement again. It generates no locations.

Console

Command: `bc genloc pins`

Puts only the location map's pins and the start position into the location registry, and generates nothing. A pin that is there already stays as it is, and the console says how many it put in. It is for a registry that something else emptied, for example after Upgrade World's `locations_restore` (see Other mods). A world with no location map and no start position override has no pins, and the command says so. `bc genloc help` says all this on the console.

Console

Command: `bc reset`

Redraws the minimap (which reveals all of it), rebuilds the alt-biome grid and the loaded terrain, then regenerates the zones as `bc regen` does. Most changes do all of this for you.

Zone regeneration runs only in a world that uses Better Continents. In a world without it, `bc regen` says so and changes nothing.

Maps

Reloading a map

Console

Command: `bc reload <map>`

Reads a map's picture and legend again from its file and applies it to the loaded world, then rebuilds and regenerates as any change does. For most maps, if the file the world read is gone, the file the config names now is used instead.

`<map>`	**Map**
`hm`	heightmap
`rm`	roughmap
`fm`	flat map (only worlds of settings version 6 or older have one)
`bm`	biome map
`ab`	alt-biome map; also replants the alt biomes
`terrain`	terrain map
`paint`	paint map

<map>	Map
heat	heat map
lava	lava map
moss	moss map
fom	forest map
vegetation	vegetation map
lm	location map
spawn	spawn map
all	every map the world has

Only the maps the world has are offered, and bc reload exists only in a world with at least one map. A biome map is replaced only when its picture and its legend both read without errors; otherwise the world keeps the one it has, and the log says why.

Choosing a map's file

Each map's group has a value fn, the file the map comes from: bc h fn, bc r fn, bc b fn, bc terrain fn, bc fo fn, bc heat fn, bc l fn, bc spawn fn, bc vegetation fn, bc paint fn, bc lava fn, bc moss fn and bc ab fn. Give it a full path, a folder (the map's standard file name in it is used), or a file name in the Directory. Without an argument it prints the file in use. To switch a map off, clear its field in the settings window.

Settings

Each group below holds the settings of one config section, by short names. Every setting gives each setting's range and what it does.

Group	Values and the settings they change
bc g	skipdefaultlocations Skip Default Locations, cs Continent Size, worldsize World Size, edgesize Edge Size, sl Sea Level Adjustment, ag Ashlands Gap, ng Deep North Gap, r Rivers, me Map Edge Drop-off, mc Mountains Allowed At Center
bc h	fn the heightmap's file, am Heightmap Amount, bl Heightmap Blend, ad Heightmap Add, ma Heightmap Mask, ov Heightmap Override All, alpha Heightmap Alpha, ht High Terrain
bc r	fn the roughmap's file, bl Roughmap Blend
bc b	fn the biome map's file, p Biome precision
bc terrain	fn the terrain map's file
bc fo	fn the forest map's file, sc Forest Scale, am Forest Amount, ffo Forest Factor Overrides All Trees, mu Forestmap Multiply, add Forestmap Add
bc st	os Override Start Position, x Start Position X, y Start Position Y
bc heat	fn the heat map's file, sc Heatmap Scale
bc l, bc spawn, bc vegetation, bc paint, bc lava, bc moss	fn the location, spawn, vegetation, paint, lava or moss map's file
bc ab	listed under Alt biomes, below

A few changes need more than a value:

- bc fo ffo takes effect the next time the world is loaded.
- bc h alpha reads the loaded heightmap again at once, by the world's own rule.

- `bc h ht` takes `auto`, `on` or `off`. It switches the game's height rules at once and regenerates the zones, but leaves the loaded terrain, the noise and the minimap as they are; in a world whose land can pass 9,000 m it places the alt biomes again too. Players already connected keep the mode until they join again. See Heights and the heightmap.
- `bc h fn`, `bc reload hm`, `bc reload all` and `bc h alpha` also read a `heightmap-fine.png` beside the heightmap's file (see Fine heights).
- `bc b p` rebuilds the loaded terrain and grass and regenerates the zones; it does not rebuild the minimap or the alt-biome grid.
- The start position is used when the world's locations are placed, when the world is made, so `bc st` does not move the start of a world that exists.

Height layers

`bc hl` builds the height layers: stacked noise that makes terrain without a map, or adds to a heightmap. The layers are numbered from 0. Worlds from noise explains how they combine.

Command	What it does
`bc hl add`	Adds a layer at the end.
`bc hl <n> npreset def`, `bc hl <n> npreset ri`	Sets layer n's noise to the default noise, or to ridged noise.
`bc hl <n> n <name> [value]`	Layer n's noise settings, by the names in the next table.
`bc hl <n> nw on true`	Adds a warp to layer n's noise, set under `bc hl <n> nw`. `on false` removes it.
`bc hl <n> mpreset def`, `25%`, `ri`	From layer 1 on: gives the layer a mask preset.
`bc hl <n> m on true`	From layer 1 on: gives the layer a mask, set under `bc hl <n> m`. `on false` removes it.
`bc hl <n> mw on true`	For a layer with a mask: adds a warp to the mask, set under `bc hl <n> mw`.
`bc hl <n> delete`	Deletes layer n.
`bc hl <n> down`	Swaps layer n with the next one (not for the first or the last layer).

The noise, warp and mask groups share these values. Some appear only where they apply: the fractal values when there is a fractal type, `ps` for the ping-pong type, the `c` values for cellular noise, and `dt` and `da` for warps.

Value	Setting	Value	Setting
`nt`	noise type	`dt`	domain warp type
`fq`	frequency	`da`	domain warp amplitude, 0 to 20,000
`asp`	aspect ratio	`in`	invert
`ft`	fractal type	`ust`, `sts`, `ste`	smooth threshold: on, start, end
`fo`	fractal octaves, 1 to 10	`uth`, `th`	threshold: on, value
`fl`	fractal lacunarity, 0 to 10	`ura`, `ras`, `rae`	range: on, start, end
`fg`	fractal gain, 0 to 2	`uop`, `op`	opacity: on, value
`ws`	weighted strength, −2 to 2	`blm`	blend mode
`ps`	ping-pong strength, 0 to 10	`cf`, `ct`, `cj`	cellular distance function, return type, jitter

A value chosen from a list (`nt`, `ft`, `cf`, `ct`, `dt` and `blm`) takes the name of one of its choices, in any case. The settings window lists the choices, and shows a preview of each layer.

Alt biomes

`bc ab` holds the alt-biome report and the world's alt-biome options. A change redoes only as much of the alt-biome grid, regions and placement as it affects, then regenerates the zones. Connected players keep their placement until they reconnect. See Alt biomes.

Command	What it does
`bc ab info`	A summary of this world's alt biomes: why each did or did not place, what each legend line planted, and up to 40 planted regions.
`bc ab list [filter]`	Up to 60 regions with alt biomes. The filter is `planted`, `random`, `all`, a biome name, or part of an alt-biome name.
`bc ab here`	The region you stand in, what was planted on it, and how each alt biome of its biome fared.
`bc ab show [filter]`	Pins every region that has an alt biome, or those whose alt biome matches the filter.
`bc ab hide`	Removes the pins `bc ab show` added.
`bc ab export`	Writes `altbiomes-<time>.png`, the region grid with alt-biome regions tinted and planted ones striped, and a `.txt` report, to `<save data>/BetterContinents/<world name>/`.
`bc ab hash`	This machine's grid and placement hashes.
`bc ab names [filter]`	The game's alt biomes, with their default legend colours and random-placement rules.
`bc ab rebuild`	Rebuilds the grid, the regions and the placement from the current settings.
`bc ab fn [file]`	Sets the alt-biome map and reads it with its legend.
`bc ab mode [Random|PlantedOnly|Off]`	`Mode`
`bc ab grid [WorldEdge|Vanilla]`	`Grid`
`bc ab seed [value]`	The placement seed: a number or any text; `world` uses the world seed.
`bc ab chance [0 to 10]`	`Chance Multiplier`
`bc ab amount [0 to 10]`	`Amount Multiplier`
`bc ab regionscale [0.1 to 50]`	`Region Size Scale`
`bc ab distancescale [0 to 10]`	`Distance Scale`
`bc ab thickness [0 to 20]`	`Min Sector Thickness`
`bc ab meanheight [true|false]`	`Mean Sector Height`
`bc ab neighbourfix [true|false]`	`Fix Neighbour Check`
`bc ab set <name|pattern|*> <field> <value>`	Sets one override field. `default` clears it.
`bc ab clear [name|pattern|*]`	Clears one entry's overrides, or every override when no name is given.
`bc ab overrides`	Prints the world's overrides.
`bc ab reroll [seed]`	Sets a new fixed placement seed, random or the one given, and places the alt biomes again.

The override fields are `enabled`, `chance`, `min`, `max`, `mindist`, `minedge`, `maxedge`, `minheight`, `maxheight` and `ignorebounds`.

The minimap, pictures and presets

Command	What it does
`bc clouds`	Turns the clouds over the large map off or on.
`bc mapds [0 to 3]`	How finely the minimap is redrawn after a change: at 1/2, 1/4 or 1/8 of the game's size per side for 1, 2 and 3, full size for 0. The default is 2. A smaller picture is redrawn faster.
`bc scr [size]`	Saves the large map as a square PNG, 2048 px unless you give a size, to `<save data>/BetterContinents/<world name>/`, named by the date and time.
`bc savepreset [name]`	Saves the loaded world's settings and maps as a preset named `<name>`, or after the world when you give no name, with a 256 px picture of the map, in `<save data>/BetterContinents/presets/`. An older preset of that name is kept, with `.old-<time>` added to its name.

Export, import, cache and twins in the `bc` tree

The host's `bc` tree has the same tools that every player has, with the same options:

Command	Same as
`bc export run [options]`, `bc export status`, `bc export cancel`	`bc_export [options]`, `bc_export status`, `bc_export cancel`
`bc import run [what] [config]`, `bc import list`, `bc import status`	`bc_import [what] [config]`, `bc_import list`, `bc_import status`
`bc cache ...`	`bc_cache ...`
`bc twins ...`	`bc_twins ...`

Commands for every player

bc_altbiomes

Console

Command: `bc_altbiomes [here | names [filter] | hash | list [filter]]`

On a client or a host. On its own: the hashes and whether they agree with the server's, and the region you stand in. `here` shows that region in full; `names` lists the alt-biome names and their default colours (in the main menu too); `hash` shows the hashes; `list` shows up to 200 regions with alt biomes. Because `list` reveals where every alt biome is, a client needs devcommands for it.

bc_export

Console

Command: `bc_export [status | cancel | help | server <options> | <options>]`

Exports the loaded world as maps that rebuild it, into `<save data>/BetterContinents/<world name>/export-<time>/`. `status` shows its progress, `cancel` stops it and deletes its partial files and preset, and `help` prints the options and the defaults in use. `server <options>` asks a dedicated server to export its own world into its own save folder (admins only); `bc_export server status` and `bc_export server cancel` work the same way.

Option	What it means
a number, or `size=<n>`	Pixels per side, 128 up to `Largest Size` (16,384 by default). The default is `Default Size`, 4096.

Option	What it means
`amount=<value>`	The Heightmap Amount the heights are encoded for: above 0, up to 81. The default is `Default Heightmap Amount`, 1.
`sealevel=<value>`	The Sea Level Adjustment they are encoded for, 0 to 1. The default is `Default Sea Level`, 0.5.
`heatscale=<value>`	The Heatmap Scale the heat is stored for: above 0, up to 100. The default is 10.
`edge=auto`, `edge=on`, `edge=off`	How the world's edge is stored. The default is auto.
`forest=additive`, `forest=exact`	How the forest is stored: additive, black is the game's own forest; exact, black is no forest. The default is additive.
`fine=auto`, `fine=off`, `fine=4`, `fine=8`	Whether the export writes `heightmap-fine.png` beside the heightmap, and with how many bits. auto, the default, writes 4 bits from a Heightmap Amount of about 4.92 and none below. See Export and import.
`noheight`, `nobiomes`, `nolocations`, `noforest`, `noheat`, `nocold`, `noaltbiomes`, `nolava`, `nomoss`, `nopaint`	Leave that map out. `nocold` leaves out `coldmap.png`, where the Deep North is.
`nosources`	Leave out the `sources/` folder.
`nopreset`, `preset`	Don't make the New World preset, or make one. A dedicated server makes none unless told to.
`perpoint`	Sample heights one point at a time instead of the way the game builds terrain. For testing only.
`editor`, `editoronly`, `eformat=`, `eunit=`, `eamount=`, `esize=`, `emasks=`, `eprotectpad=`, `eprotectfeather=`	The editor pack: an `editor/` folder with the ground as 32-bit floats and masks, for Gaea and other terrain editors. Off unless `editor` is given. `bc_export help` lists them; the Export & Import guide's chapter 10 explains them.

A client's export holds only the locations the server has shown that player on the map. Export from the host or the dedicated server's console for a complete location map. See Export and import.

bc_import

> **Console**
>
> Command: `bc_import [list | status | help | edited | <number> | <folder> | <world>] [config]`
>
> Makes a New World preset from an export folder, or from any folder of Better Continents maps, and selects it. On its own it takes your newest export: of the loaded world, or else the last one. `list` numbers the export folders, newest first, for `bc_import <number>`. `<world>` takes that world's newest export, and `<folder>` a full path or one under `BetterContinents/`; spaces need no quotes. A trailing `config` copies the folder's `export.cfg` into `BetterContinents.cfg` instead (your old file is kept beside it) and selects From Config. `status` shows how the import is going. `edited` brings back the ground that Gaea or another terrain editor made from an export's `editor/` folder, as a new export folder `<export>-edited`: see the Export & Import guide's chapter 10 (`bc_import edited help` lists its words).

bc_cache

> **Console**
>
> Command: `bc_cache [export | import <file or folder> | list | help]`
>
> `export` writes the world's settings and maps as a shareable `.bcworld` file, with a `.txt` description beside it, in `<save data>/BetterContinents/<world name>/`: on the host or a dedicated server from the world itself, on a client from its cached copy of the world it is in. `import` adds one `.bcworld`

> file, or every one under a folder, to your cache; an entry you already have is skipped, never overwritten. `list` shows the cache. See Multiplayer and dedicated servers.

bc_twins

> **Console**
> Command: `bc_twins [radius] | remove [radius] | remove confirm [radius] | server <the same> | help`
>
> Finds and removes vegetation twins: two copies of one plant on one spot, or within the radius (1 m unless you give one, 0.1 to 5 m). On its own it counts them; a client counts only the area around it. `remove` shows what a cleanup would delete and deletes nothing: exact twins only, or near ones too when you give a radius. `remove confirm` deletes the extra copies, 1,000 a frame. Removal runs only where the world is; on a dedicated server, use `bc_twins server ...` (admins only). See Forests and vegetation.

> **Warning:** `bc_twins remove confirm` cannot be undone. Back up the world first.

Next

- Changing an existing world: what a `bc` change keeps and what it regenerates.
- Every setting: the settings behind the `bc` values.

21 Troubleshooting and FAQ

This chapter answers the questions people ask most, and walks through the common problems: where to look, what the log says, and what to change. Most answers end with a link to the chapter that explains more.

Finding things

Where are my worlds?

In Valheim 1.0 each local world is its own folder in `worlds_local`:

- Windows: `%AppData%\..\LocalLow\IronGate\Valheim\worlds_local`
- Linux: `~/.config/unity3d/IronGate/Valheim/worlds_local`

To copy or move a world, copy the whole world folder, including the file named `BetterContinents` (no extension) inside it. That file holds the world's Better Continents settings and maps.

Where is the log?

Better Continents starts each of its log lines with `[BetterContinents]`. Look in BepInEx's log, `BepInEx/LogOutput.log`, or in the game's own `Player.log`, which is in `%AppData%\..\LocalLow\IronGate\Valheim` on Windows.

When a world is made or loaded, the log prints the world's full settings, the same lines as `bc info`. The first of them is the settings version, `Version 11` or `Version 12`.

Where is BetterContinents.cfg?

See Install Better Continents.

The world is not what I made

The menu says Better Continents is DISABLED for my world.

The Esc menu shows "Better Continents v... ENABLED for this world" or "DISABLED for this world". DISABLED means one of these:

- The world was made with the Better Continents box in the New World screen set to Disabled, which is what a new config selects. Choose From Config, or a preset, when you make the world.
- `Enabled` in `[00 BetterContinents.Debug]` was false when the world was made From Config.
- A dedicated server made the world. A server never makes a Better Continents world: it starts a world that doesn't exist yet as a vanilla one, and its log says `Couldn't find loaded settings for this world at ..., mod is disabled`. Make the world on your own computer and copy its folder to the server (see Multiplayer and dedicated servers).
- The world's folder has lost its `BetterContinents` file. Put it back from a backup.

A world's settings are fixed when it is made, so for the first three, make the world again.

My world is all ocean, or much lower than I painted it.

Check the waterline first. At the default Sea Level Adjustment of 0.5, the waterline is heightmap value **0.30** (grey 77 of 255) at Heightmap Amount 1, and **0.15** (grey 38) at Amount 2. A heightmap drawn for Amount 2 and read at Amount 1 puts every lowland biome under water. See Heights and the heightmap.

Then check what the world was made with. The settings are stored in the world when it is made, so changing `Heightmap Amount` or `Sea Level Adjustment` afterwards changes only worlds you make next. In Debug Mode you can change the loaded world with `bc h am` and `bc g sl`; see Changing an existing world.

A heightmap from a world export is encoded for the Heightmap Amount in its `export.cfg`. A world made From Config with `Directory` set to the export folder reads that `export.cfg` too, so its heights come out right. A `heightmap.png` copied without its `export.cfg` needs the right Amount and Sea Level Adjustment in your own config. An exported heightmap also carries a record of its Amount and Sea Level inside the PNG: when a new world reads it at other values, the log warns and names both, and `bc info` shows the record.

I edited my map pictures, but the world did not change.

A world stores its maps when it is made, so editing a picture never changes a world that exists. Make a new world, or, in Debug Mode, read the new picture into the loaded world with `bc reload <map>` (for example `bc reload bm`). Better Continents then regenerates the zones, except the ones around what players built or worked on, a tombstone, another player, or you inside a dungeon.

Where each location stands was decided when the world was made, and a regenerated zone builds its location again on the same spot. For a new location layout, make a new world. See Changing an existing world.

A map was refused as too big.

A map can be at most 16,384 × 16,384 pixels. A bigger one is refused before it is read, and the world is made without it. The log says:

```
Cannot use texture <file>: it is W x H pixels, and the largest map Better Continents reads is
  16384 x 16384. Scale it down (at 2 m a pixel, 16384 pixels hold a world 32.7 km across).
```

If the message names a smaller limit and `Max Map Size`, that setting in `[07 BetterContinents.Misc]` is set to 4096 or 8192 on this machine: raise it, or scale the map down. A map must also be square; one that isn't is refused with a line that gives its width and height. See Image maps.

Fine heights are not used.

A `heightmap-fine.png` beside the heightmap refines its heights (see Heights and the heightmap). It is read only when a new world is made, by `bc_import`, and by `bc h fn`, `bc reload hm`, `bc reload all` and `bc h alpha` in Debug Mode. When a world reads the heightmap and does not use the fine picture, the log says why, in a line that starts `Fine heights:`. The reasons, in the order they are checked:

1. There is no `heightmap-fine.png` beside the heightmap (nothing is logged).
2. The world is of settings version 11, which has no fine heights.
3. `Heightmap Alpha` as it was before 0.10 (Legacy) reads the heightmap as 8-bit grey.
4. `Compact Maps` in `[07 BetterContinents.Misc]` is Off.
5. `heightmap.png` cannot be read.
6. `heightmap.png` is not 16-bit grey (grey, or grey and alpha).
7. `heightmap-fine.png` cannot be read.

8. It is bigger than `Max Map Size` allows.
9. It is not 8-bit grey (no alpha, no palette).
10. It is a different size from `heightmap.png`.
11. Its record's Fine Format is not 1.
12. Its record names another CRC-32: it was made for another `heightmap.png`. After you edit `heightmap.png`, make `heightmap-fine.png` again (export again) or delete it.
13. Its image data cannot be read.
14. Every byte is 0, so it changes nothing.

In each case the world is made from `heightmap.png` alone.

Locations are missing.

Run `bc locs`: it counts every location in the world and warns when the start temple, a boss, Haldor or the Queen's entrance is missing. If the biome map leaves a biome out, the game cannot place the locations that need it, and the log says so, once per biome:

```
The biome map has no Mistlands, so no ... exist for it. Locations that require Mistlands will
  be skipped.
```

`Skip Default Locations` in `[01 BetterContinents.Global]` also leaves out every location that the location map does not place. See Locations, creatures and the start.

An alt biome is not where I painted it, or not anywhere.

Run `bc ab info`: it says what each legend line planted, and why each alt biome did or did not place. `bc ab here` shows the region you stand in. The usual reasons:

- The world still has the old map. Colours are read when the map is stored, so after editing the picture or the legend, read it again with `bc reload ab`.
- A colour claims only the base biomes its alt biomes belong to: Dark Meadows painted over a beach leaves the beach alone.
- The pixels' colour is more than 20 RGB units from every legend colour, so they are unplanted. The log lists the most common such colours.
- For random alt biomes: the game gives them only to regions of the right size, far enough from the centre, and whose border's height, (lowest + highest) / 2, is 30 m or more. Big hand-drawn regions usually need `Region Size Scale` of 2 to 10.

See Alt biomes.

I changed Biome precision, but my world looks the same.

`Biome precision` in `[03 BetterContinents.Biomemap]` is stored in a world when it is made, so changing the config changes only worlds you make next. To change it in a world that exists, use `bc b p <value>` in Debug Mode; the world saves the new value. A world made by an older Better Continents (settings version 11) uses at most 5, whatever is stored. See Biomes and the biome map.

A square of water looks deeper or shallower than the water around it.

Each 64 m zone's water works out how deep the sea is under it when the zone loads, and keeps that depth for its look and its waves. In a Better Continents world, a zone's water takes the depth of its ground again whenever the ground is rebuilt in place, as after a `bc` change. A world without Better Continents is left as the game has it: there, going about 300 m away and back, or logging out and in, loads the zone again and shows it right.

Trees and rocks stand in pairs, a copy within a metre of each one.

A zone that the game fills with vegetation a second time over its own objects doubles every tree and rock in it. That happens with a save put back together from mismatched files, or a world tool's zone command. Better Continents keeps new twins out of every world it makes, and `bc_twins` finds and removes the ones already there:

1. Back up the world folder.
2. `bc_twins` counts them. On a dedicated server, ask the server: `bc_twins server`.
3. `bc_twins remove` shows what a cleanup would delete, and deletes nothing.
4. `bc_twins remove confirm` deletes the extra copies. The copy nearest the ground stays, and nothing a player built is removed.

See Forests and vegetation.

Multiplayer and big worlds

A player cannot join: the versions differ.

On a world that uses Better Continents, the server and every player must run exactly the same Better Continents version, and a player with another version is refused. Update the server and every player together. See Multiplayer and dedicated servers.

Joining takes a long time.

A player who doesn't have the world yet is sent its Better Continents settings and maps before the join finishes, at `Settings Transfer Rate` in `[07 BetterContinents.Misc]` (default 512 KB/s). At that rate a 52.6 MB world takes about 1.7 minutes, and at MB3 about 17 s; these times follow from the transfer rate. Their game keeps a copy, so the next join is quick.

- Raise `Settings Transfer Rate` on the server, if its upload can carry it.
- Better, give players the world's `.bcworld` file: `bc_cache export` on the machine that runs the world writes it, and a player who imports it joins with no download.

See Multiplayer and dedicated servers.

A world bigger than 16 km loses creatures, objects or portals.

A world whose World Size + Edge Size is past 16,350 m meets up to three problems in the game itself unless it has Wide Sectors: past about 16.4 km every object shares one list; past about 22.1 km, a patch of zones shares a save file with every portal in the world; past 32,767 m, the game saves the hidden spawners of the farthest zones on the 20 km line, and those zones stop spawning their own creatures after a restart. With `Wide Sectors` in `[07 BetterContinents.Misc]` at its default, Auto, a new world that big is made wide from the start. An existing big world is converted only with On, which is one way: back it up first. Read World size and big worlds before you change anything.

Versions

I want to open a world in an older Better Continents.

Back up the world folder (`worlds_local/<World>`) first. Then check what the world uses:

The world	In an older version
a wide world (Wide Sectors)	An older Better Continents stops at it (*Unknown feature: 68*) and gives it vanilla terrain, and its saves write objects twice: two of every chest. Don't.
made with fine heights	0.10.0 to 0.10.2 cannot read it (*a map saved in format 2, which this version of Better Continents cannot read*), and it gets vanilla terrain. Don't.
with `High Terrain` On or Off	0.10.2 and older stop at it (*Unknown feature*) and give it vanilla terrain. Don't.
read at a Heightmap Amount above 5, `High Terrain` on Auto	Older versions open it with its terrain, but without the high-terrain rules.
compact (Compact Maps)	Better Continents 0.9 cannot read it, and it gets vanilla terrain. 0.10.0 and later read it.
settings version 12	0.9 opens it as a version 11 world, and its next save writes it as one.

See Coming from 0.9.x or older.

Exports

I can't find my exports.

They are in `<save data>/BetterContinents/<world name>/export-<date and time>/`, beside the game's `worlds_local` folder, never inside the world's own folder. `bc_import list` numbers every export, newest first.

The export window doesn't offer 16,384 px.

`Largest Size` in `[09 BetterContinents.Export]` is set lower on this machine. The window hides bigger sizes, and `bc_export` refuses them with a message that names the setting. Set it to 16384.

A 16,384 px export made no New World preset.

That is by design: above 8,192 px the export leaves the preset to `bc_import`, and its `README.txt` says so. Run `bc_import` in the main menu to make the preset: at 16,384 px that took about 1.4 GB more memory and 19 s, measured on one machine. See Export and import.

Other mods

Expand World Size, Expand World Data, ZenMap, Server Devcommands and Upgrade World each have notes in Other mods.

Appendix A

Formulas

This appendix gathers the mathematical formulas used to turn images into a Valheim world.

Height from a heightmap

The height of terrain at each point, when a heightmap is loaded with Heightmap Override All on (the default):

```
v                 = pixelValue / 65535              (for a 16-bit pixel)
finalHeight       = v × Amount − 0.15 + SLA
SLA               = Lerp(1, −1, Sea Level Adjustment) = 1 − 2 × Sea Level Adjustment
worldHeight (m)   = finalHeight × 200
altitudeAboveWater = worldHeight − 30               (the sea is at 30 m)
```

This is with `Heightmap Blend` 1, `Heightmap Add` 0 and `Heightmap Mask` 0, the defaults. With the default Sea Level Adjustment of 0.5, SLA is 0, and this simplifies to:

```
altitude (m) = (pixelValue / 65535 × Amount − 0.30) × 200
```

Example: Heightmap pixel at 8-bit value 128 (16-bit 32,768, or 0.5), Amount 2, default Sea Level Adjustment:

```
v = 32768 / 65535 = 0.50
finalHeight = 0.50 × 2 − 0.15 + 0 = 0.85
height = 0.85 × 200 = 170 m
altitude = 170 − 30 = 140 m above water
```

Waterline

The pixel value at the waterline (ground at sea level, 30 m) when a heightmap is loaded:

```
waterlinePixel = 0.30 / Amount
```

Example: Amount 2:

```
waterlinePixel = 0.30 / 2 = 0.15
8-bit grey = 0.15 × 255 ≈ 38
```

Everything darker than the waterline is under water. Black is the deepest seabed, 60 m below the water at every Amount.

Map span and pixel size

Every map covers a square from −(World Size + Edge Size) to +(World Size + Edge Size) on each axis. With the default World Size 10,000 m and Edge Size 500 m, the square is 21,000 m across.

The size of each pixel:

```
metresPerPixel = 2 × (World Size + Edge Size) / pixelsAcross
```

Example: 4,096 × 4,096 px map, default sizes:

```
metresPerPixel = 2 × 10,500 / 4,096 = 5.13 m
```

At 5.13 m per pixel, one 64 m zone (the game's own grid) is 12.5 pixels across.

The 16-bit step

When your heightmap is saved as 16-bit greyscale, each step between adjacent pixel values represents:

```
stepSize (m) = 200 × Amount / 65,535
```

Examples:

- Amount 1: `200 × 1 / 65,535 = 0.003 m` (3 mm)
- Amount 10: `200 × 10 / 65,535 = 0.031 m` (31 mm)
- Amount 81: `200 × 81 / 65,535 = 0.247 m` (24.7 cm)

At Amount 81, steps of 25 cm appear as stairs on gentle slopes. Use fine heights (below) to refine them.

Fine heights

When using fine heights, each heightmap pixel is refined with a fine picture byte. The calculation is:

```
c                     = 16-bit heightmap value
f                     = fine picture byte (−128 to 127)
refinedHeightmapValue  = (256 × c + f) / 16,776,960
```

16,776,960 is 256 × 65,535, so with f = 0 the result is exactly the heightmap's own value, c / 65,535. Each unit of f is 1/256 of a 16-bit step: 24 bits of precision in all.

The byte is read as a signed number: 0 to 127 as they are, then 255 is −1, down to 128, which is −128.

Example: a fine byte of 16 adds 16/256 = 1/16 of a 16-bit step. At Amount 81, where a step is 0.247 m, that is about 1.5 cm. A 4-bit fine picture uses only multiples of 16 (from −128 to 112), so its steps are 1/16 of a 16-bit step.

The world export writes a 4-bit fine picture by default (`fine=auto`) where a 16-bit step is more than 1.5 cm: 200 × Amount / 65,535 > 0.015, so from an Amount of about 4.92.

Sea Level Adjustment

`Sea Level Adjustment` moves the waterline. The offset is:

```
SLA = Lerp(1, −1, Sea Level Adjustment)
```

This offset is added to finalHeight before multiplying by 200. Each 0.1 below the default of 0.5 raises the land by 40 m:

```
altitude change (m) = (Sea Level Adjustment − 0.5) × −400
```

Example: Sea Level Adjustment 0.35:

```
SLA = Lerp(1, −1, 0.35) = 1 − 2 × 0.35 = 0.3
altitude change = (0.35 − 0.5) × −400 = 60 m higher
```

The whole heightmap is raised 60 m, so white stands 200 m above the water instead of 140 m, at Amount 1.

The waterline is at this pixel value:

```
waterlinePixel = (2 × Sea Level Adjustment − 0.7) / Amount
```

At Amount 1 that is 0.30 at the default 0.5, and 0 at 0.35: the sea is gone. At 0.85 or more, everything is under water. Each 0.05 of Sea Level Adjustment is 20 m of water.

Example: choosing Heightmap Amount

You have drawn a heightmap for a world with a peak at 200 m above water and another at 150 m. Which Amount should you use?

At Amount 1, the ceiling is 140 m above water: neither peak fits. You would have to redraw them shorter.

At Amount 2, the ceiling is 340 m: both peaks fit. You should set `Heightmap Amount` to 2 in `BetterContinents.cfg` before you create the world.

A peak at 200 m above water is at pixel value:

```
altitude = 200 m
finalHeight = 200 / 200 + 0.15 = 1.15
value = (1.15 + 0.15) / Amount = 1.30 / 2 = 0.65
8-bit grey ≈ 166
```

Draw it at grey 166 (or thereabouts) for Amount 2. At Amount 1, the same pixel would become `value × 1 = 0.65`, which is `(0.65 − 0.30) × 200 = 70 m` high: far too low. This is why the Amount must be chosen early and kept consistent.

Appendix B

Credits and history

The mod

- **billw2012** created Better Continents in 2021, the pioneer of Valheim heightmap and custom continent generation.
- **Jere Kuusela** (JereKuusela) maintained and expanded it from 2023, up to version 0.7.31.
- **Wubarrk** took it over in 2026, brought it to Valheim 1.0 with version 0.8.0, and maintains it now.

Other contributors helped along the way. Better Continents is free software under the GNU Lesser General Public License, version 2.1. Its source code is at github.com/RGlabs84/BetterContinents10, where each release is tagged, and its changelog lists every change.

The guide

- The original *Better Continents Guide* was written by **billw2012** and is maintained by **Jere Kuusela**, at jerekuusela.github.io/BetterContinents-Docs. It describes Better Continents 0.7.x.
- From September 2026, an edition for Valheim 1.0 updated that guide chapter by chapter, for Better Continents 0.8.0 to 0.10.2, with boxes that marked each change.
- This edition was rewritten from the ground up for Better Continents 0.10.x by **Wubarrk**, with **Claude** (Anthropic) as the drafting agent.

The guide is licensed under the GNU General Public License, version 3: see License.

Versions for Valheim 1.0

Version	What it brought
0.8.0	Better Continents on Valheim 1.0. A world's settings live in the world's own folder, in the `BetterContinents` file, and a world from before Valheim 1.0 is moved over on its first save.
0.8.1	Alt biomes planted with colours (the alt-biome map), per-world alt-biome settings, and the `bc ab` and `bc_altbiomes` commands.
0.9.0	World export and import, the export HUD, settings that apply while the game runs, `Settings Transfer Rate`, the `.bcworld` world cache, and Biome precision working again.
0.9.1	Rebuilt for a Valheim 1.0 update.
0.9.2	Works with ZenMap and no-map worlds: the minimap is finished before other mods see it.
0.9.3	Expand World Data's biomes on biome maps.
0.9.4	A `Directory` that holds a world export brings the export's own settings.
0.10.0	Settings version 12 for new worlds: World Size lays out the whole world, Heightmap Alpha blends, Biome precision goes to 31, location pins land exactly, and Expand World Data's rules reach the maps. For every world: zone regeneration after `bc` changes, the vegetation twin guard and `bc_twins`, maps held as tiles, and Compact Maps.
0.10.1	Rebuilt for a Valheim 1.0 update. The New World panel shows the mod's emblem.
0.10.2	`Forest Scale` defaults to 0.5, the game's own size of forest and clearing patches.
0.10.3	Maps up to 16,384 px and Compact Maps as Auto, On or Off, worlds past 16 km with Wide Sectors, terrain up to 16 km high with High Terrain, fine heights, exports at 16,384 px that write fine heights, and big worlds that load in seconds.

Versions before 0.8.0 ran on Valheim before 1.0. This guide covers Better Continents 0.10.x.

Appendix C

GNU General Public License, version 3

The GNU General Public License, version 3, reproduced verbatim from `original/LICENSE` (identical to `updated/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>.
```