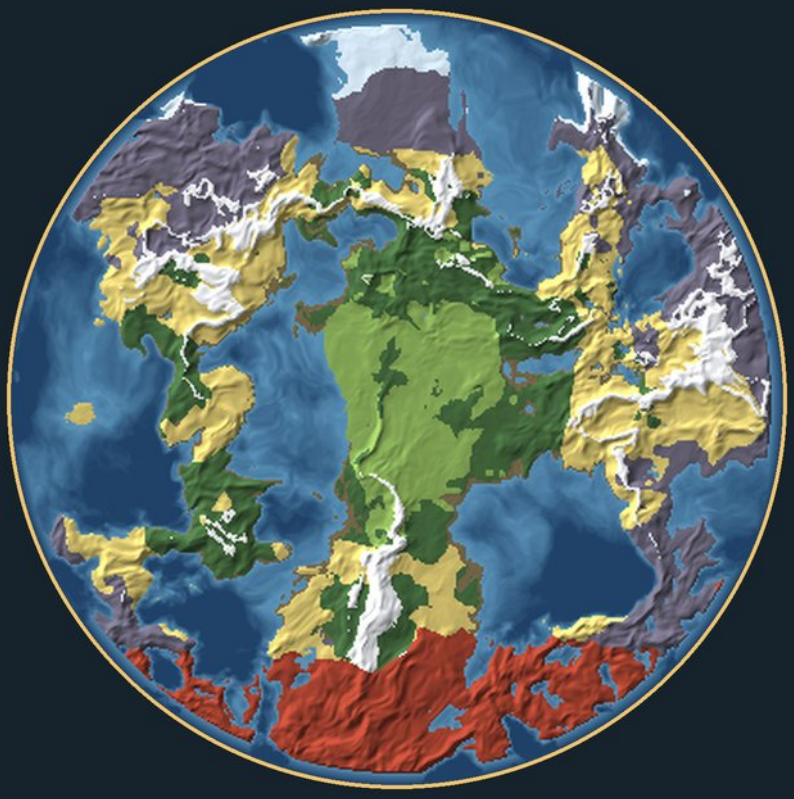


Export & Import

the easy guide to Better Continents 0.10.x

Turn your Valheim world into pictures, change them, and make a new world from them.

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

For Valheim 1.0 and Better Continents 0.10.x (github.com/RGlabs84/BetterContinents10)

24 September 2026, updated 4, 5, 7 and 26 October 2026

Written for players who have never edited a config file or a heightmap.

Every picture in this guide is a drawing, not a screenshot: the game looks different.

Licensed under the GNU General Public License, version 3, WITHOUT ANY WARRANTY. The licence text is the final appendix; the Markdown source, the pictures and the build scripts are attached to this PDF (export-guide-source.zip). The cover map is made up.

Contents

Start here: what export and import do . . . 6
What comes along, and what stays behind . . . 6
What is new in 0.10 . . . 6
The whole thing in four steps . . . 7
Words used in this guide . . . 8
What you need . . . 8
How to read this guide . . . 8

License . . . 9

1 Turn on the export HUD . . . 10
Way 1: in the game, with Configuration Manager . . . 10
Way 2: edit the config file . . . 10
The status box (F9) . . . 11
The window (F7) . . . 11
Change the keys . . . 12
All the settings of the Export section . . . 12

2 Export your world . . . 13
Before you start . . . 14
Export with the window . . . 14
Every map in one line . . . 15
Which size? . . . 16
Export with the console instead . . . 16
What the progress looks like . . . 17
Where the folder is . . . 18
What each file is . . . 19
Cancel an export . . . 20

3 Make a new world from an export . . . 21
Why it has to be a new world . . . 21
Four ways . . . 21
Way A: pick the preset . . . 21
Way B: bc_import, after editing the pictures . . . 23
Way C: the config, for editing and testing . . . 24
Way D: set only Directory . . . 24
The Import tab: ways B and C with buttons . . . 26

Check the new world . . . 26
Using the export with BetterWorlds . . . 27

4 Edit the pictures . . . 28
The golden rules . . . 28
Your image editor . . . 28
Find a place in the pictures . . . 29
The heightmap: heightmap.png . . . 29
How high can it go? . . . 31
Fine heights: heightmap-fine.png . . . 31
The biome map: biomemap.png . . . 32
The location map: locationmap.png . . . 32
The forest map: forestmap.png . . . 33
The heat map: heatmap.png . . . 33
The cold map: coldmap.png . . . 34
The location exclusion mask: no_spawns_mask.png . . . 34
Alt biomes: altbiomemap.png . . . 34
Lava, moss and paint . . . 35
Letting the game make a part itself . . . 35
Make the preset again . . . 35

5 Cut and paste between worlds . . . 36
A worked example: move a mountain range from world A to world B . . . 37
The seam . . . 37
What moves with the paste, and what does not . . . 38

6 Biome precision . . . 39
What it does, in plain words . . . 39
When to raise it . . . 40
How to set it . . . 40

7 Multiplayer and servers . . . 41
Who can export . . . 41
Why a player's export has fewer locations . . . 41
For server owners: Allow Export . . . 41
For server owners: the HUD . . . 42
For admins: export a dedicated server's world . . . 42
Share a world made from an export . . . 42

8 When it goes wrong . . . 43

The new world is wrong . . . 43
The editor pack and bc_import edited . . . 44
The export does not start or does not finish . . . 45
The HUD or the keys do nothing . . . 46
The preset or the import . . . 46
World creation seems to hang . . . 47
Joining a server takes long . . . 47
Where to look: the log . . . 47
Getting help . . . 48

9 Give your players the world in advance . . . 49
What happens when a player joins . . . 49
Faster: the server's Settings Transfer Rate . . . 49
No wait at all: hand out the world in a file . . . 50
If you run the world . . . 50
If you are a player . . . 51
How the file is checked . . . 51
A player can make the file too . . . 51

10 Edit the ground in Gaea and other heavy editors . . . 52
When to use which . . . 52
What you need . . . 52
Step 1: export with the editor pack . . . 53
Way 1: the window . . . 53
Way 2: the console . . . 53
What is in the editor folder . . . 54
The masks . . . 54
The numbers: how the ground is stored . . . 55
Step 2: load the ground in Gaea . . . 55
Step 3: edit . . . 56
Step 4: save the result in editor/return/ . . . 56
Step 5: bring it back . . . 56
How the numbers come across . . . 57
What the new folder holds . . . 58
What does not follow the ground . . . 58
Limits, said plainly . . . 58
What we have not yet checked with a real Gaea build . . . 59
Where to go next . . . 59

Appendix A The console commands . . . 60

bc_export . . . 60
bc_import . . . 62
bc_import edited . . . 62
bc_cache . . . 63
bc_twins . . . 64
The bc commands (host, cheats) . . . 64

Appendix B export.cfg, key by key . . . 66
[00 BetterContinents.Debug] . . . 66
[01 BetterContinents.Global] . . . 67
[02 BetterContinents.Heightmap] . . . 68
[03 BetterContinents.Biomemap] . . . 68
[04 BetterContinents.Forest] . . . 68
[05 BetterContinents.StartPosition] . . . 69
[06 BetterContinents.Maps] . . . 69
[07 BetterContinents.Misc] . . . 69
[08 BetterContinents.AltBiomes] . . . 69
What the import never takes from export.cfg . . . 69

Appendix C manifest.json, field by field . . . 70
The export . . . 70
Size and place . . . 71
Heights . . . 71
The other maps . . . 72
The preset, the files and the notes . . . 72
A manifest that bc_import edited writes . . . 72
editor/editor.json . . . 73

Appendix D The formulas, for the curious . . . 74
Pixel and world . . . 74
Heights . . . 74
Fine heights . . . 75
Heat . . . 75
Cold . . . 75
Forest . . . 76
Lava, moss and paint . . . 76
The editor pack . . . 76

Appendix E GNU General Public License, version 3 . . . 78

Start here: what export and import do

Better Continents 0.10.x can turn a Valheim world into pictures, and pictures back into a world.

Export copies the world you are standing in into a folder of pictures (PNG files). One picture shows how high the ground is, one shows which biome is where, one has a dot for every boss, trader and dungeon, and a few more hold the forests, the Ashlands heat, the lava and the rest. Nothing in your world changes.

Edit is up to you. Open the pictures in an image editor and change what you like: raise a mountain, paint a new island, move a boss, or cut a piece out of one world and paste it into another.

Import turns the folder back into a *preset*: a ready-made recipe for a world that you pick on Valheim's New World screen. You press Done, and Valheim builds a new world from your pictures.



Illustration, not a screenshot: from your world to a folder of pictures, and from the pictures to a new world.

What comes along, and what stays behind

Comes along to the new world	Stays in the old world
the shape of the land and the sea floor	buildings, and anything else players placed
which biome is where	ground players dug, raised or flattened
forests, the Ashlands heat and lava, the Mistlands ground cover	items, portals, tamed animals
alt biomes (Dark Meadows, Wolf Mountain and the rest)	the explored map, and which bosses were beaten there
every location: start temple, bosses, traders, dungeons	

The old world is never changed. The new world is a separate world next to it.

What is new in 0.10

If you used the export with Better Continents 0.9.x, these are the differences. Your old exports and presets still work, and a preset still makes the world it always made.

Each world has a *settings version*. The differences below belong to worlds of the newest version, 12, which is what most worlds made from an export get, 0.9.x exports of a world of the normal size included. A world gets the older version, 11, and plays as it did in 0.9.x, when it is made from one of the presets that come with the mod, from a preset that 0.9.x saved, from an export of a world that was resized under 0.9.x or made under Expand World Size at another size, or while the testing setting `Override version` is 11 (every world that 0.9.x made is version 11 too). Chapter 3 shows how to read a world's version.

- **Heights are stored for Heightmap Amount 1** (it was 2): the picture holds ground from -30 m to 170 m on the game's scale, where the sea is at 30 m (60 m under the sea to 140 m above it), and ground higher than that is clipped. Amount 2 is one setting away for worlds with very high mountains (chapters 2 and 4).
- `heightmap.png` **records its Heightmap Amount and Sea Level inside the PNG**, and a new world that reads it at other values warns in the log (chapters 4 and 8).
- **The pictures cover the world's size.** In a version 12 world they span its World Size + Edge Size on each side of the middle: 21 km in a normal world (chapters 2 and 4).
- **Biome precision goes up to 31** in a version 12 world (chapter 6).
- **The commands:** `bc_export server` also answers the admin who asked, and `bc_twins` finds doubled vegetation (appendix A).
- **Exports go up to 16384 px**, and take about 100 MB of memory at any size. Above 8192 px the export leaves the New World preset to `bc_import` (chapters 2 and 3). `Largest Size` in the config can stop the window at a smaller size (chapter 1).
- **Heights go up to 16 km**: the Heightmap Amount goes up to 81, and from an Amount of about 5 the export also writes `heightmap-fine.png`, finer heights for tall worlds (chapter 4).
- **The export holds more of the world.** It writes `coldmap.png`, where the Deep North is, so a world made from it keeps the Deep North's weather, snow and calm sea; the world's location exclusion mask, `no_spawns_mask.png`; and `world.json`, the world's size and layout. `export.cfg` pins the mountain and Mistlands noise to 0, because the heights hold it already, and `heatmap.png` is exact for the game's own Ashlands test (chapters 2 and 4, appendices B and C). The other mod BetterWorlds reads such a folder as it is (chapter 3). Every export has these, whatever the world's settings version.
- **An editor pack for Gaea and other terrain editors.** `bc_export ... editor` adds the ground as a 32-bit float picture and masks, and `bc_import edited` brings the edited ground back (chapter 10). It is checked with Better Continents' own code, not yet with a real Gaea build, and chapter 10 says so.

The whole thing in four steps

1. **Turn on the export HUD** (the export's on-screen window). Chapter 1.
2. **In your world, press F7 and then Start export.** Wait a few minutes. Chapter 2.
3. **Go to the main menu and make a New World.** In the Better Continents box, pick the preset named after your world and the time of the export. Chapter 3.
4. **Press Done.** Valheim builds the new world.

Chapters 4 and 5 show how to edit the pictures and how to cut and paste between worlds. Chapter 8 lists what can go wrong and how to fix it. Chapter 9 is for whoever runs a server: how to get the world to the players faster, or before they even join. Chapter 10 is for sculpting the ground in a terrain editor such as Gaea instead of an image editor.

Words used in this guide

Word	What it means
export folder	the folder one export writes, `BetterContinents/<world>/export-<date>-<time>/`
preset	a saved world recipe: a file in `BetterContinents/presets/` that the New World screen lists
config	`BetterContinents.cfg`, the text file with Better Continents' settings, in `BepInEx/config/`
console	Valheim's command line: press F5 to open and close it
HUD	the status box and the window Better Continents draws over the game
heightmap	a grey picture of the ground: the brighter a pixel, the higher the ground
editor pack	the `editor/` folder an export can add: the ground as 32-bit floats and masks, for a terrain editor such as Gaea (chapter 10)
legend	a `.txt` file next to a colour picture that says what each colour means
pixel	one dot of a picture; in a normal export one pixel is about 5 m of the world

What you need

- **Valheim 1.0.**
- **BepInEx**, the mod loader. A mod manager such as Gale or r2modman installs it for you.
- **Better Continents 0.10.x.** The status box of chapter 1 shows the exact version in its title.
- **For editing: an image editor that can save a 16-bit greyscale PNG**, for the heightmap. A normal picture has 256 shades of grey, which stores the ground's height in steps of about 0.8 m, so slopes become stairs. A 16-bit picture has 65,536 shades: steps of about 3 mm.

Editor	Price	Heightmap (16-bit grey)	Colour maps
GIMP	free	yes	yes
Krita	free	yes	yes
Photoshop	paid	yes	yes
Paint.NET	free	**no**: it works in 8 bits	yes

For sculpting ground in a terrain editor (chapter 10) you need one that reads a 32-bit float TIFF, such as Gaea. It is a separate program by someone else, and nothing else in this guide needs it.

You do not need cheats (devcommands) to export or import. Configuration Manager, a separate mod, makes settings easy to change in the game, but a text editor works as well.

How to read this guide

- Every chapter starts with an **In short** box. If it is all you need, skip the rest of the chapter.
- Steps are numbered, one action each. After a step, a line starting with **You see:** says what should happen.
- Green boxes are tips. Red boxes are mistakes worth avoiding. Blue boxes are good to know.
- Every picture is a drawing made for this guide, not a screenshot. The game looks different; the drawings show where things are and what they say.

License

Better Continents: Export & Import, the easy guide was written by Wubarrk, with Claude (Anthropic's model) as the drafting agent, and is released under the **GNU General Public License, version 3** (GPL-3.0), the licence of the Better Continents Guide it accompanies, so that the two documents can be shipped and changed together under one licence. Copyright 2026 Wubarrk.

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

This guide describes Better Continents 0.10.x as its source code is written: the command names, settings, defaults, file names and messages were copied from the code, and the examples of the export's text files follow what the code writes (the numbers in them are examples). The pictures are drawings, not screenshots. Where the game shows something different from this guide, the game is right; please report it (see chapter 8).

Better Continents is free software under the GNU Lesser General Public License, version 2.1 (created by billw2012, maintained by Jere Kuusela from 2023 to 2026 and by Wubarrk since September 2026). The Better Continents Guide is GPL-3.0 (written by billw2012). Neither is reproduced here beyond short statements of how the mod behaves. Valheim is the property of Iron Gate Studio; the menu labels quoted here are the game's own English texts, quoted so that you can find the buttons.

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

- `docs/`: the Markdown source of this guide and its pictures;
- `tools/make_figures.py`: the script that draws every picture (the maps are made up from noise);
- `build_export_guide.py` and `build_pdf.py`: the build scripts (the second is the Better Continents Guide's renderer, which the first reuses);
- `LICENSE`: the GPL-3.0 text.

1 Turn on the export HUD

> **In short:** set `Hud` to `true` in the `[09 BetterContinents.Export]` section of `BetterContinents.cfg`, or tick **Hud** in Configuration Manager. In a world, **F9** then shows a small status box and **F7** opens the export window. Prefer typing commands? You can skip this chapter: chapter 2 shows the console way too.

The HUD (heads-up display: the box and the window Better Continents draws over the game) is off when you first install the mod. It only ever shows while you are in a world, never in the main menu.

Way 1: in the game, with Configuration Manager

Configuration Manager is a separate, free mod that lists the settings of all your mods inside the game. Many mod packs already include it; search your mod manager for "Configuration Manager" if yours does not.

1. **Start Valheim and load any world.**

   You see: your world, as usual.

2. **Press F1** (Configuration Manager's own key).

   You see: a window listing your mods and their settings.

3. **Find Better Continents, and in it the section `09 BetterContinents.Export`.**

4. **Tick Hud.**

   You see: at once, a dark box with a gold title, *Better Continents 0.10.x export*, near the top-left corner of the screen.

5. **Press F1 again** to close Configuration Manager.

Way 2: edit the config file

The config file is a plain text file named `BetterContinents.cfg`. BepInEx writes it the first time the game starts with Better Continents installed.

1. **Find the file.** It is in the `BepInEx/config/` folder:
   - with a mod manager (Gale, r2modman, Thunderstore Mod Manager): inside your profile's folder. Every manager can open that folder for you, and most have a config editor that opens the file directly;
   - without a mod manager: inside the Valheim game folder (in Steam: right-click Valheim, then Manage, then Browse local files).
2. **Open it with a plain text editor**: Notepad on Windows, gedit or Kate on Linux. Not Word.
3. **Find the section `[09 BetterContinents.Export]`**, near the end of the file.
4. **Change `Hud = false` to `Hud = true`, and save the file.**

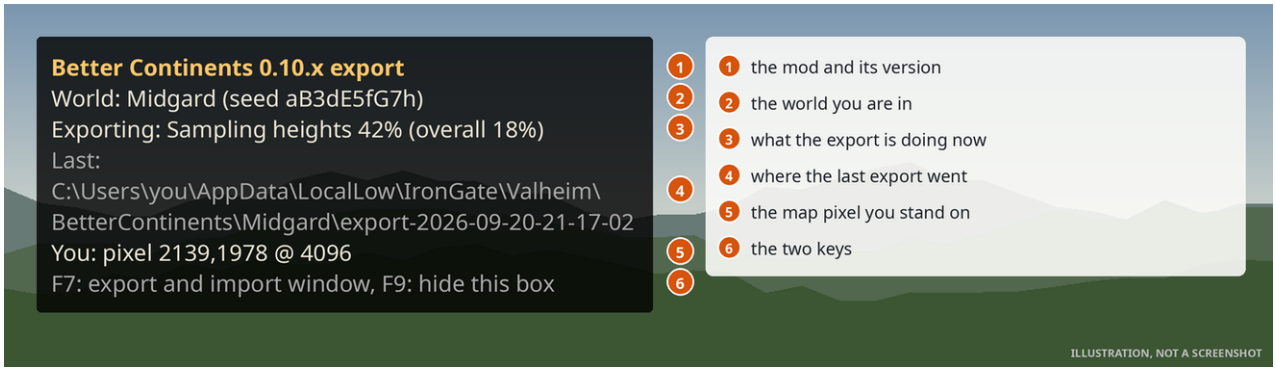
   You see: if the game is running and you are in a world, the status box appears within a second or so. No restart is needed: Better Continents reads the file again whenever it changes.

The section looks like this (the real file has comment lines starting with # above every setting):

```
[09 BetterContinents.Export]
Hud = true
Allow Export = true
Hud Hotkey = F9
Window Hotkey = F7
Default Size = 4096
Default Heightmap Amount = 1
Default Sea Level = 0.5
Largest Size = 16384
```

Watch out: on a multiplayer server, the server's own Hud setting is used instead of yours (chapter 7). If the box never shows up on a server, ask the server's owner to turn Hud on in the server's config.

The status box (F9)



Illustration, not a screenshot: the status box while an export runs, with its lines numbered.

	Line	What it tells you
1	Better Continents 0.10.x export	the mod and its exact version; the HUD needs 0.9.0 or later
2	World: Midgard (seed aB3dE5fG7h)	the world you are in, and its seed
3	Status: Idle, or Exporting: …	what the export does now: the step, how far the step is, and how far the whole export is
4	Last: …	the folder of the last export you made since the game started
5	You: pixel 2139,1978 @ 4096	where you stand, as a pixel of a 4096 px export: 2139 pixels from the left (west) edge and 1978 from the top (north) edge. Chapter 4 uses it to find places in the pictures
6	F7: …, F9: …	the two keys

The box can show three more lines: *Importing: …* while a preset is being made, *Error: …* after an export failed, and *Cannot export: …* with the reason when you may not export right now.

Press F9 to hide the box, and again to show it.

The window (F7)

Press F7 to open the export and import window, and again to close it. It has two tabs, **Export this world** (chapter 2) and **Import an export** (chapter 3).

While the window is open, the mouse pointer is free and the game ignores your clicks and number keys, so you can click and type in the window. You can still walk. Drag the window by its title to move it.

> **Good to know:** the two keys only react to a plain key press, with no Shift, Ctrl or Alt held. They are ignored while you type: in the chat, the console, a sign, a map pin, the build menu's search box, or one of the window's own number fields. Click an empty part of the window to let go of a number field.

Change the keys

In the same section, `Hud Hotkey` (F9) and `Window Hotkey` (F7) set the two keys. Any key name works (F1 to F12, a letter, and so on), and `None` turns a key off. The change applies at once.

All the settings of the Export section

Setting	Default	What it does
`Hud`	false	shows the export HUD in a world
`Allow Export`	true	on a server: whether connected players may export the world (chapter 7)
`Hud Hotkey`	F9	shows or hides the status box
`Window Hotkey`	F7	opens or closes the window
`Default Size`	4096	the size of an export unless you pick another (128 to 16384)
`Default Heightmap Amount`	1	how heights are stored, from 0.01 to 81; leave it at 1 unless your world has ground more than 140 m above the sea (chapter 4 explains)
`Default Sea Level`	0.5	how heights are stored; leave it at 0.5
`Largest Size`	16384	the largest export this computer makes: 4096, 8192 or 16384. The window offers sizes up to it, and `bc_export` refuses bigger ones

All eight apply at once, while the game runs. Every other Better Continents setting only matters when a new world is created (chapter 3), with these exceptions: `Settings Transfer Rate` (chapter 9), `Map Memory` and `Max Map Size` belong to your computer and apply while the game runs too; `Wide Sectors`, for worlds more than 16,350 m from the middle to the edge, is read when a world loads as well (leave it at Auto); and `Override version` is read at every save (leave it empty).

> **Good to know:** `Default Heightmap Amount` was 2 in Better Continents 0.9.x. A `BetterContinents.cfg` that still holds that old default is set to 1 once, and the log says so; a 2 you choose afterwards stays.

2 Export your world

> **In short:** stand in the world, press **F7**, leave every map ticked, keep the size at 4096, press **Start export** and wait until it says Done, usually a few minutes. Or type `bc_export` in the console (F5). The pictures land in `BetterContinents/<world>/export-<date>-<time>/` in Valheim's save folder, and a New World preset is made for you (up to 8192 px).

This is the export window, on its **Export this world** tab. The steps below go through it.



Illustration, not a screenshot: the Export this world tab, its parts numbered.

	Part	What it is
1	the two tabs	**Export this world**, and **Import an export** (chapter 3)
2	the maps	every picture the export can write; untick one to leave it out
3	the hint	what the map under the mouse pointer holds and costs
4	Size	how many pixels each picture has on a side: 1024 to 16384
5	Heightmap Amount, Sea Level, Heat Scale	how heights and heat are stored in the pictures: keep 1, 0.5 and 10
6	the options	Fine heights (keep it on auto), and how heights, forest and the world's edge are stored: keep them as they are
7	the estimate	the pictures' size, the memory, and a rough time
8	Start export, Defaults	**Start export** starts; **Defaults** puts every choice back to the config's defaults
9	Into	the folder the export will write

Before you start

- **Load the world you want to export.** Single player is simplest. On a multiplayer server, read chapter 7 first: only the machine that runs the world knows where every location is.
- **Stay in the world until the export says Done.** You can keep playing while it runs; it works in the background. Leaving the world before the maps are written stops the export.
- **Nothing you do in the world changes the export.** It copies the land as the game generated it, not your buildings or digging.

Export with the window

1. **Press F7.**

   You see: the export and import window, on its **Export this world** tab.

2. **Leave every map ticked.** Point at a map's name to read, below the list, what it holds and what it costs.

3. **Keep the size at 4096.** See "Which size?" below.

4. **Leave Heightmap Amount at 1, Sea Level at 0.5 and Heat Scale at 10**, leave **Fine heights** on **auto**, and leave the options under them as they are.

   You see: the line *Waterline at 0.3; heights from -30 m to 170 m, in steps of 0.003 m.* These are the game's own heights, which put the sea at 30 m: 60 m under the sea to 140 m above it.

   If your world has mountains higher than that, they are clipped to the top of the scale; the export tells you how many pixels. Type 2 in Heightmap Amount to keep them (*heights from -30 m to 370 m*, up to 340 m above the sea), and read the note under "The heightmap" in chapter 4 first. A Better Continents world can be far taller: chapter 4's "How high can it go?" says which Amount to pick.

5. **Read the estimate** under the options: how big the pictures are before compression, how much memory the export needs, and roughly how long it takes. Notes under it warn about big sizes.

6. **Press Start export.**

   You see: two progress bars, the current step and *Overall*. The F9 box shows the same, and the console (F5) says *World export started*.

7. **Wait.** Close the window with F7 if you like, and keep playing.

   You see: at the end, the window's *Last export* part, with the folder and a short summary, and in the console *World export finished in … s*.

Every map in one line

Tick	Files	What it holds
Heightmap	`heightmap.png`	how high the ground is, in 16-bit grey
Biomes	`biomemap.png`, `biomemap.txt`	which biome is where, one colour per biome
Locations	`locationmap.png`, `locationmap.txt`	one dot per location: start temple, bosses, traders, dungeons and the rest
Forest	`forestmap.png`	how dense the forest is, in 16-bit grey
Heat	`heatmap.png`	the Ashlands heat that burns boats and players, in 16-bit grey
Cold	`coldmap.png`	where the Deep North is, which decides its weather, snow and calm sea, in 16-bit grey
Alt biomes	`altbiomemap.png`, `altbiomemap.txt`	the alt biomes (Dark Meadows, Wolf Mountain...) exactly where the game put them
Lava	`lavamap.png`	where the Ashlands lava is
Moss	`mossmap.png`	the Mistlands ground cover
Paint	`paintmap.png`, `paintmap.txt`	the paint on the ground of every biome, such as snow; it takes the most space
Sources	`sources/`	a Better Continents world's own maps, as it was made (a location exclusion mask and a cold map of its own included); nothing for a vanilla world
New World preset	a file in `BetterContinents/presets/`	the preset `<world> <date> <time>`, ready in the New World screen, for exports up to 8192 px
Editor pack (off unless you tick it)	`editor/`	the ground as 32-bit floats and masks, for Gaea and other terrain editors (chapter 10)

Below the map switches the window has one more block, **Editor pack (Gaea and other terrain editors)**. It is off by default, and nothing in the export changes unless you tick it. Chapter 10 explains it.

The export also always writes `world.json`, one line with the world's radius, edge size and layout, which the other mod BetterWorlds reads (chapter 3, "Using the export with BetterWorlds"). It has no tick box.

With the heightmap ticked, the **Fine heights** choice can add one more file, `heightmap-fine.png`: finer heights for tall worlds. On **auto** it is written only from a Heightmap Amount of about 5, so a normal export has none (chapter 4).

> **Try this:** the first time, keep everything ticked. A map you untick is left to the game in the new world: the export then writes settings that let the game make that part itself, from the new world's seed. Only the heightmap is different: without it the new world gets a completely new land, which the other maps no longer fit.

Which size?

Size	One pixel is	Memory while it runs	Memory while the preset is made	Rough time
1024	20.5 m	about 85 MB	about 20 MB	well under a minute
2048	10.3 m	about 99 MB	about 51 MB	well under a minute
4096 (default)	5.1 m	about 99 MB	about 205 MB	under a minute
8192	2.6 m	about 99 MB	about 819 MB	a minute or two
16384	1.3 m	about 99 MB	no preset: `bc_import` makes it, with about 1.4 GB	a few minutes

The memory while it runs and the rough time are the window's own guesses, as it shows them; with a `heightmap-fine.png` (chapter 4) it says 112 MB from 2048 px up. The export writes each picture while it samples the world, so it holds about the same memory at any size. The memory for the preset is the window's estimate too, which follows what was measured on one computer with every map ticked, on top of what the game itself uses. The window's time is a guess from the size alone; measured on one computer, a dedicated server, an 8192 px export took 43 s and a 16384 px export about 3 minutes. Your computer and your world decide the real numbers. The pixel sizes are for a normal world, 21 km across; a version 12 world with another World Size has larger or smaller pixels, because the pictures span its World Size + Edge Size on each side.

- **4096 is right for almost everyone.**
- **2048 and up** keep the alt-biome map exact. Below 2048, alt-biome borders can move by a pixel.
- **8192 and 16384** give the finest detail and make the biggest files, and every player who joins a world made from them downloads the maps (chapter 9). 16384 is meant for worlds bigger than normal: at 2 m a pixel it covers a world 32.7 km across.
- **For cut and paste between worlds**, export both worlds at the same size (chapter 5).

> **Good to know:** above 8192 px the export does not make the New World preset itself: building it decodes every map, about 1.4 GB more memory at 16384 px (measured on one computer). The window, the summary and `README.txt` say so. Make the preset afterwards with `bc_import` in the main menu (chapter 3, way B), where the game holds no world. At 8192 px and below, if your computer is short of memory, untick **New World preset** and do the same.

> **Good to know:** the window offers sizes up to `Largest Size` in the config (chapter 1), 16384 unless someone lowered it. To make a world from 16384 px pictures, `Max Map Size` in the `[07 BetterContinents.Misc]` section must be 16384 too, its default, on the computer that reads the pictures: where you run `bc_import`, or make the world with From Config.

Export with the console instead

1. **Press F5.**

   You see: the console.

2. **Type `bc_export` and press Enter.**

   You see: *World export running; 'bc_export status' shows progress, 'bc_export cancel' stops it.*

3. **Type `bc_export status`** whenever you want to see how far it has got.

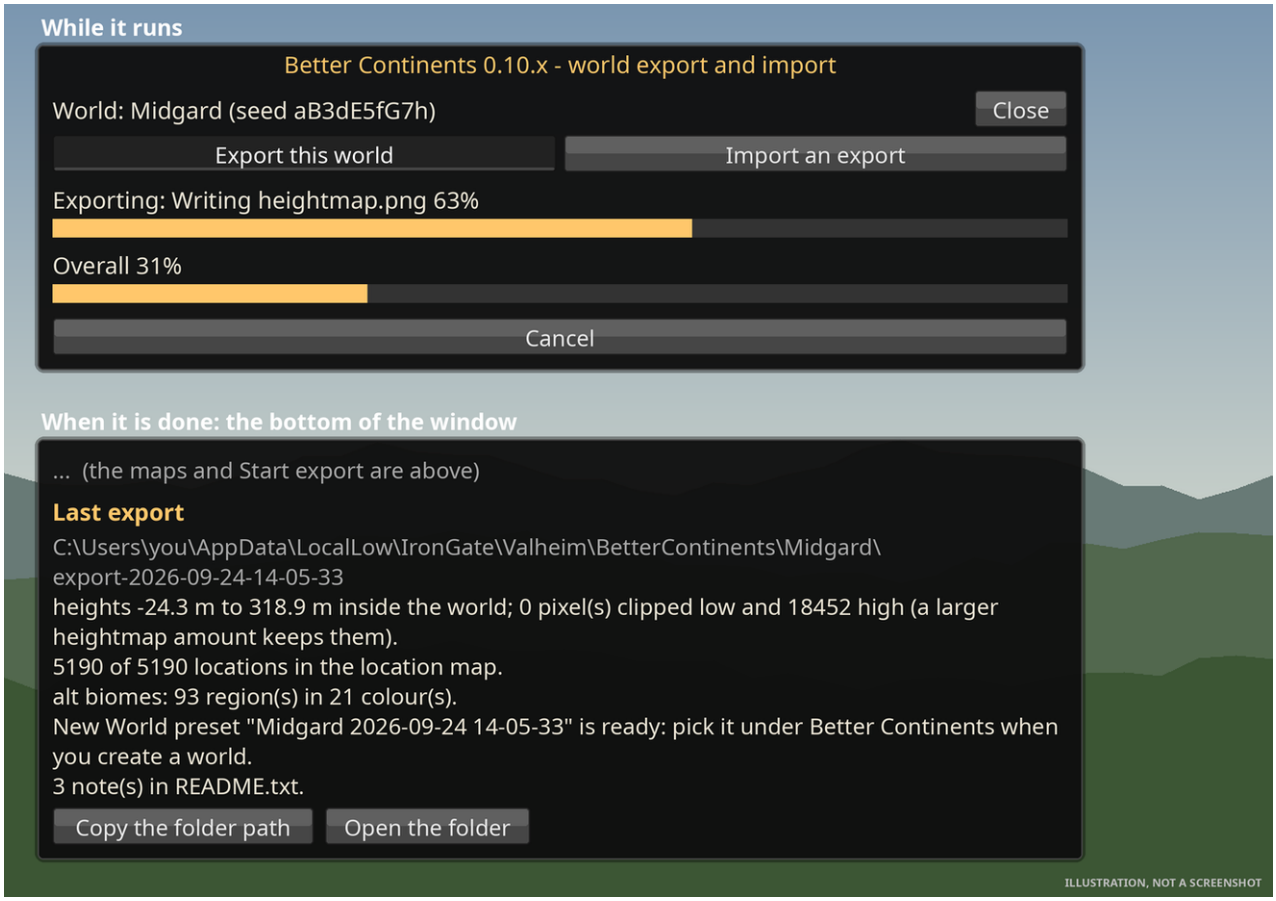
More examples:

```
bc_export                every map, 4096 px, the usual settings, and a preset
bc_export 2048           the same at 2048 px
bc_export amount=2       heights stored for Heightmap Amount 2, to keep ground over 140 m
  above the sea
bc_export 8192 nopreset  8192 px without a preset (make it later with bc_import)
bc_export 16384          16384 px; the export makes no preset at this size (make it with
  bc_import)
bc_export amount=20      heights stored for Amount 20, up to 3,940 m above the sea, with
  heightmap-fine.png
bc_export fine=off       no heightmap-fine.png, whatever the Amount
bc_export nopaint        everything but the paint map
bc_export nocold         everything but the cold map
bc_export 4096 editor    the usual export and the editor pack for Gaea (chapter 10)
bc_export status         how far it has got
bc_export cancel         stop the export and delete what it wrote
bc_export help           the full list of options
```

Appendix A lists every option.

> **Good to know:** `bc_export` works for every player; you do not need devcommands (cheats). On a server, the server decides whether players may export (chapter 7).

What the progress looks like



Illustration, not a screenshot: the window while an export runs, and the summary at the bottom of the window when it is done. The numbers are examples.

The steps come in this order. Each picture is written while it is sampled, so most steps say *Sampling*; maps you unticked are skipped:

Sampling zone corners → *Sampling heights* (the heightmap, lava, moss and paint maps together) → *Writing heightmap-fine.png* (only with fine heights) → *Sampling biomes* → *Sampling forest* → *Sampling heat* → *Sampling alt biomes* → *Placing locations* → *Writing locationmap.png* → *Copying sources* → *Sampling cold* → *Sampling the editor pack* (with the editor pack, chapter 10) → *Making the New World preset* (up to 8192 px) → *Writing export.cfg, README.txt and manifest.json* → *Done*.

When it is done, the console shows a summary like this one (your numbers differ):

```
World export finished in 184 s: C:\Users\you\AppData\LocalLow\IronGate\Valheim\BetterContinents
  \Midgard\export-2026-09-24-14-05-33
World export: heights -24.3 m to 318.9 m inside the world; 0 pixel(s) clipped low and 18452
  high (a larger heightmap amount keeps them).
World export: 5190 of 5190 locations in the location map.
World export: alt biomes: 93 region(s) in 21 colour(s).
World export: New World preset "Midgard 2026-09-24 14-05-33" is ready: pick it under Better
  Continents when you create a world.
World export: 3 note(s) in README.txt.
World export: README.txt in the folder says how to make a world from it, in three ways.
```

- **heights**: the lowest and the highest ground inside the world, counted the game's way (the sea is at 30 m). *No pixel clipped* means every height fitted into the picture; a count of pixels clipped high means some ground is higher than the picture can hold (140 m above the sea at Heightmap Amount 1, which the game counts as 170 m), and export again at a larger Amount to keep it: 2 holds 340 m above the sea (chapter 4).
- **cold**: the number of pixels that are Deep North, *cold: the Deep North is N pixel(s).*, or *cold: the world's own cold map was copied* when a Better Continents world has one of its own (with Sources on). With the editor pack, one more line says its size, its Heightmap Amount and how many masks it wrote, and a *WARNING* when some ground is above what that amount holds (chapter 10).
- **fine heights**: only when the export wrote `heightmap-fine.png`, for example *fine heights: heightmap-fine.png refines 1.1% of the pixels from steps of 24.7 cm to 1.5 cm.*
- **locations**: how many locations the location map holds. A line with *WARNING* means some are missing; chapter 7 explains why.
- **alt biomes**: how many alt-biome regions, in how many colours.
- **New World preset**: the preset's name. It is ready in the New World screen. Above 8192 px the line says instead *no New World preset at this size*, with the `bc_import` line that makes it.
- **notes**: things worth knowing about this export, written in `README.txt`.

Where the folder is

Every export gets a folder of its own, named after the time it started:

```
<save folder>/BetterContinents/<world>/export-2026-09-24-14-05-33/
```

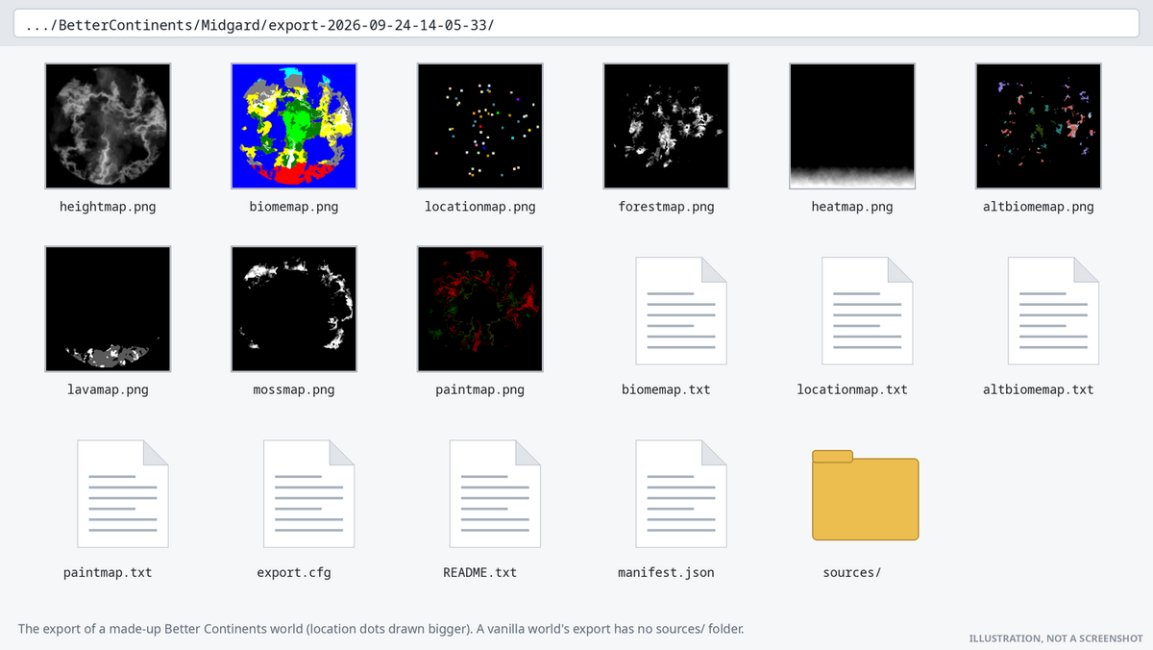
The save folder is the one where Valheim keeps your worlds and characters:

Where Valheim runs	Save folder
Windows	`C:\Users\<you>\AppData\LocalLow\IronGate\Valheim`
Linux, the native game	`~/.config/unity3d/IronGate/Valheim`
Linux, the Windows game under Proton	`~/.local/share/Steam/steamapps/compatdata/892970/pfx/drive_c/users/steamuser/AppData/LocalLow/IronGate/Valheim`
a dedicated server	the server's own save folder (chapter 7)

The easiest way there: after the export, press **Open the folder** in the window, which opens it in your file manager, or **Copy the folder path** and paste the path into your file manager's address bar. On Windows the AppData folder is hidden: paste

`%USERPROFILE%\AppData\LocalLow\IronGate\Valheim\BetterContinents` into the address bar of File Explorer.

> **Good to know:** an export never goes into a world's own save folder (`worlds_local`). A file there could be mistaken for part of a world save, so the export refuses such a folder.



Illustration, not a screenshot: an export folder, with every picture shown small.

What each file is

File	What it is	Edit it?
`heightmap.png`	how high the ground is, 16-bit grey: black is 60 m under the sea, white 140 m above it at the default Heightmap Amount 1 (340 m at Amount 2). It records its Heightmap Amount and Sea Level inside the PNG	yes
`heightmap-fine.png`	only with fine heights: finer steps for `heightmap.png`, 8-bit grey, with a record of the `heightmap.png` it belongs to (chapter 4)	no: after editing `heightmap.png`, delete it or export again
`biomemap.png`	which biome is where	yes
`biomemap.txt`	the biome colours (the biome map's legend)	rarely
`locationmap.png`	one dot per location	yes, carefully
`locationmap.txt`	which colour is which location	when you add a colour
`forestmap.png`	forest added to the game's own	yes
`heatmap.png`	the Ashlands heat	yes, carefully
`coldmap.png`	where the Deep North is (its weather, snow and calm sea), 16-bit grey. If the world had a cold map of its own, this is that map, byte for byte	yes, carefully
`no_spawns_mask.png`	only from a Better Continents world that has a location exclusion mask (and with Sources on): no location is placed where it is above black. Also in `sources/`	yes, carefully

File	What it is	Edit it?
`altbiomemap.png`	the alt biomes as the game placed them	yes
`altbiomemap.txt`	which colour is which alt biome, with a count of regions	when you add a colour
`lavamap.png`	the Ashlands lava	yes
`mossmap.png`	the Mistlands ground cover	yes
`paintmap.png`	the paint on the ground (snow and the like)	rarely; delete it after big height changes
`paintmap.txt`	a note that the paint map needs no colour list	no
`terrainmap.png`, `vegetationmap.png`, `spawnmap.png` and their `.txt`	only from a Better Continents world that had them: its ground colours, which plants grow where, which creatures appear where	yes
`lakes.txt`, `clearmask.bin`	only from a world that has lakes above the sea, or a mask of ground kept free of plants (an experimental part of Better Continents, in Tarn's file formats): the world's lakes and its mask, so a new world made from the folder has the same. See the Water and lakes guide	only with that guide
`sources/`	a Better Continents world's own maps and settings, as it was made; not loaded	no
`world.json`	the world's radius, edge size and layout, one line, for BetterWorlds	no
`editor/`	only with the editor pack: the ground as floats, masks and a description (chapter 10)	in an editor, not by hand
`export.cfg`	the settings that load this folder (appendix B)	only to change a setting
`README.txt`	the export's own instructions, and notes about this export	no
`manifest.json`	every number about the export, for tools (appendix C)	no

Cancel an export

Press **Cancel** in the window, or type `bc_export cancel` in the console.

You see: *World export cancelled; its partial files were deleted.* Everything the export had written, its preset included, is gone again.

3 Make a new world from an export

> **In short:** an export always becomes a **new** world. The easy way: in the main menu make a New World, pick the preset `<world> <date> <time>` in the Better Continents box, and press **Done**. After editing the pictures, type `bc_import` in the console first. To keep editing and testing, use `bc_import ... config`, or set `Directory` in the config to the export folder.

Why it has to be a new world

Better Continents copies its settings and every map into a world when the world is created, and that world keeps its copy for good. Changing a preset, the config or the pictures afterwards changes nothing in a world that already exists. So an export always becomes a new world, next to the old one, and the old world stays as it is.

(Changing the maps of an existing world is possible with Better Continents' `bc` cheat commands, which are for the host or single player: on a player's own computer a change reaches only that player's copy of the settings, never the world. After each change Better Continents regenerates the world's zones itself, leaving the zones around what players built or worked, and around other players; the ground of those zones follows the change too, so a base can be left standing in the air or buried. It is meant for test worlds, and this guide does not cover it.)

Four ways

Way	Best when	Reads the pictures	Needs the folder afterwards
A. Pick the preset	right after an export	as they were when the export finished	no
B. `bc_import`	after you edited the pictures	as they are when you run `bc_import`	no
C. The config	while you keep editing and testing	as they are when you create each world	yes
D. `Directory` **only**	you would rather change one setting than type a command, or you moved the folder	as they are when you create each world	yes

Way D needs Better Continents 0.9.4 or later.

Way A: pick the preset

Every export up to 8192 px makes a preset for you, named after the world and the time of the export, for example `Midgard 2026-09-24 14-05-33`. A 16384 px export makes none: use way B to make it.

1. **Leave your world:** press Esc and choose **Log Out**.

   You see: Valheim's main menu.

2. **Press Start Game, pick your character and press Start.**

   You see: the **Select World** screen, with your worlds.

3. **Press New World.**

   You see: the New World panel with **Name**, **Seed** and the **Better Continents** box.

4. **Type a name for the new world.** It needs at least 5 characters; **Done** stays greyed out until then.

5. **Type the seed.** For the closest copy of your old world, type the old world's seed: the F9 box and the export's `README.txt` show it. Another seed keeps the land, the biomes and the locations, but changes

the small things the pictures do not hold, such as where each tree and rock stands and the rooms of dungeons.

6. **In the Better Continents box, open the list and pick your preset.**

   You see: the preset's picture, a small map of the world.

7. **Press Done.**

   You see: Valheim creates the world and loads it. The loading screen stays up a moment longer while Better Continents draws the map. (Before 0.9.2, the message *Better Continents: generating minimap …* counted up instead.)



Illustration, not a screenshot: where the Better Continents preset list sits on the New World panel. The real screen looks different.

	Part	What to know
1	Name	at least 5 characters
2	Seed	the old world's seed for the closest copy
3	the Better Continents box	the list of presets: *Disabled*, the presets that come with the mod (Continents, Pangaea, Swirls, Vanillaish), yours, and *From Config*
4	the preset's picture	a small map drawn from the preset's heightmap and biome map
5	Done	creates the world

Good to know: the preset is a copy of the export folder as it was when the export finished. You can move or delete the folder and the preset keeps working. It does not see changes you make to the pictures afterwards: use way B for those.

Watch out: if the Better Continents box says **Disabled** when you press Done, the world is made without Better Continents: an ordinary Valheim world. Pick the preset first.

Watch out: a world made from an export with a `heightmap-fine.png` (chapter 4) needs Better Continents 0.10.3 or later, on your computer, on the server and for every player. Better Continents 0.10.0 to 0.10.2 cannot read it, and would give it the game's own land.

Way B: bc_import, after editing the pictures

`bc_import` makes the preset again from the pictures as they are now, and selects it for you. It works in the main menu and in a world, for every player. It is also how a 16384 px export gets its preset: run it in the main menu, where no world takes up memory. Making the preset of a 16384 px export took about 1.4 GB of memory, on top of what the game itself uses, and 19 s, measured on one computer.

1. **Press F5** to open the console.
2. **Type `bc_import list` and press Enter.**

   You see: a numbered list of your exports, newest first, for example:

   ```
   Export folders in C:\Users\you\AppData\LocalLow\IronGate\Valheim\BetterContinents, newest
     first:
      1. Midgard  2026-09-24 14:05  4096 px  (preset made)   Midgard/export-2026-09-24-14-05-33
      2. Midgard  2026-09-20 21:17  2048 px   Midgard/export-2026-09-20-21-17-02
   'bc_import <number>' makes that one a New World preset; add 'config' to load it from the
     config instead.
   ```

3. **Type `bc_import 1`**, with the number of the export you edited.

   You see: lines like these, the last ones a few seconds later:

   ```
   World import: making the preset "Midgard 2026-09-24 14-05-33" from
     C:\...\Midgard\export-2026-09-24-14-05-33. This takes a few seconds; the game keeps
     running.
   World import: Preset "Midgard 2026-09-24 14-05-33" is made and selected: create a New World
     to use it.
   World import: 9 map(s): heightmap, biomemap, ...; Heightmap Amount 1, sea level 0.5, Biome
     precision 0, world 10000 m + 500 m edge.
   World import: saved as C:\...\BetterContinents\presets\Midgard 2026-09-24
     14-05-33.BetterContinents.
   ```

   When the folder has a `heightmap-fine.png`, the third line ends with *fine heights 4 bits* (chapter 4).

4. **Make a new world as in way A.** The preset is already picked in the Better Continents box.

Other ways to name the folder:

Type	To import
`bc_import`	your newest export: of the world you are in; in the main menu, the export you made last, else the newest of the world you played last
`bc_import 2`	the export on line 2 of `bc_import list`
`bc_import Midgard`	the newest export of the world Midgard
`bc_import D:\Maps\My island`	any folder of Better Continents maps, even one you made yourself; spaces need no quotes
`bc_import status`	how the last import went
`bc_import edited`	not for presets: it brings back the ground you made in Gaea or another terrain editor, as a new export folder (chapter 10)
`bc_import help`	all of the above

> **Good to know:** making the preset again replaces the old preset of the same name that this folder made. A preset of the same name that something else made is kept: it is renamed, with `.old-<time>` added to its name.

Way C: the config, for editing and testing

With way C, every new world reads the pictures from the folder at the moment you create it. Change a picture, make another new world, and you see the change: no `bc_import` in between.

1. **In the console, type** `bc_import 1 config` (or press **Use as config** on the Import tab).

   You see:

   ```
   World import: BetterContinents.cfg now loads C:\...\Midgard\export-2026-09-24-14-05-33 (31
     setting(s) changed); the preset is "From Config".
   World import: your config as it was is saved as
     C:\...\BepInEx\config\BetterContinents.cfg.before-import-2026-09-24-15-12-40.
   World import: create a New World now; it reads the PNGs in the folder as they are when you
     create it. No restart is needed.
   ```

2. **Edit the pictures, and save them.**
3. **Make a new world as in way A.** The Better Continents box already says **From Config**.

   You see: a world made from the pictures as they were when you pressed Done.

4. **Look around, leave, change the pictures again, and make another new world.** Delete the test worlds you no longer need on the Select World screen.

> **Try this:** give test worlds short numbered names, such as `Test01`, `Test02`, so they are easy to find and delete later.

> **Watch out:** with way C, a world reads the folder when you press Done. Do not move or rename the folder while you still make worlds from it; if you do, point `Directory` at its new place (way D). A world that exists has its own copy and does not care.

To go back to your old settings, delete `BetterContinents.cfg` and rename the file ending in `.before-import-<time>` back to `BetterContinents.cfg`. Better Continents reads the file again within a second.

Doing it by hand: open `export.cfg` in a text editor, copy all of it, paste it at the very end of `BetterContinents.cfg` and save. A later line wins over an earlier one, so the pasted lines replace your settings, and `SelectedPreset = From Config` picks From Config. Keep a copy of your config first.

> **Watch out:** every Better Continents setting outside the Export section is a default for **new** worlds, with these exceptions: `Settings Transfer Rate`, `Map Memory` and `Max Map Size` belong to your computer and apply at once; `Wide Sectors` is read when a world loads too (leave it at Auto); and `Override version` is read at every save, so an edit changes the next save of a loaded world (leave it empty). Changing any other one never changes a world that exists.

Way D: set only Directory

With way D you change one setting by hand, `Directory`, and Better Continents takes the rest from the folder's `export.cfg` each time you make a world. As with way C, every new world reads the pictures as they are when you create it, but your own config is left as it is. Way D needs Better Continents 0.9.4 or later. Older versions read only the pictures this way and kept every other setting of your config, such as the default Heightmap Amount of 1, so a 0.9.x export (stored for Amount 2) came out almost all ocean (chapter 8).

1. **Copy the folder's path:** press **Copy the folder path** in the export window (chapter 2), or copy it from your file manager's address bar.

2. **Open** `BetterContinents.cfg` (in `BepInEx/config/`) in a text editor, or open Configuration Manager.
3. **In the section** `[00 BetterContinents.Debug]`**, set** `Directory` **to the folder's path, and save.** Leave every other setting as it is. For an export you copied to `D:\Maps\Midgard`, for example:

   ```
   [00 BetterContinents.Debug]
   Directory = D:\Maps\Midgard\export-2026-09-24-14-05-33
   ```

4. **Make a new world as in way A, but pick From Config** in the Better Continents box.

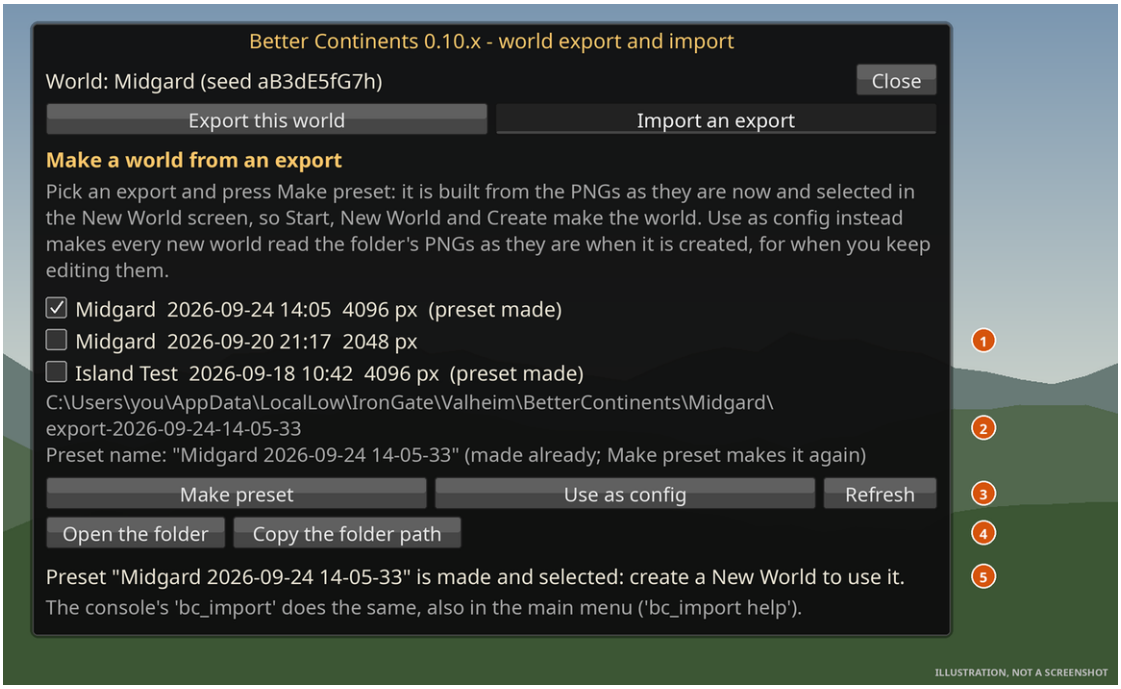
   You see: a world made from the pictures and the export's settings. The log (chapter 8) says *Directory … holds a world export, so the new world uses its export.cfg over BetterContinents.cfg* and lists every setting that differs from your config, for example *[01 BetterContinents.Global] Skip Default Locations = true (BetterContinents.cfg: false)*.

Watch out: with way D, `export.cfg` wins. Every setting it lists comes from it, whatever `BetterContinents.cfg` says; only the settings it does not list come from your config. To use a value of your own, for example another Heightmap Amount, change it in `export.cfg`, or delete `export.cfg` to take every setting from `BetterContinents.cfg` (the heights then still need the Heightmap Amount and Sea Level Adjustment the export recorded there: `Heightmap Amount = 1` and `Sea Level Adjustment = 0.5` for a default export). The same holds for way C, which points `Directory` at the folder too.

Good to know: way D writes nothing: `BetterContinents.cfg` keeps your values and `export.cfg` is only read. `Directory` is always the one in `BetterContinents.cfg`, so a folder you moved or copied works from where it is now; the `Directory` line inside `export.cfg` names the folder it was written in, and is ignored. A folder without `export.cfg`, such as maps you painted yourself, works as it always did: every setting comes from `BetterContinents.cfg`. A preset brings its own settings, so `Directory` does not change it.

The Import tab: ways B and C with buttons

In a world, press F7 and click **Import an export**:



Illustration, not a screenshot: the Import an export tab, its parts numbered.

	Part	What it does
1	your exports	newest first; *(preset made)* means it has a preset, *(unfinished)* means the export did not finish
2	the folder and the preset name	where the pictures are, and the name its preset gets
3	Make preset, Use as config, Refresh	way B, way C, and read the folders again
4	Open the folder, Copy the folder path	to find the pictures
5	the result	what the last import did

Check the new world

- The F9 box shows the new world's name.
- Open the map (M): the land you have explored looks like the pictures. You start at the start temple, where the location map put it.
- `bc_import status` in the console says how the last import went.
- `bc info` in the console (cheats on; it works on the machine that runs the world, and on a player's computer, with the Server Devcommands mod, it shows the settings the server sent) prints the world's settings version first: 12 for most worlds made from an export; 11 for a world made from an export of a world that was resized under 0.9.x (or made under Expand World Size at another size), from a preset of the mod or of 0.9.x, or while `Override version` is 11 (appendix B).

Using the export with BetterWorlds

BetterWorlds is another mod, made by someone else, with its own documentation; it is not part of Better Continents. It can read a Better Continents export folder as it is. The export writes what it needs to rebuild the same world without guessing:

- `coldmap.png`, the Deep North, so the rebuilt world keeps its weather, snow and calm sea (chapter 4);
- `no_spawns_mask.png`, the location exclusion mask, when the world has one;
- `world.json`, one line with the world's radius, edge size and layout (`scaled` for a world of the newest settings version, `vanilla` for a version 11 one);
- in `manifest.json`, the settings version, the export revision, the world's size and edge, and the sizes of the files (appendix C). The format is still `bc-export/1`: every one of those keys is new, none changed.

The export's `README.txt` names BetterWorlds' own ways in. Use the command names it gives; they are BetterWorlds', and its documentation says what they do:

- type `bw_import newest` in the console, or
- use *Import project* in its *Utilities & Layers*, or
- run `python3 tools/betterworlds.py --migrate <this folder> --output <folder>`.

> **Good to know:** `export.cfg` sets `Mountain Noise` and `Mistlands Noise` to 0, because the heights of the export already hold that noise (appendix B), and a rebuilt world must not add it a second time. The noise is baked into the heightmap at the size of a pixel, so detail finer than a pixel (the noise has features of 2.5 m and 10 m) is not carried. BetterWorlds and a Better Continents rebuild both start from those baked heights.

> **Watch out:** BetterWorlds takes at most 256 MiB for each map file and skips a larger one. If a map of your export is larger, the export says so in `README.txt` and in the manifest's notes, so that you can export again at a smaller size.

4 Edit the pictures

In short: keep every picture a square PNG of the same size, north at the top. Keep `heightmap.png`, `forestmap.png`, `heatmap.png` and `coldmap.png` 16-bit grey. Paint the colour maps with a hard pencil in exact colours. Then make the preset again with `bc_import` (chapter 3).

The golden rules

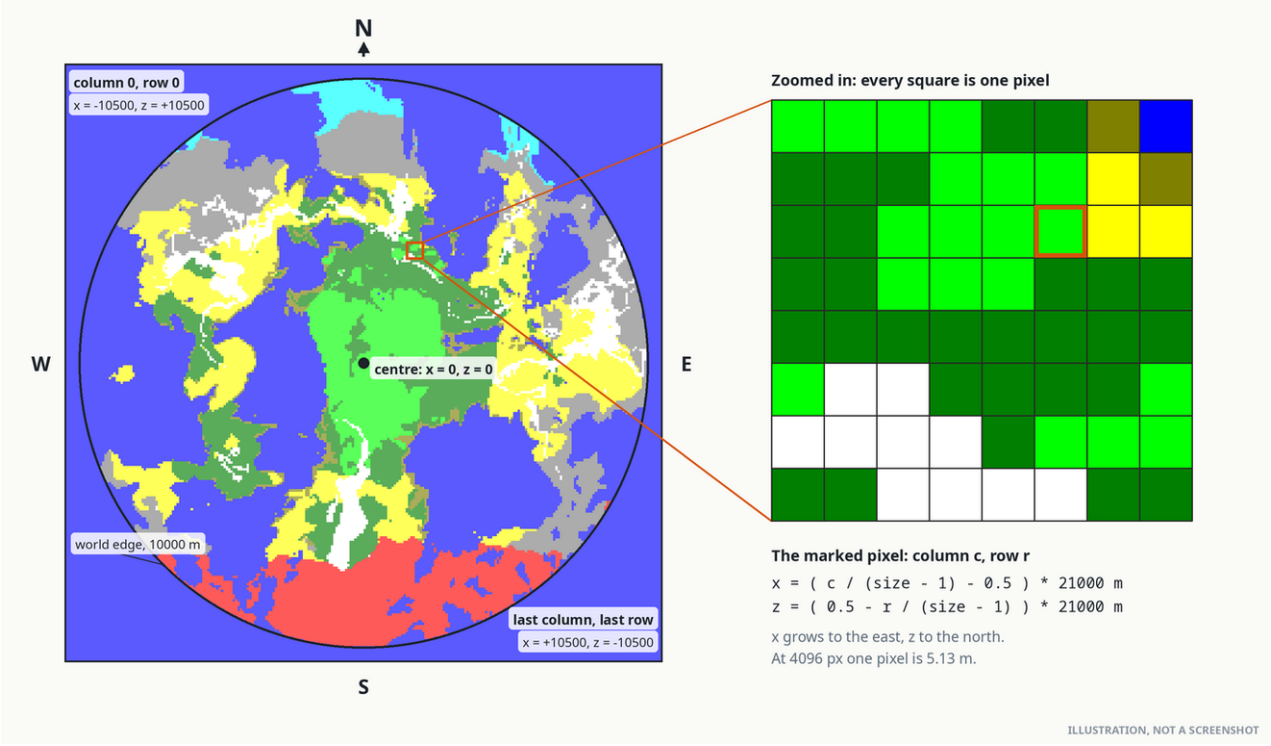
1. **Keep the file names.** The import looks for these exact names.
2. **Keep every picture square and the same size.** Never crop and never change the canvas size. Every picture covers the whole world square, 21 km on a side in a normal world (`totalSize` in `manifest.json` says for yours), so a cropped picture stretches the world.
3. **North is at the top, east on the right.** The middle of the picture is the middle of the world; the corners lie beyond the world's edge.
4. **Save as PNG, never as JPEG.** JPEG changes colours and heights a little everywhere.
5. **Keep `heightmap.png`, `forestmap.png`, `heatmap.png` and `coldmap.png` 16-bit greyscale.**
6. **Paint the colour maps (biomes, locations, alt biomes) with hard edges and exact colours.** No soft brushes, no smoothing, no anti-aliasing.
7. **Keep the `.txt` legends next to their pictures.**
8. **If the folder has a `heightmap-fine.png`, delete it after you edit `heightmap.png`**, or export again. It belongs to the `heightmap.png` it was exported with (see "Fine heights: heightmap-fine.png" below).
9. **When you are done, make the preset again** with `bc_import`, or use the config way (chapter 3).

Your image editor

Editor	16-bit heightmap	How to keep 16 bits
GIMP (free)	yes	the PNG opens in 16-bit; save with File > Export As, and in the PNG dialog leave the pixel format on automatic, or pick 16bpc GRAY. Do not change Image > Precision
Krita (free)	yes	the PNG opens in 16-bit grey; save with File > Export, leave "Force convert to sRGB" off and untick "Store alpha channel"
Photoshop	yes	save with File > Save a Copy (or Save As) as PNG. Export As and Save for Web always write 8-bit
Paint.NET	**no**	it keeps every picture in 8 bits per colour. Use it only for the colour maps and for the lava and moss maps

Watch out: editors show grey in different units: 0 to 255 (even in 16-bit pictures), 0 to 100 %, or 0 to 65535 (and Photoshop's 16-bit mode counts to 32768). The percentage is the same everywhere, so this guide gives it too: the waterline is 30 % at the default Heightmap Amount 1 (15 % at Amount 2).

Find a place in the pictures

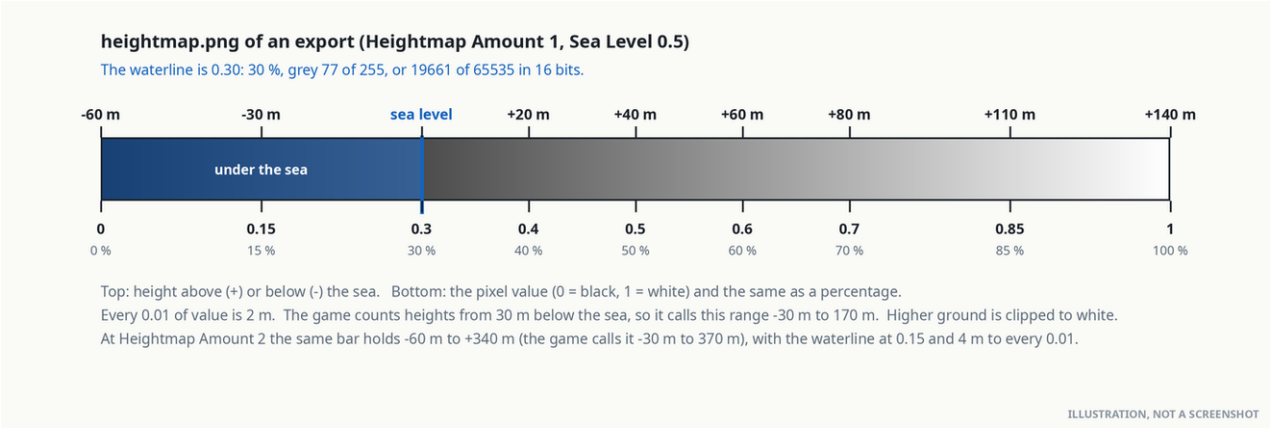


Illustration, not a screenshot: a picture of an export, north up, and how a pixel's column and row give its place in the world.

- The picture is a square; the world is the circle in it, 10,000 m from the middle to the edge in a normal world. Beyond the circle the land falls away into the sea.
- **To find where you stand**, read the F9 box in the game: *You: pixel 2139,1978 @ 4096* means 2139 pixels from the left and 1978 from the top of a 4096 px picture. Your editor shows the same two numbers for the mouse pointer, usually at the bottom of its window.
- The box counts for the size picked in the export window. Pick the size of your export there first.
- One pixel of a 4096 px export is 5.13 m of the world; of a 2048 px export, 10.3 m.

The heightmap: heightmap.png

The brighter a pixel, the higher the ground. An export stores heights like this (Heightmap Amount 1, the default):



Illustration, not a screenshot: the grey scale of an export's heightmap, with the waterline at 0.30.

Height	Value	Percent	8-bit grey	16-bit grey
60 m under the sea (black)	0	0 %	0	0
30 m under the sea	0.15	15 %	38	9830
sea level: the waterline	**0.30**	**30 %**	**77**	**19661**
20 m above the sea	0.4	40 %	102	26214
40 m above the sea	0.5	50 %	128	32768
90 m above the sea	0.75	75 %	191	49151
140 m above the sea (white)	1	100 %	255	65535

The rule behind the table: **height above the sea = 200 × value − 60 m**. Each 1 % of grey is 2 m.

> **Good to know:** ground higher than 140 m above the sea (170 m on the game's own scale) does not fit at Heightmap Amount 1: it is **clipped** to white, and the export tells you how many pixels. To keep tall mountains, export at **Heightmap Amount 2** (type 2 in the window, or `bc_export amount=2`). Amount 2 stores -30 m to 370 m: **height above the sea = 400 × value − 60 m**, the waterline is 0.15 (15 %, grey 38), white is 340 m above the sea, and each 1 % of grey is 4 m. The price is coarser steps: one 8-bit grey level is 1.6 m instead of 0.8 m.
>
> | Height at Amount 2 | Value | Percent | 8-bit grey | 16-bit grey |
> |---|---|---|---|---|
> | 60 m under the sea (black) | 0 | 0 % | 0 | 0 |
> | **sea level: the waterline** | **0.15** | **15 %** | **38** | **9830** |
> | 20 m above the sea | 0.2 | 20 % | 51 | 13107 |
> | 100 m above the sea | 0.4 | 40 % | 102 | 26214 |
> | 200 m above the sea | 0.65 | 65 % | 166 | 42598 |
> | 340 m above the sea (white) | 1 | 100 % | 255 | 65535 |

- **Keep the export's Amount.** `export.cfg` and `manifest.json` carry the Amount and Sea Level an export was made at, and so does `heightmap.png` itself (a PNG text chunk named `BetterContinents`, which reads like `Heightmap Amount = 1; Sea Level Adjustment = 0.5`; some image editors keep it). The new world must read the heights at those values. If it does not, the log warns (chapter 8).
- **Land is above the waterline, sea below it.** The waterline is 30 % at Amount 1 (15 % at Amount 2); a beach is just above it.
- **Use soft brushes, smudging and blur** on the heightmap: smooth slopes look natural. (That is the opposite of the colour maps.)
- **Never use auto levels, stretch contrast, normalize or equalize.** They move every height of the world.
- **Rivers:** paint them into the heightmap, below the waterline. The game's own rivers are switched off in a world made from an export, because the old world's rivers are already in the heightmap.
- **A new mountain also needs the Mountain colour in the biome map**, or it becomes, say, a very tall meadow. The biome map decides the biome; the heightmap only the height.
- **Mountains stop at white.** At Amount 1 the highest ground is 140 m above the sea. For taller mountains, export the world at a larger Amount first (below).
- **The world's edge:** past the world's radius (10,000 m from the middle in a normal world) the ground falls away into the sea, whatever you paint there.

How high can it go?

The Heightmap Amount goes up to 81: `Default Heightmap Amount` in the config (chapter 1) takes 0.01 to 81, and `bc_export amount=` any Amount above 0 up to 81. White is then this high above the sea:

Amount	White is above the sea	One 16-bit step	Use it for
1 (the default)	140 m	3 mm	normal Valheim land; the game's highest mountains are clipped
2	340 m	6 mm	every mountain of a normal world
5	940 m	1.5 cm	high mountains
20	3,940 m	6 cm	very tall ranges
81 (the most)	16,140 m	25 cm	the tallest Better Continents allows

The rule: **white is 200 × Amount − 60 m above the sea**, and black stays 60 m under it at every Amount.

> **Try this:** pick the smallest Amount that holds your highest mountain. The bigger the Amount, the bigger each step of grey, and on gentle slopes big steps show as stairs. Above Amount 5, Better Continents also changes some of the game's rules for high ground, such as where grass grows (the Better Continents Guide's chapter *Heights and the heightmap* explains them).

Fine heights: heightmap-fine.png

From a Heightmap Amount of about 5, one step of grey in `heightmap.png` is more than 1.5 cm, and the export writes a second picture beside it, `heightmap-fine.png`. It is 8-bit grey and the same size, and it splits every step of `heightmap.png` into 16 finer ones (4 bits), so gentle slopes stay smooth. It is 0 under water and on steep ground, so it stays small: in the tests, 30 KB at 4096 px, and 5.7 MB beside an 86.7 MB `heightmap.png` at 16384 px.

The window's **Fine heights** choice (chapter 2), or `bc_export fine=`, decides it:

Choice	`bc_export`	What the export writes
auto (the default)	`fine=auto`	`heightmap-fine.png` with 4 bits from a Heightmap Amount of about 5; below that, none
off	`fine=off`	no `heightmap-fine.png`
4 bits	`fine=4`	it at any Amount, each step split in 16
8 bits	`fine=8`	it at any Amount, each step split in 256

Inside `heightmap-fine.png` is a record of the `heightmap.png` it belongs to: a PNG text chunk named `BetterContinents` that reads like `Fine Format = 1; Fine Bits = 4; Heightmap CRC-32 = 2723AFF1`. The last part is a fingerprint of `heightmap.png`. When you edit `heightmap.png`, its fingerprint changes, and Better Continents no longer uses the fine picture: the log says *was made for another heightmap.png*.

> **Watch out:** after editing `heightmap.png`, delete `heightmap-fine.png` or export again. Then make the preset again with `bc_import` (chapter 3).

> **Good to know:** a world made with fine heights needs Better Continents 0.10.3 or later, and it is always kept compact (chapter 9).

The biome map: biomemap.png

Each biome has one colour. `biomemap.txt` lists them, and these are the colours an export uses:

Swatch	Biome	Hex	R, G, B
	None: left to the game	`000000`	0, 0, 0
	Meadows	`00FF00`	0, 255, 0
	BlackForest	`007F00`	0, 127, 0
	Swamp	`7F7F00`	127, 127, 0
	Mountain	`FFFFFF`	255, 255, 255
	Plains	`FFFF00`	255, 255, 0
	Mistlands	`7F7F7F`	127, 127, 127
	AshLands	`FF0000`	255, 0, 0
	DeepNorth	`00FFFF`	0, 255, 255
	Ocean	`0000FF`	0, 0, 255

- **Every pixel counts as the nearest colour of the list.** A blend of two colours can therefore turn into a third biome: a soft coast between Meadows and Ocean becomes a thin line of Mistlands. Paint with a hard pencil.
- **A place takes the biome that most of its 4 nearest pixels have.** Borders do not blend.
- **You may leave a biome out.** The locations that need it are then skipped: no Mistlands, no Queen.
- **A biome does not change the height.** Paint Mountain where the heightmap is high, and so on.
- **Inside each 64 m square, the ground follows the biomes more loosely than the picture.** Chapter 6 shows how to make it follow more closely.
- The mod's package has `BetterContinents.gpl`, a palette of these colours for GIMP and Krita.

The location map: locationmap.png

The location map is black, with one pixel for each location. `locationmap.txt` says which colour is which location, one line each: `Name: red,green,blue`.

- **To move a location**, paint its pixel black and draw one pixel of the same colour where you want it: the pencil tool, 1 pixel, no anti-aliasing.
- **To remove a location**, paint its pixel black.
- **To add a location**, draw a pixel in the colour of its line in `locationmap.txt`. If the location is not listed, add a line with the game's name for it and a colour no other line uses.

> **Watch out:** three rules the game keeps you to.
>
> - **Two touching pixels of the same colour become one location.** Keep black between them.
> - **One location fits in each 64 m square of the world.** A second one in the same square is dropped. At 4096 px keep dots at least 13 pixels apart; at 2048 px, 7.
> - **The location is built exactly where the dot is, even in the sea.** The game does not check. Put each dot where its location belongs: a boss on dry land of its own biome.

The main locations keep their usual colours in an export:

Swatch	Location	Who or what	R, G, B
	StartTemple	where every player starts	255, 0, 0
	Eikthyrnir	Eikthyr's altar	255, 153, 0
	GDKing	The Elder's altar	0, 255, 0
	Bonemass	Bonemass's altar	0, 255, 255
	Dragonqueen	Moder's altar	74, 134, 232
	GoblinKing	Yagluth's altar	255, 255, 0
	Mistlands_DvergrBossEntrance1	the Queen's entrance	153, 0, 255
	Vendor_BlackForest	Haldor, the trader	0, 0, 255
	Hildir_camp	Hildir, the trader	255, 105, 180

Every other kind of location gets a colour of its own, and the export writes it into `locationmap.txt`: always take colours from your export's `locationmap.txt`, not from memory.

> **Good to know:** you may see several dots for Haldor. The game keeps a few possible places for some locations and builds only one of them: the first one a player comes close to. The new world does the same.

An export from the machine that runs the world holds every location, and its `export.cfg` switches the game's own location placement off (`Skip Default Locations = true`): the new world gets the dots in the map and nothing else. So a dot you delete is a location the new world does not have.

The forest map: forestmap.png

- 16-bit grey. **Black means "the game's own forest here"**; the brighter a pixel, the more trees on top of it.
- In this form the forest map cannot make a place barer than the game makes it. To clear forests, export with **Exact forest** ticked (chapter 2): then black means no trees and white the densest forest, but the game's own forest pattern is part of the picture, which makes it harder to paint.

The heat map: heatmap.png

- 16-bit grey. **Wherever it is above black, the ground counts as Ashlands**: its weather, burning water and lava damage. White is the hottest.
- **Everywhere else it must be exactly black.** A faint grey smear from a soft brush turns that ground into Ashlands.
- **The exported map is exact.** It is above black exactly where the game counts the world as Ashlands, so a world made from it is Ashlands in the same places.
- When you paint Ashlands somewhere new in the biome map, paint heat there too, and a little way into the sea around it: hot water is what burns boats on the way in.

The cold map: coldmap.png

- 16-bit grey. **Wherever it is above black, the ground counts as the Deep North**: its cold weather, its snow and its calm sea. In the Deep North the value is the game's wave fade (a ramp over the 200 m at the border, from 0 to 1) divided by 10, and never less than one grey step; the world reads it back with `Coldmap Scale` 10.
- **Everywhere else it must be exactly black.** A faint grey smear turns that ground into Deep North.
- **A world with a cold map of its own** exports that map, byte for byte, with the world's own `Coldmap Scale` in `export.cfg` (and the export needs **Sources** on to do it). Without a map of its own, the export samples the Deep North where the game puts it.
- When you paint the Deep North somewhere new in the biome map, paint cold there too, as with the heat map.
- A world whose own cold map is not in use (`Coldmap Scale` 0) gets a new map sampled; its own map is still in `sources/`.
- Untick **Cold** (or `nocold`) to leave the map out. The new world's Deep North then follows its biome map, and the game's own rule where the biome map says nothing.

The location exclusion mask: no_spawns_mask.png

A Better Continents world can have a *location exclusion mask*: wherever it is above black (more than 1 % of white), the game places none of its own random locations. It only appears in an export of such a world, with **Sources** on, at the top of the folder and in `sources/`, and a world made from the folder loads it. Keep it as it is, or edit it as a grey picture. The start temple and the locations you place yourself with dots in `locationmap.png` are not blocked by it.

Alt biomes: altbiomemap.png

Alt biomes are Valheim's variations of a biome, such as Dark Meadows or Wolf Mountain. In the alt-biome map, each colour stands for one alt biome, or a set of them, named in `altbiomemap.txt`. A line looks like `Dark Meadows: 2D4613 # 3 region(s)`; everything after `#` is a note.

- **Black means no alt biome.** The export's settings (`Mode = PlantedOnly`) tell the game to add none of its own, so the map is all there is.
- **Paint a region black** to take its alt biome away; **paint it in a colour of the list** to give it that alt biome. Every patch you paint becomes a region of its own.
- **A colour only works on the biome its alt biome belongs to.** Dark Meadows painted over a meadow and the beach next to it changes the meadow and leaves the beach alone.
- **A new colour needs a new line** in `altbiomemap.txt`, such as `Wolf Mountain: 93C7BC`. The usual colours are in the palette `BetterContinents.gpl` and in the Better Continents Guide.
- **A name with `!` in front is forced**: the export adds it where the game's own rules would otherwise refuse that alt biome.
- From 2048 px up, the map matches the game's 12 m alt-biome grid exactly.

Lava, moss and paint

- `lavamap.png` (8-bit grey) only matters in the Ashlands. Lava wherever the grey is above 60 % (153 of 255); black means no lava.
- `mossmap.png` (8-bit grey) only matters in the Mistlands: its ground cover.
- `paintmap.png` (colour) is the paint on the ground in every biome, such as how deep the snow is. It keeps the old paint where you change the heights: after big height changes, delete it and make the preset again, and the game paints the ground itself.

Letting the game make a part itself

Delete that map from the folder, then make the preset again. Two maps need a matching change in `export.cfg` as well, or the new world ends up with none of them:

When you delete	Also set in export.cfg	Or else the new world has
`locationmap.png`	`Skip Default Locations = false` (section `[01 BetterContinents.Global]`)	only the start temple: no bosses, traders or dungeons
`altbiomemap.png`	`Mode = Random` (section `[08 BetterContinents.AltBiomes]`)	no alt biomes at all

> **Try this:** better still, untick the map before you export (chapter 2). The export then writes the right settings by itself.

Make the preset again

Run `bc_import` (chapter 3, way B). If you use the config way (way C) or `Directory` (way D), simply make another new world: since Better Continents 0.9.4 it reads the folder's `export.cfg` too, so a change made there counts as well.

5 Cut and paste between worlds

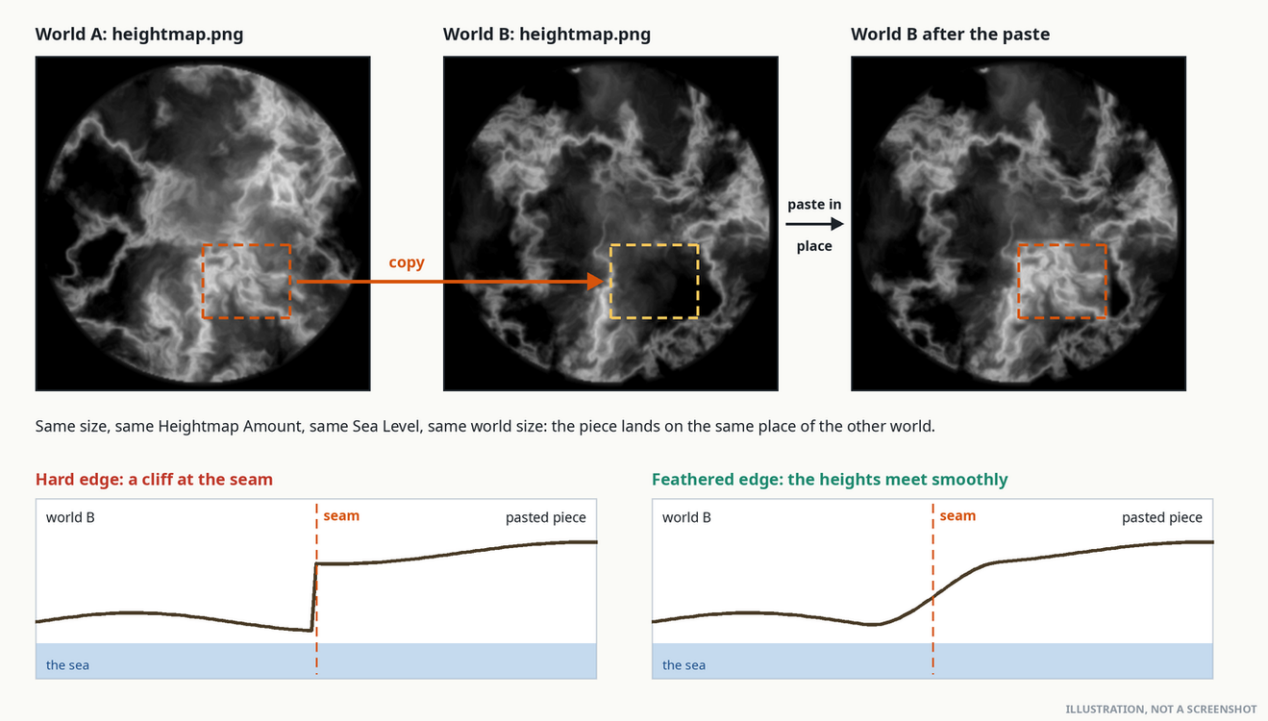
> **In short:** export both worlds with the same size and the usual settings. Cut the same rectangle out of every picture of world A and paste it in place into world B's pictures. Feather the heightmap's edge so the land meets without a cliff; never feather the colour maps. Then `bc_import` world B's folder.

Two exports **line up pixel for pixel** when they agree on four numbers. Then a piece cut from one picture lands on exactly the same place of the other world, and its heights come out the same.

Must match	Where to check it	Usual value
the size of the pictures	`"size"` in `manifest.json`	4096
the Heightmap Amount	`"heightmapAmount"` in `manifest.json`	1 (2 for an export made with `amount=2`)
the Sea Level	`"seaLevel"` in `manifest.json`	0.5
the size of the world	`"totalSize"` in `manifest.json`	21000 (a normal world)

`manifest.json` is a text file in each export folder; open it with a text editor. Hand-made maps drawn for the same Heightmap Amount and Sea Level (1 and 0.5 for a default export), at the same size, line up with exports too.

> **Watch out:** the exports of Better Continents 0.9.x were made at Heightmap Amount 2, and an export made now is at Amount 1 unless you ask for 2. Pictures made at different Amounts do not line up: a pasted piece would come out at the wrong height. Export both worlds again with the same `amount=`, and check that `heightmapAmount` is the same in both `manifest.json` files.



Illustration, not a screenshot: a piece of world A's heightmap pasted into world B, and the seam seen from the side.

A worked example: move a mountain range from world A to world B

This example uses GIMP; other editors have the same tools under similar names.

1. **Export both worlds** with the same size and the usual settings (chapter 2). Check the four numbers in both `manifest.json` files.
2. **Copy world B's export folder**, and work on the copy. The original stays as a spare.
3. **Open world A's `heightmap.png`, and the copy of world B's `heightmap.png`.**
4. **In world A's heightmap, select the rectangle around the mountains** with the Rectangle Select tool. Write down its position and size: GIMP shows them in the tool options.

   You see: the selection's corners, as pixel numbers.
5. **Feather the selection**: Select > Feather, 16 pixels.

   You see: the selection's corners become round. The feathered edge lets the heights of A and B blend over 16 pixels instead of meeting in a cliff.
6. **Copy** (Ctrl+C), **switch to world B's heightmap**, and **Paste In Place** (Edit > Paste In Place).

   You see: the piece sits at the same pixel position it had in world A.
7. **Anchor it** (Layer > Anchor Layer) and **save**: File > Export As, the same file name, 16-bit.
8. **Repeat steps 4, 6 and 7 for every other map**, with the same rectangle and **without** feathering for the colour maps: `biomemap.png`, `locationmap.png` and `altbiomemap.png`. The grey maps, `forestmap.png`, `heatmap.png`, `coldmap.png`, `lavamap.png` and `mossmap.png`, and `paintmap.png`, may be feathered like the heightmap.
9. **Copy the missing legend lines.** For every colour in the pasted piece of `altbiomemap.png` or `locationmap.png` that world B's `.txt` does not list, copy that line from world A's `.txt` into world B's.

   If world B's folder has a `heightmap-fine.png` (exports from a Heightmap Amount of about 5 up have one), delete it: it belongs to the old `heightmap.png`, and Better Continents would not use it with the new one (chapter 4).
10. **Make the preset**: `bc_import` with the path of the copied folder (chapter 3), then make a new world.

The seam

Heights are read smoothly between pixels. Where two heights meet at a hard edge, the step between them becomes a slope just one pixel wide, about 5 m at 4096 px: a cliff. There are three cures:

- **feather the selection** before you copy, as in the example;
- **cut along the sea**, where both worlds are under water anyway;
- **smooth the seam afterwards**, with a soft brush or a blur along it, in 16-bit.

Biomes never blend: a place takes the biome most of its 4 nearest pixels have, so the border sits exactly where the colours meet.

What moves with the paste, and what does not

Moves with the pasted piece	Does not move
the shape of the land and the sea floor	buildings, dug ground, anything players placed: an export never has them
the biomes, and the alt biomes (with their legend lines)	where exactly each tree and rock stands: the new world's seed decides that
extra forest: the pasted woods keep their extra trees	the game's own forest pattern under them, which belongs to the new place
the Ashlands heat, the Deep North's cold, lava and the Mistlands ground cover	the rooms inside dungeons
the locations whose dots are in the piece	locations whose dots you did not paste

Watch out: if both worlds have the same boss in their location maps, the new world gets both: two of Eikthyr's altars, say. Paint the one you do not want black.

Watch out: a dot must not touch another dot of its colour, and two locations never fit in one 64 m square of the world. Look along the edges of the pasted piece of `locationmap.png` for dots that now crowd each other.

6 Biome precision

> **In short:** Valheim decides the ground's biome in squares of 64 m, so on the ground biome borders follow a 64 m grid and small patches can vanish. **Biome precision** 1 to 5 makes the ground follow your biome map more closely, down to 11 m, and a version 12 world can go up to 31, down to 2 m. It is a setting of the world, chosen before you create it: `Biome precision` in `export.cfg` or in `BetterContinents.cfg`. 0 is the game's own way. It works again in Better Continents 0.9.0.

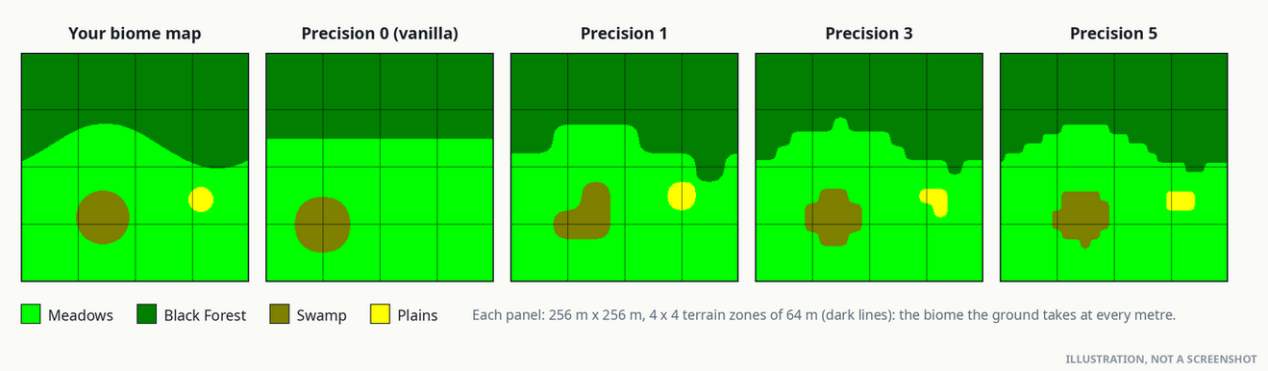
What it does, in plain words

Valheim builds the ground in squares of 64 m, called zones. For each zone it looks at the biome at the zone's four corners only, and everything inside the zone gets its biome from those four corners: the ground's textures, the grass, the plants and trees, and where creatures appear.

So on the ground, the borders between biomes follow the 64 m grid rather than the lines you painted, and a patch of biome that falls between four corners can disappear from the ground altogether, like the small yellow Plains in the picture below.

Biome precision makes each zone look at more points: with precision N, a grid of (N + 1) × (N + 1) cells per zone instead of one.

Precision	Cells per zone	One cell is about
0 (the game's own way)	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 (new worlds only)	8 × 8	8 m
15 (new worlds only)	16 × 16	4 m
31 (new worlds only)	32 × 32	2 m



Illustration, not a screenshot: the biome the ground takes, metre by metre, over 4 × 4 zones, drawn from the same rule the mod uses. At precision 0 the small Plains patch is gone; from precision 1 it is there.

It changes the ground's biome only. **The heights never change**, and neither does anything that is not decided per zone: the places the game picks for locations, for example, already follow the biome map.

When to raise it

- **Your biome map has small islands, thin strips or detailed borders**, such as a narrow swamp along a river, and you want the grass, plants and creatures to follow them.
- **You notice grass or trees of one biome running tens of metres past a border you drew.**

Leave it at 0 for worlds that should behave exactly like the game's own. An export copies the old world's precision into `export.cfg`, so the new world matches the old one unless you change it.

The higher the number, the more the game has to work out each time it builds a piece of ground. Try 2 or 3 first.

Values above 5 work only in a version 12 world (chapter 3 shows how to read a world's version); an older world (settings version 11) reads them as 5. Each 64 m square then has (N + 2) × (N + 2) points to work out when it is built, 1,089 at 31 where the game's own way uses 4, so go up step by step and watch how the game runs.

How to set it

For a new world made from an export:

1. **Open the export's** `export.cfg` in a text editor.
2. **In the section** `[03 BetterContinents.Biomemap]`**, set** `Biome precision`, for example:

   ```
   [03 BetterContinents.Biomemap]
   Biome precision = 3
   ```

3. **Save it, and make the preset again with** `bc_import` (chapter 3). With way C or D, just make another new world: since Better Continents 0.9.4 it reads `export.cfg` in the folder, and the value there wins. (Before 0.9.4, way C needed it set in `BetterContinents.cfg` instead, under the same section.)

   You see: the import's summary says *Biome precision 3*.

In a world that exists, for testing (the host, with cheats on):

1. Open the console (F5) and type `devcommands`.
2. Type `bc b p 3`.

   You see: the loaded ground is built again with the new precision, and a *Zone regeneration* line in the console says how many zones are being reset. The world's zones are regenerated, so plants stand according to the new precision when you come near each zone. The world keeps the new value when it saves.

Watch out: `bc b p` regenerates the world's zones: Better Continents does it itself (or runs the command you put in its *Debug Reset Command* setting, if you did). Every generated zone is reset and generates again when somebody comes near, except the zones within one zone of something a player built, a tombstone, ground a player has worked (a field, a path, a road), or you inside a dungeon, and the zones another player's game has loaded around them. A location that reaches into more than one zone is kept or regenerated whole. Try `bc b p` in a test world, never in a world you care about.

Good to know: precision is a setting of the world. In multiplayer every player gets it from the world; nobody has to set anything on their own computer.

7 Multiplayer and servers

> **In short:** the machine that runs the world (single player, the host, a dedicated server) can always export, and knows every location. A player on a server can export only while the server's **Allow Export** is on, and gets only the locations the map shows. On a dedicated server an admin types `bc_export server`. The server's **Hud** setting decides whether its players see the export HUD.

Who can export

Where you play	Can you export?	Locations in the export
single player	yes	all of them
you host the game for friends	yes	all of them
you joined someone's server	while the server's `Allow Export` is on (the default)	only the ones the map shows: the start temple, bosses, traders
a dedicated server	an admin asks it with `bc_export server`	all of them, in the server's own folder

Why a player's export has fewer locations

Every computer builds the land itself, from the world's settings, so the heights, biomes, forests and the rest of a player's export are complete. Only the machine that runs the world knows where every location is. A player's computer knows just the locations the server shows on the map. So an export made on a player's computer holds only those, and its `export.cfg` lets the game place all the other locations in the new world, by its own rules: they end up in other places.

You see, in the export's summary: *… locations in the location map; WARNING: sampled on a client, which knows only the map-icon locations, so the game places the rest.*

> **Try this:** for a complete export of a server's world, ask an admin to run `bc_export server` (below), or ask the server's owner for a copy of the world and export it in single player.

For server owners: Allow Export

In the server's `BetterContinents.cfg`, section `[09 BetterContinents.Export]`:

```
Allow Export = true
```

- **true** (the default): players connected to the server may export its world.
- **false**: they may not. `bc_export` and **Start export** then say *the server does not allow world export (its Allow Export is off)*, and an export a player is running stops.
- The server itself, a host and a single player can always export.
- A change applies at once, also to players who are already connected, on servers that run Better Continents 0.9.0 or later. With an older server, or one without Better Continents, each player's own `Allow Export` decides.

For server owners: the HUD

On a server that runs Better Continents 0.9.0 or later, the server's `Hud` setting replaces each player's own. Set `Hud = true` in the server's config to give your players the status box and the export window; leave it `false` to keep the HUD hidden. Players can still type `bc_export` when the HUD is hidden, while `Allow Export` is on.

For admins: export a dedicated server's world

A dedicated server has no screen, so an admin asks it from the game.

1. **Be an admin of the server**: your ID must be in the server's `adminlist.txt`.
2. **Join the server.**
3. **Press F5 and type `bc_export server`.** Options work as with `bc_export`, for example `bc_export server 4096 nopaint`.

   You see: *Asked the server to export its world (admins only). It writes under its own save folder; its answer and the export's end come back here and go to its log. 'bc_export server status' and 'bc_export server cancel' go the same way.* If you are not an admin, the game answers *You are not admin*.
4. **Wait.** The server exports its world into its own save folder, in `BetterContinents/<world>/export-<date>-<time>/`. Its messages go to the server's log and, since 0.10, to your console too: `bc_export server status` and `bc_export server cancel` answer in both, and the end of the export comes back to you.
5. **Copy the export folder from the server to your computer.** The easiest place is your own save folder's `BetterContinents/<world>/` folder: then `bc_import list` and the Import tab show it.
6. **Make the preset in your game** with `bc_import` (chapter 3). A dedicated server makes no preset itself: it has no New World screen.

> **Good to know:** a dedicated server holds about 100 MB more memory while it exports, at any size. Measured on one server, a 16384 px export took about 3 minutes and an 8192 px export 43 s.

> **Good to know:** the editor pack of chapter 10 works the same way: `bc_export server 4096 editor` makes the server write its own `editor/` folder, with every location in `protect-locations.png`. Copy the whole export folder, as in step 5.

> **Good to know:** a dedicated server's save folder is the folder that holds its `worlds_local` folder: by default the same place as a game's for the account that runs the server, or the folder given to the server with `-savedir`. Hosting companies usually show it in their file manager.

Share a world made from an export

A world made from an export is an ordinary Better Continents world.

- Players who join download its settings and maps from the server by themselves; they need Better Continents installed too, the same version as the server (a player with another version is refused). Bigger exports (8192 or 16384 px, the paint map) mean a longer download on joining. Chapter 9 shows how to make that download faster (the server's **Settings Transfer Rate**) or skip it altogether (a `.bcworld` file you hand out).
- A dedicated server cannot create a Better Continents world itself. Create the world in your game, then copy the world's folder, `worlds_local/<world>/`, into the server's `worlds_local` folder.
- Biome precision (chapter 6) is part of the world: everyone gets it from the world.

8 When it goes wrong

> **In short:** find what you see in the tables below. Most problems come from one of four things: the new world was not made from the export's settings, a picture lost its 16 bits or its hard edges, a map was deleted without the matching setting, or the export was made on a player's computer instead of the server.

The new world is wrong

What you see	Why	What to do
The new world is almost all ocean, or only mountain tops stick out	the heightmap was read with a Heightmap Amount or Sea Level other than the export's (in `export.cfg`, and recorded inside `heightmap.png`: 1 and 0.5 for a default export, 2 and 0.5 for an export of 0.9.x), such as your config's default Amount of 1 against an export at Amount 2. The log has a warning that names both pairs of values. Most often you set only `Directory` to the export folder and chose **From Config**, and a Better Continents older than 0.9.4 ignored `export.cfg`	update to Better Continents 0.9.4 or later, which then uses `export.cfg` (way D, chapter 3), and make another new world; or use way A, B or C. A `heightmap.png` without its `export.cfg` needs the Heightmap Amount and Sea Level Adjustment it was made for in `BetterContinents.cfg` (chapter 4 says how to read them)
The land is far too high, with hardly any sea	the same mix-up the other way round	the same
Mountain tops are flat, as if cut off	the export clipped ground higher than the picture holds: 140 m above the sea at Heightmap Amount 1 (its summary says how many pixels)	export again at a larger Amount: 2 (`bc_export amount=2`) holds 340 m above the sea, and the most, 81, 16,140 m (chapter 4). Pictures at one Amount do not line up with pictures at another for cut and paste (chapter 5)
Gentle slopes of a tall world look like stairs	the heights are stored at a large Amount, where each step of grey is big, and the fine heights are missing: `heightmap-fine.png` was deleted, written with **Fine heights** off, or not used (the log has a line starting *Fine heights:* that says why)	export again with **Fine heights** on auto, and keep `heightmap-fine.png` with the `heightmap.png` it came with (chapter 4)
The log says *Fine heights: … was made for another heightmap.png*	`heightmap.png` was edited after the export, so its `heightmap-fine.png` no longer belongs to it, and the new world was made from `heightmap.png` alone	delete `heightmap-fine.png`, or export again (chapter 4)
The log says *Fine heights: … Compact Maps is Off*	`Compact Maps` in your config's `[07 BetterContinents.Misc]` section is Off, and a world with fine heights must be compact	set `Compact Maps` back to Auto, its default, and make the world again
An older Better Continents gives the world the game's own land	the world was made with fine heights, which 0.10.0 to 0.10.2 cannot read	use Better Continents 0.10.3 or later, everywhere the world is played
A setting you changed in `BetterContinents.cfg` makes no difference to the new world	`Directory` points at an export folder, and its `export.cfg` lists that setting: since 0.9.4 `export.cfg` wins (ways C and D). The log lists every setting in which the two differ	change the setting in `export.cfg`, or delete `export.cfg` to take every setting from `BetterContinents.cfg`

What you see	Why	What to do
The world is an ordinary Valheim world, with none of the export	the Better Continents box said **Disabled**, or another preset, when you pressed Done	make another new world, with the export's preset picked
Your edits to the pictures are missing	the world was made from the preset the export made, before you edited	make the preset again with `bc_import` (way B), or use the config way (way C or D)
Slopes you edited have become stairs	the heightmap was saved in 8 bits: by Paint.NET, by Photoshop's Export As, or as a JPEG	save it as a 16-bit grey PNG (chapter 4). If the 16 bits are lost, export again
A thin line of Mistlands (grey) along the coasts	soft or anti-aliased edges in `biomemap.png`: a blend of Meadows and Ocean is nearest to the Mistlands colour	paint the edges again with a hard pencil
Some ground has Ashlands weather or burns, where it should not	faint grey in `heatmap.png`, for example from a soft brush	make the heat map exactly black there
No bosses, traders or dungeons, only the start temple	`locationmap.png` was deleted while `Skip Default Locations = true`	set `Skip Default Locations = false` in `export.cfg` and run `bc_import` again (chapter 4)
No alt biomes at all	`altbiomemap.png` was deleted while `Mode = PlantedOnly`	set `Mode = Random` in `export.cfg` and run `bc_import` again
Fewer locations than the old world, in other places	the export was made on a player's computer, which knows only the locations the map shows	export on the machine that runs the world, or ask an admin for `bc_export server` (chapter 7)
A location you drew is missing	its dot touches another dot of its colour; or another location is in the same 64 m square; or its colour is not exactly the one in `locationmap.txt`	keep dots apart (13 pixels at 4096 px) and copy colours exactly (chapter 4)
A boss stands in the sea	its dot is in the sea: the game builds a location where its dot is	move the dot onto dry land of its biome
The ground paint (snow, for example) no longer fits edited mountains	`paintmap.png` still holds the old world's paint	delete `paintmap.png` and run `bc_import` again
You can sail on for ever past the world's edge	the export was made with **Map edge drop-off: off**	export again with *as the world* (the default)
A cliff runs along a pasted piece	the heights of the two worlds met at a hard edge	feather the edge, or smooth the seam (chapter 5)

The editor pack and `bc_import edited`

Chapter 10 explains the pack and the way back. The messages below are the way back's own.

What you see	Why	What to do
this looks like a Normalized build	the returned heights span exactly 0 to 1 while the pack's did not: the editor scaled them to its output range	build with Output Range = Raw, or say the range with `range=lo,hi` (metres) or `unit=metres`
cannot tell how the returned picture is turned … and eight scores	the picture is symmetrical, flat or heavily changed, so no turn fits clearly better than the others	say `orient=` (`probe.tif` shows which)
The terrain is upside down in the editor	the editor's row order differs	nothing to flip by hand: `orient=auto`, or `orient=flipy`
N pixels below X m and M above Y m were clipped: amount=auto keeps them.	the edit goes beyond what the Heightmap Amount and Sea Level hold	add `amount=auto` (and `sealevel=auto` for very deep ground)

What you see	Why	What to do
the TIFF is compressed with LZW …; save it uncompressed or with Deflate, or as R32	the editor saved a compression the way back does not read (LZW, PackBits, JPEG)	save it uncompressed or with Deflate, or as R32
EXR files are not read yet	the build was saved as EXR	build again as TIFF 32-bit or R32
… Better Continents reads pictures up to 16384 wide	the build is larger than the limit	build at 16384 px or less
there is no editor/editor.json to say what the values mean: give unit=metres, or range=lo,hi	the pack's description was deleted	say `unit=metres` or `range=lo,hi`
… holds several files (…): name the one with files=	more than one height file in `editor/return/`, none called `height.tif`, `height.r32` or the like	name the one with `files=` or rename it `height.tif`
… holds no height file or *there is no folder … editor/return*	the editor's result is not there	save the build in the export's `editor/return/`
that export is still being written	the export has not finished	wait for *Done*
This export has no editor/ folder: export with the editor pack on first. (in the window)	the export was made without `editor`	export again with the editor pack on
bc_export: unknown option 'editor' (or `nocold`)	an older Better Continents does not know the word	use a Better Continents that has the pack (the same version on every machine that exports)
The edited ground has no towns, biomes or locations where you moved things	biomes, locations and the other maps do not follow the ground	repaint them in the pictures (chapter 10)

The export does not start or does not finish

What you see	Why	What to do
World export cannot start: the server does not allow world export (its Allow Export is off)	the server's owner switched exports off	ask the owner, or an admin for `bc_export server` (chapter 7)
… Allow Export is off in this game's Better Continents config	your own `Allow Export` is off, and the server sends no setting of its own	set `Allow Export = true` in your config
… an export is already running	one export at a time	wait, or `bc_export cancel`
… no world is loaded	you are in the main menu	load the world first
World export failed: the world was unloaded during the export	you left the world before the maps were written	stay in the world until *Done*; the partial files were deleted for you
… every map is switched off	nothing is ticked	tick at least one map
The summary says *pixel(s) … clipped*	the world has ground more than 140 m above the sea (at Heightmap Amount 1) or 60 m below it	the clipped peaks come out flat; the deep sea floor does not matter. Export at Amount 2 to keep peaks up to 340 m above the sea, or at a larger Amount for taller ground (chapter 4); appendix D has the numbers
The game runs out of memory at 8192 px	making the preset took about 0.8 GB more at that size, measured on one computer	export at 4096, or untick **New World preset** and run `bc_import` later in the main menu
A 16384 px export made no preset	that is how it works above 8192 px: the preset takes about 1.4 GB more memory at 16384 px, so the export leaves it to you	run `bc_import` in the main menu (chapter 3, way B)

What you see	Why	What to do
World export cannot start: the size must be 128 to 4096 pixels (Largest Size in the config) …, or the window offers no 16384	`Largest Size` in the config's Export section is set lower on this computer	set `Largest Size = 16384` (chapter 1)
Locations: the world was still placing its locations … in the notes	the world had not finished placing its locations when the export ran	export again later

The HUD or the keys do nothing

What you see	Why	What to do
No status box, and F7 does nothing	`Hud` is off: on a server, the **server's** `Hud` is used (chapter 7)	turn `Hud` on (chapter 1), or ask the server's owner
Nothing in the main menu	the HUD only shows in a world	use `bc_import` in the console instead
F7 or F9 does nothing, now and then	you were typing somewhere (chat, console, a sign, a map pin, the build menu's search, a number field of the window), or Shift, Ctrl or Alt was held	click an empty part of the window, or close what you typed in, and press the key again
The box and the window vanished	the game's menu is open, or the game's own HUD is hidden (Ctrl+F3)	close the menu, or press Ctrl+F3 again
Another mod reacts to F7 or F9 as well	two mods share a key	change `Window Hotkey` or `Hud Hotkey` (chapter 1)

The preset or the import

What you see	Why	What to do
The export's preset is not in the New World list	it was switched off (**New World preset** unticked, `nopreset`, or a dedicated server), or it could not be made: the export's summary then has a line starting with *WARNING*	look in `BetterContinents/presets/` in the save folder for `<world> <date> <time>.BetterContinents`. If it is not there, run `bc_import` for that export (chapter 3)
bc_import: … holds no map Better Continents loads	the folder has no picture with one of the names the import looks for	point at the export folder itself, and keep the file names (`heightmap.png`, `biomemap.png`, …)
bc_import: no folder or export matches …	a typing error, or the folder is elsewhere	`bc_import list` and a number, or the folder's full path
that export is still being written	the export has not finished	wait for *Done*
this export has no manifest.json, so it did not finish	the game stopped during that export	export again; the import uses what the folder holds
The preset shows the Better Continents logo instead of a map	the folder has neither `heightmap.png` nor `biomemap.png`, which the picture is drawn from	nothing is wrong; the preset works
The config way stopped working after you moved the folder	`From Config` reads the folder from where it was	run `bc_import <new place> config` again, or set `Directory` to the new place (way D, 0.9.4 or later)
Done stays greyed out	the world's name is shorter than 5 characters	type a longer name

World creation seems to hang

What you see	Why	What to do
The progress bar of a new world stands still for a long time	the game is placing locations. A map with a biome left out, or with little room for some locations, makes the game try thousands of spots for each one before it gives up	wait several minutes before you give up. Since Better Continents 0.8.0 a missing biome no longer freezes world creation for ever, as it did before; the log names the missing biomes and the locations the map could not hold

Joining a server takes long

What you see	Why	What to do
A player joining your world waits a minute or more on the loading screen, the first time	the game sends the world's maps at Valheim's own speed, about 150 KB per second	on the server, set `Settings Transfer Rate` (chapter 9), or hand out a `.bcworld` file so players have the maps before they join
`Settings Transfer Rate` changed nothing	it was set on a player's computer: only the server's value counts. Or the connection is crossplay (PlayFab), where the speed cannot be changed: the server's log says *Settings Transfer Rate cannot be raised on this connection*	set it in the server's config; play over Steam
A player who has the `.bcworld` file still downloads the world	the file is for another version of the world's settings (its id differs), or it was never imported: `bc_cache list` on the player's computer does not show its id	export the file again on the server and hand out the new one; on the player's computer, look for *World cache seeded* in the log
... .bcworld: refused (...)	the file is not a Better Continents settings package: cut short in the download, or the wrong file. Better Continents 0.9.x also refuses the file of a version 12 world (the kind 0.10 makes), with *bad header*	download it again; for a version 12 world, use Better Continents 0.10 (a server and its players must run the same version in any case)
bc_cache export: no cached settings for the current world yet	on a player's computer, before the world's maps have arrived; or the world does not use Better Continents	wait until you are in the world, or ask the server's admin for the file

Where to look: the log

Better Continents writes what it does into the game's log, every line starting with `[BetterContinents]`:

- `BepInEx/LogOutput.log`, next to the config folder (in your mod manager's profile, or in the game folder);
- `Player.log` in Valheim's save folder (chapter 2 has the paths).

Useful lines to search for: *World export*, *World import*, *holds a world export* (way D), *was made by a world export for Heightmap Amount* (a heightmap read at other values than it was made for), *Fine heights* (chapter 4), *Compact Maps*, *The biome map has no*, and Valheim's own *Failed to place all*. For joining (chapter 9): *Settings transfer rate*, *Settings sent*, *World cache seeded* and *bc_cache*.

Getting help

Ask on the Better Continents Discord (discord.gg/3XW8ZntYzN) or open an issue at github.com/RGlabs84/BetterContinents10. Say which version you run, what you did step by step, and what you saw, and attach `LogOutput.log` and the export's `manifest.json`.

9 Give your players the world in advance

> **In short:** when a player joins a Better Continents world, the server sends them the world's maps before the game lets them in. For a big world that takes a while. Two things shorten the wait: the server's **Settings Transfer Rate** setting, and a `.bcworld` file you make once with `bc_cache export` and hand out, so a player has the maps before they join. Both are optional: without them everything still works, only slower.

What happens when a player joins

Every player builds the land on their own computer, from the world's settings and pictures. So when someone joins your world for the first time, Better Continents sends them those settings and pictures before the game lets them in: the same maps an export holds, packed for the network. A 2048 px world is about 12 MB; a 4096 px world with a paint map is several times that, and a world of 16384 px pictures some tens to a few hundred megabytes, even compact.

The player's game keeps what it received, in a cache, so the wait is for the first join only, and again only when the world's Better Continents settings change.

Valheim itself limits how fast anything goes through a Steam connection, to about 150 KB per second, no matter how fast the internet on either side is. At that speed 12 MB takes about a minute and a half, and to the player it looks as if the game has frozen on the loading screen.

Faster: the server's Settings Transfer Rate

Whoever runs the world (the host, or the dedicated server) can raise that limit for the time of the transfer. In the server's `BetterContinents.cfg`, section `[07 BetterContinents.Misc]`:

```
Settings Transfer Rate = KB512
```

Value	What it does
`Vanilla`	leaves the game's own speed, about 150 KB per second
`KB256`, `KB384`, `KB512` (the default), `KB768`	sends at 256, 384, 512 or 768 KB per second
`MB1`, `MB1_5`, `MB3`	sends at 1, 1.5 or 3 MB per second
`Unlimited`	sends at up to 100 MB per second; the transfer itself moves about 4 MB per second at most

- **Steam sends at exactly the speed you set.** It does not slow down for a weak connection, so pick a step that your server's upload and your players' downloads can carry. `Unlimited` is for hosted servers with plenty of both.
- **Only the server's value counts.** A player's own `Settings Transfer Rate` does nothing. This is on purpose: the server decides what it sends.
- The game's own speed comes back as soon as the transfer is done. Nothing else is changed.
- It works on Steam connections only. On a crossplay (PlayFab) connection the game's speed cannot be changed, so the server sends at the game's own speed, and its log says so once.
- The setting is read when a player joins, and Better Continents re-reads the file when it changes, so a new value counts for the next player who joins. No restart is needed.

The server's log says what it did: *Settings transfer rate: 512 KB/s (server setting)* when it starts to send, and *Settings sent: 11840771 bytes in 22.8 s (506 KB/s)* when it has finished.

Measured on a dedicated server on the same computer, with a world of 11.8 MB:

Setting	Time to send the world
`Vanilla`	78 s
`KB512`	23 s
`Unlimited`	3 s

> **Good to know:** on a computer that hosts the game for friends, the setting is in your own `BetterContinents.cfg`, the same file as the export settings of chapter 1. Configuration Manager shows it under *07 BetterContinents.Misc*.

> **Good to know:** a world with very large maps is made **compact**: it saves and sends its maps compressed, so the world file and the download are far smaller. `Compact Maps` in `[07 BetterContinents.Misc]` decides it. On **Auto**, the default, a new world is compact when one of its pictures is more than 8192 px across, or when its heightmap has fine heights (chapter 4); **On** makes every new world compact, and **Off** none. One world with five 16384 px pictures came to 52.6 MB compact, against 352.5 MB as pictures; at the default `KB512`, 52.6 MB takes about 1.7 minutes to send (that follows from the rate). Better Continents 0.9 cannot read a compact world.

No wait at all: hand out the world in a file

The maps a player would download can be put in a file, once, and given to the players before they join. A player who has the file joins straight away, every time, until the world's Better Continents settings change. The file is a **world cache file**, and its name ends in `.bcworld`.

If you run the world

1. **Load the world**, the usual way: single player, as the host, or on the dedicated server itself.
2. **Open the console and type `bc_cache export`.** In the game, press F5 first. On a dedicated server, type into its window.

   You see: *bc_cache export: wrote ...\BetterContinents\Midgard\Midgard-3a88cb8d....bcworld (11845213 bytes, id 3a88cb8d...).* and *A description for whoever you give it to is in ...\Midgard-3a88cb8d....txt.*
3. **Find the two files** in Valheim's save folder (chapter 2 says where it is), in `BetterContinents/<world>/`, next to your exports: `<world>-<id>.bcworld`, and a `.txt` beside it that says what the file is, for whoever finds it without you.
4. **Give the `.bcworld` file to your players**: on Discord, a file share, or in a mod pack (below). The `.txt` file can go with it.
5. **When you change the world's Better Continents settings** (another heightmap, another sea level, a new alt-biome map, ...), export again. The new file has a new id in its name; the old file is simply no longer used.

> **Try this:** make a mod pack that holds nothing but the `.bcworld` file, for example "MyWorld-WorldCache" on Hexium or Thunderstore. Every player who installs it has the world before they join, with nothing to type: Better Continents looks for `.bcworld` files in every mod's folder when the game starts.

If you are a player

With a mod pack: install it like any other mod, with Gale or r2modman, or by copying it into `BepInEx/plugins`. That is all. Better Continents finds the file the next time the game starts.

With the file itself:

1. **Install Better Continents** the usual way, if you have not yet.
2. **Start the game.** In the main menu press F5 and type `bc_cache import` followed by the file's path, for example:

   ```
   bc_cache import C:\Users\you\Downloads\Midgard-3a88cb8dac50ed221b369c0dcdad2b33.bcworld
   ```

   Spaces in the path need no quotes. A folder works too: every `.bcworld` in it is imported.

   You see: *Midgard-3a88cb8d….bcworld: imported (id 3a88cb8d…)* and *bc_cache import: 1 imported, 0 already present, 0 refused.* If you had the file already, the line says *already present* instead.
3. **Join the world.** The maps are already on your computer, so the loading screen goes straight on.

> **Good to know:** instead of the console, you can drop the file into `BepInEx/config/BetterContinents/seed/` (make the folder if it is not there). Better Continents imports everything it finds there when the game starts. That is the same thing a mod pack does.

How the file is checked

The file's name does not matter: Better Continents works out the world's id from what is in the file, and uses the file only if it is a genuine settings package. A renamed file works; a broken one is refused with one line, and nothing else happens. An import never replaces what you already have. The version of Better Continents counts too: 0.9.x refuses the file of a version 12 world (the kind 0.10 makes) as a bad header.

`bc_cache list` shows what your cache holds: one line per world, with its id, size and date. The cache itself is `BetterContinents/cache/` in the save folder; a `.bc` file there holds the same data as a `.bcworld`.

A player can make the file too

A player who has joined the world once has its maps in their cache. `bc_cache export` on a player's computer, while in that world, writes them as a `.bcworld` file just as the server would: the same file, byte for byte. Useful when the admin is not around.

10 Edit the ground in Gaea and other heavy editors

In short: export with the **editor pack** (`bc_export 4096 editor`, or tick **Editor pack** in the export window). It adds an `editor/` folder with the ground as a 32-bit float picture (`height.tif`) and masks for water, slope, forest, biomes and locations. Load `height.tif` in Gaea (or another terrain editor), sculpt or erode, save the result in `editor/return/`, then type `bc_import edited` in the console. That makes a new export folder, `<export>-edited`, with your ground in its `heightmap.png`; make the preset from it as for any export (chapter 3). Nothing about an ordinary export changes when you do not ask for the pack.

Chapters 4 and 5 edit the export's pictures in an ordinary image editor. That is enough for painting a coast or moving a mountain. A *heavy editor* does more: it erodes mountains the way water and rock really wear them, carves rivers and valleys, and lets you sculpt at a resolution and with a precision that a 16-bit picture cannot hold. This chapter shows how to hand the world's ground to such an editor and how to bring the result back. **Gaea** is the example, because it is the editor the pack was made for; anything that reads a 32-bit float TIFF follows the same steps.

Good to know: Gaea is a separate program, made by someone else. This guide does not install it, sell it or teach it; it only shows what Better Continents hands to it and what it takes back. Gaea's own documentation explains its nodes, its versions and what each version may do. You do not need a terminal for any of this: "the console" in this guide is Valheim's own command line (press F5 in the game).

When to use which

You want to	Use
paint a coast, an island, a lake bed, or move a mountain	the heightmap in an image editor (chapters 4 and 5)
erode, carve rivers, or sculpt with an editor that keeps decimals	the editor pack, in this chapter
change biomes or locations	their pictures (chapter 4); an editor of heights does not move them (see "What does not follow the ground")

What you need

- **A world to export, and Better Continents with the export HUD on** (chapter 1), or just the console.
- **A terrain editor that loads a 32-bit float TIFF** (or a headerless `.r32`) and saves one back. Gaea does. World Machine, Houdini and others read float pictures too; the steps below name Gaea's windows, and the idea carries over.
- **A few hundred MB of free disk space.** The float picture takes 4 bytes a pixel: 64 MB at 4096 px, 256 MB at 8192 px and 1 GB at 16384 px, for each float file (the TIFF, and the `.r32` if you ask for it too).

Step 1: export with the editor pack

Way 1: the window

1. **Press F7** to open the export window, on its **Export this world** tab.

   You see: the maps, the size, and below the map switches a block called **Editor pack (Gaea and other terrain editors)**.

2. **Tick Editor pack.**

   You see: more choices appear: **Float files** (TIFF and .r32), **Editor amount (or auto)**, **Pack size**, **Also height-metres.tif**, a switch for each mask, and a line that says how much disk the pack takes.

3. **Leave the defaults**, or choose what you need (the table below). TIFF alone, the amount on auto and the default masks are right for a first try. The default masks are water, inland water, slope, forest, biomes, the biome index and the protected locations.

4. **Press Start export** and wait for *Done*, as in chapter 2.

Way 2: the console

Press F5 and type:

```
bc_export 4096 editor
```

That writes the whole export of chapter 2 and the pack with its defaults. More examples:

```
bc_export 4096 editor                          the pack with the defaults: one float TIFF,
  auto amount, the default masks
bc_export 8192 editor eformat=tif,r32 emasks=all eunit=metres
bc_export 4096 editoronly                      heightmap.png and the editor folder, little else
bc_export 2048 editor esize=8192 nobiomes      a 2048 px export whose pack is 8192 px
```

The pack has its own options. They are the same words in the console and in the window, and each applies to one export only (no setting in the config holds them):

Option	Default	What it means
`editor`, `noeditor`	off	write the `editor/` folder, or not
`editoronly`		`editor`, and no biome, location, forest, heat, cold, alt-biome, lava, moss, paint or sources maps and no preset. `heightmap.png`, `export.cfg`, `README.txt` and `manifest.json` stay: the way back needs the heightmap
`eformat=tif`, `eformat=r32`, `eformat=tif,r32`	`tif`	the float files. `tif` is `height.tif`, a 32-bit float TIFF that keeps any value. `r32` is `height.r32`, headerless floats clamped to 0 to 1, the form some editors like. `exr` is refused with a message that it is not written yet
`eunit=v`, `eunit=metres`	`v`	`metres` also writes `height-metres.tif`: the same picture in true metres
`eamount=auto` or `eamount=`N	`auto`	the Heightmap Amount the pack's values are stored for, 0.01 to 81. `auto` is the larger of 2 and the world's own Heightmap Amount
`esize=`N	the export's size	pixels per side of the pack, 128 up to `Largest Size` (16384 at most). A pack above 16384 px is refused: tiles come in a later version
`emasks=default`, `all`, `none`, or a list	`default`	which masks to write, from `water`, `inland`, `slope`, `forest`, `biomes`, `index`, `soft`, `protect`. `default` is all but `soft`. `rivers` is refused: not written yet
`eprotectpad=`M, `eprotectfeather=`M	8, 32	how many metres a protected disc grows past a location's radius, and the width of its soft edge

`etile`, `eregion`, `epitch` and `eexrchannel` are known words that are refused with "comes in a later version". `bc_export help` lists the editor options in two lines, and `bc_export server editor ...` asks a dedicated server to write its own pack, in the server's own save folder (chapter 7).

> **Watch out:** when both `heightmap.png` and the pack are on, the world's ground is sampled twice: the pack has its own pass, at its own size. That makes the export take about as long as the heightmap's pass longer. For a first try that you only want to edit, `editoronly` keeps it to one pass. The window shows a rough estimate of the pack's disk use; the time depends on your computer and your world.

What is in the editor folder

```
export-<date>-<time>/
  (everything an export writes, unchanged)
  editor/
    height.tif            the ground, 32-bit float TIFF
    height.r32            optional: the same, headerless floats clamped to 0 to 1
    height-metres.tif     optional (eunit=metres): the same grid in true metres
    editor.json           how the pixels map to the world, and the numbers of this pack
    README-editor.txt     these steps with this export's own numbers filled in
    probe.tif             a 64 by 64 float test picture that shows how an editor turns a file
    masks/
      water.png                8-bit
      water-inland.png         8-bit
      slope.png                16-bit
      forest-density.png       16-bit
      biome-<Name>.png         8-bit, one for each biome that has ground
      biome-soft-<Name>.png    16-bit (the soft mask; only with emasks=soft or all)
      biome-index.png          8-bit, with biome-index.txt (index: name: R,G,B)
      protect-locations.png    8-bit
    return/               empty: this is where the editor's result goes
```

The game never reads `editor/`, and neither does a `Directory` load (chapter 3), which reads the top of the folder only. The pack never replaces a map of the export: the maps that exist stay where they are, and `editor.json` points at them.

The masks

A mask is a picture with the same grid as `height.tif`. Use it in the editor to keep or change a kind of place.

Mask	Bits	White (or bright) means
`water`	8	the ground is under the game's 30 m water plane
`water-inland`	8	the same, where the biome is not Ocean: lakes, rivers, marsh
`slope`	16	the steepness: the slope in degrees divided by 90, so white is a vertical wall
`forest-density`	16	how dense the forest is: black is bare, white is the densest
`biome-<Name>`	8	the biome of that name; one file for each biome that has ground
`biome-index`	8	the biome's number in each pixel; `biome-index.txt` lists the numbers and colours. A starting point if you want to repaint biomes
`biome-soft-<Name>`	16	how much of the ground's height blend that biome has: the real "where did this biome's height formula act"
`protect-locations`	8	around every location (towns, dungeons, boss places): white within the location's radius plus the pad (8 m by default), fading to black over the feather (32 m by default)

Appendix D has the exact definitions. Three limits: a biome only a few pixels large may have no file of its own (the export's notes say how many pixels; `biome-index.png` still has it); `protect-locations.png` needs the locations, so export on the host or on the dedicated server (a player's own computer knows only the locations the map shows, and the export says so); and the water masks are the game's 30 m

plane, so lakes above the sea are not in them.

The numbers: how the ground is stored

You can skip this part to start. It matters when a number in Gaea looks odd.

The world's height in metres, h, is stored as a *value* v:

```
v      = ( h / 200 + 0.15 - adjustment ) / amount          amount = the pack's Heightmap
  Amount
file   = max( v, 0 )                                      the floor: nothing below 0 is kept
metres = ( file * amount - 0.15 + adjustment ) * 200
```

`adjustment` is 0 at Sea Level 0.5, the default. At Amount 2, which the pack uses unless the world is taller, the file's 0 is -30 m, 1 is +370 m, and the sea (30 m) is 0.15. It is the same arithmetic as `heightmap.png` (appendix D), with these differences:

- **No upper limit.** `height.tif` keeps a value above 1 if the ground is higher than the amount holds. The export's summary says so and says which `eamount` keeps every value inside 0 to 1. (`height.r32` is clamped to 1 either way.)
- **The floor.** Ground deeper than the floor, and the corners of the square far beyond the world, are flattened to 0. `editor.json` counts them (`clippedLow`, `clippedHigh`).
- **The edge drop-off is not undone.** The pack shows the ground as the world has it. Past the world's radius it includes Better Continents' edge drop-off. The way back copies that ring from the export's `heightmap.png` (see "Step 5").
- **The grid.** Pixel i is at `(i / (size - 1) - 0.5) * totalSize`, on both axes: the same grid as every Better Continents picture (appendix D). File row 0 is **north**, and columns run east.

Units. The default, v, is the form above. With `eunit=metres` you also get `height-metres.tif`, true metres, which is easy to read but is not what the way back expects unless you say `unit=metres` (Step 5).

Step 2: load the ground in Gaea

The settings below follow Gaea's documentation; menu names may differ in your version. **None of them has been checked against a real Gaea build yet** (see the box at the end of the chapter). Open `editor/README-editor.txt` in the export folder: it repeats these steps with your export's own numbers. To find the folder, press **Open the folder** in the window's *Last export* part (chapter 2).

1. **Start a new Gaea project.** In its **Terrain Definition**, set the **Height** to the amount times 200 m (400 m at Amount 2), and the **Scale** to `gaea.scaleMetresKeepPixelPitch` from `editor.json`. The plain total size (`gaea.scaleMetresAsTotal`) also works; the two differ by one pixel in `size`. `README-editor.txt` has both numbers.
2. **Add a File node**, set its **Scale** to **Raw**, and load `height.tif` (or `height.r32`).

   You see: the node's minimum and maximum read about the values in `editor.json`'s `height.valueRange`. If it reads 0 and 1 while that range is smaller, the editor normalised the file: switch the node to Mapped or Normalized as `editor.json` says, and remember it for Step 4.
3. **Check the orientation once.** Add a second File node and load `probe.tif`.

   You see: a slope that is high at the top (north), with one small bright patch in the upper left part. If your editor shows it turned or mirrored, do not flip any file by hand: the way back finds the turn out itself (`orient=auto`), or you tell it (`orient=flipy` and the like).
4. **Put a sea at the waterline.** The waterline value is in `editor.json` (`gaea.seaNodeLevel`: 0.15 at Amount 2).

Step 3: edit

Edit as you like. A few uses of the masks:

- **Keep towns, dungeons and boss places where they are.** Use `protect-locations.png` as a mask in the editor, so that erosion and sculpting leave the ground around each location alone.
- **Water.** `water.png` and `water-inland.png` mark the seas and lakes the world has now.
- **Texture and placement.** `slope.png`, `forest-density.png` and the biome masks are for your own texturing; Gaea's masks do not come back into the world (see below).

> **Watch out:** a location is placed where its dot is on `locationmap.png`, whatever you do to the ground. A boss altar under a new mountain is still placed there. Keep locations out of the places you reshape, or move their dots in the location map (chapter 4).

Step 4: save the result in `editor/return/`

1. **Build** (Gaea's build or export step) the height with **Output Range = Raw**, the format **TIFF 32-bit** (or R32), at the **same resolution** as the pack.
2. **Save it as** `height.tif` (or `height.r32`) in the export's `editor/return/` folder. Any other name works if the file is the only one of its kind in that folder, or you give its path with `files=` (Step 5). If several files are there, the way back takes `height.tif`, then `height.tiff`, `height.r32`, `height.raw` and `height.png`, in that order; with none of those names and more than one file, it says so and asks for `files=`.

> **Watch out:** a Normalized or Custom Output Range changes the numbers, and the way back cannot know the range. Build with Raw. If you cannot, say what range the build used when you bring it back: `range=lo,hi` (in metres) or `unit=metres` for a build that wrote metres. The way back stops with a clear message when it sees a result that spans exactly 0 to 1 while the pack did not ("this looks like a Normalized build").

What the way back reads. TIFF (classic or BigTIFF, little or big endian) with **one** grey sample of 8, 16 or 32 bits (unsigned, or 32-bit float), in strips or tiles, **uncompressed or Deflate** (with the predictor 1, 2 or 3). Headerless `.r32` (32-bit floats) and `.raw` or `.r16` (16-bit), and a 16-bit grey `.png`, are read too. LZW, PackBits, JPEG, several samples per pixel and planar TIFFs are refused by name with "save it uncompressed or with Deflate, or as R32". EXR is refused: not read yet. The picture must be square, 16 to 16384 px; it may differ from the export's size. The TIFF's own Orientation tag is ignored (the report says so), and the orientation check handles the rest.

Step 5: bring it back

Type in the console (press F5; this works in the main menu too, with no world loaded):

```
bc_import edited
```

That picks your newest export. Give the export if you want another: its number from `bc_import list`, its folder's name, or a full path (a folder with spaces needs no quotes).

Or use the window: press F7, open the **Import an export** tab, pick the export, and find the new **Editor return** part. It shows what it found in the export's `editor/return/`, and has choices for the orientation and for whether the values are the pack's or metres, tick boxes to pick the Heightmap Amount and the Sea Level automatically (`amount=auto`, `sealevel=auto`) and to *Make the New World preset from the result at once*, and a **Bring it back** button with a progress bar and **Cancel**.

You see: lines like these in the console, the last ones a while later:

```
height.tif: 4096 x 4096 pixels, 32-bit float TIFF, uncompressed, strips.
Orientation: flipy ...
Heights 50 m to 70.5 m; encoded for Heightmap Amount 2 and Sea Level 0.5 (-30 m to 370 m).
Done in 1 s: <folder>
Next: bc_import <world>/<folder>
```

(Your numbers differ.) It makes a **new** folder, `<export>-edited` (or `-edited-2`, and so on), beside the export. The export itself is never changed. Then `bc_import <export>-edited` makes the New World preset (or add `preset` to the first command to do both), and chapter 3 takes it from there.

The options, after `bc_import edited`:

Option	What it does
`unit=v`, `unit=metres`	how to read the numbers. `v` (the default when `editor.json` exists) is the pack's own form; `metres` is for a build that wrote metres
`range=lo,hi`	the build was normalised to a range you know: metres = lo + value x (hi - lo)
`amount=auto` or `amount=`N	the Heightmap Amount the result is stored for. By default the pack's own. `auto` picks the smallest of 2, 3, 4, 5, 6, 8, 10, 16, 32, 64 or 81 that holds the highest edited ground
`sealevel=auto` or `sealevel=`S	the same for the Sea Level. `auto` raises it in steps of 0.05 until the floor holds the lowest ground
`orient=auto`, `none`, `flipy`, `flipx`, `rot180`, `transpose`, `antitranspose`, `cw90`, `ccw90`	how the editor turned the picture. `auto` (the default) finds out
`ering=keep`, `ering=gaea`	the ring beyond the world's radius. `keep` (the default) copies it from the export's own `heightmap.png`; `gaea` works it out from the returned height
`fine=8`, `fine=4`, `fine=off`	the fine file of the result (chapter 4): 8 bits by default
`files=`path	the returned file, when it is not the only one in `editor/return/`. Quote a path with spaces: `files="C:\My Maps\height.tif"`
`size=`N	the side of a headerless file whose length is not a square
`preset`	make the New World preset from the result at once

`bc_import edited help` prints this in the game, `bc_import edited cancel` stops a running one, and `bc_import status` says what it is doing.

How the numbers come across

- **A value outside the range is clipped, counted and said.** If you raised a mountain above what the amount holds, the report says "N pixels below X m and M above Y m were clipped: amount=auto keeps them." Use `amount=auto` (and `sealevel=auto` for very deep ground) to keep them.
- **Precision.** The result is a 16-bit `heightmap.png` and an 8-bit `heightmap-fine.png`, 24 bits in all. At Amount 2 that is within 1 mm of the editor's metres on gentle ground (the test measures about 0.02 mm), and within half a grey step, 3 mm, where the ground is steep or under water, which get no fine bits by the rule of chapter 4.

- **Orientation.** `orient=auto` compares a small copy of the returned picture, turned each of the eight ways, with the pack's own `height.tif` (with the average and the spread taken out, so a rescaled build still compares), and picks the best when the runner-up is at least four times worse. Otherwise it stops and prints the eight scores ("cannot tell how the returned picture is turned … say orient=…"). A flat world, or a picture that is symmetrical or changed all over, cannot be told: then say `orient=`, with `probe.tif` showing which. It needs `editor/height.tif` or `.r32` to compare with; without them the picture is taken as it is, and the report says "not checked".
- **Bad numbers.** A pixel that is not a number is restored from the export, and counted.

What the new folder holds

`heightmap.png` and `heightmap-fine.png` (if any pixel needs it) with your ground; every other top-level file of the export, copied byte for byte (maps, legends, `placements.bcp`); a rewritten `export.cfg` (`Directory` is the new folder, the Heightmap Amount and Sea Level are the result's, every other line as exported); a new `manifest.json` (appendix C) and `README.txt`. The pack and `sources/` stay with the export. Chapter 3's ways B, C and D work on the new folder exactly as on an export.

What does not follow the ground

> **Good to know:** the biomes, alt biomes, locations, forest, lava, moss and paint do **not** move with ground you reshape. A river you carve crosses a Meadows biome that the biome map still calls Meadows; a mountain you raise sits under whatever paint the old ground had. The location map is copied byte for byte, so every dot stays where it was. Repaint what you need in the pictures (start from `biome-index.png`), or edit around `protect-locations.png`. Gaea's own masks do not come back into the world in this version.

Limits, said plainly

Limit	What it means
16384 px	Better Continents reads maps up to 16384 px across. An edit at 1 m over a 21 km world is stored at 1.3 m. Tiles and larger packs are a later project
Half a pixel	The grid puts pixel 0 and the last pixel on the world's edges. Whether Gaea treats a pixel as a point or a cell is not documented, and the way back maps pixel to pixel, so a feature can sit up to half a pixel off: 0.64 m at 16384 px, 2.6 m at 8192 px, 5.1 m at 4096 px, in a 21 km world. `gaea.scaleMetresKeepPixelPitch` keeps the pitch the same as Better Continents'.
Disk	A float container is 1 GiB at 16384 px. The default is one container
Time and memory	The pack has its own terrain pass, about as long as the heightmap's. Reading a 16384 px return took 11 s and about 55 MB more memory on an offline test world, measured; a real world has not been measured
The result's precision	24 bits: about a millimetre at Amount 2 on gentle ground, 3 mm where steep or under water
Lakes above the sea	not in the pack's masks; the water masks are the game's 30 m plane

> **Good to know:** the tests of the pack and the way back use Better Continents' own writers and readers on an offline stand-in for the world, including TIFFs with strips, tiles, Deflate and both byte orders, and pictures of another size than the export. They show that what the pack writes, the way back reads. They cannot show how Gaea itself behaves.

What we have not yet checked with a real Gaea build

> **Good to know:** this chapter says plainly what is settled and what is not. The pack and the way back are tested with Better Continents' own code. They have **not** yet been run through a real Gaea. Until they have, the steps for Gaea follow Gaea's documentation, and the open points are in the table below. Each one shows itself in a minute if you look.

Not checked yet	How you see it, and what covers you meanwhile
Whether Gaea shows the TIFF with north at the top	Load `probe.tif` (Step 2, point 3). `orient=auto` finds the turn either way
Whether Gaea counts a pixel as a point or a cell	Up to half a pixel, see the limits. Nothing to do
Whether the File node in Raw keeps values above 1 or below 0 from a float TIFF	The File node's minimum and maximum (Step 2, point 2). The pack keeps `v` near 0 to 1 at Amount 2, and below 0 is floored in the file
Whether Gaea's "Height = amount x 200 m" is what its 1.0 means	It only matters to Gaea's own absolute-height nodes. Gaea's documentation never states it
Which compression Gaea's own TIFF writer uses	The way back reads uncompressed and Deflate files and names anything else; the report's first line says what the file was. LZW is refused by name: save as R32 or uncompressed instead
Whether Gaea imports the masks' PNGs	Try a mask in a File node. Gaea's newer documentation lists 8 and 16-bit PNG as importable; its older import list does not
Gaea's EXR and tile numbering	Not read or written yet
Time and memory on a real world	Only an offline stand-in was measured

A short test with a real Gaea, a few minutes, that settles most of these. In a world, type `bc_export 512 editoronly` (a small export, fast). Load its `editor/probe.tif` and `editor/height.tif` in Gaea as in Step 2 and note: where north is in `probe.tif`, what the File node reads as its minimum and maximum, and whether the node keeps values outside 0 to 1. Then build the height at 512 px as in Step 4 (Raw, TIFF 32-bit), save it as `editor/return/height.tif`, and type `bc_import edited`. The first lines of the report say what kind of file Gaea wrote and how it was turned. Save a copy of that file and the report: they show what to fix, if anything.

Where to go next

- Chapter 3 makes a world from the `-edited` folder, and chapter 8 lists what can go wrong, including the messages of the way back.
- Appendix A lists the options; appendix C the new manifest keys; appendix D the formulas.

Appendix A

The console commands

Open the console with F5. Commands and options are not case-sensitive, and a decimal number takes a point: `amount=2.5`.

bc_export

```
bc_export [status | cancel | server <options> | <options>]
  options: [size] [amount=1] [sealevel=0.5] [heatscale=10]
           [edge=auto|on|off] [forest=additive|exact] [fine=auto|off|4|8]
           [noheight] [nobiomes] [nolocations] [noforest] [noheat] [nocold]
           [noaltbiomes] [nolava] [nomoss] [nopaint] [nosources]
           [nopreset] [perpoint]
  editor pack (chapter 10):
           [editor | editoronly] [eformat=tif,r32] [eunit=v|metres]
           [eamount=auto|<n>] [esize=<n>] [emasks=default|all|none|<list>]
           [eprotectpad=<m>] [eprotectfeather=<m>]
```

(`bc_export help` prints the same on one line.)

For every player, in a world, without cheats. On a server, a player may export only while the server's `Allow Export` is on; the machine that runs the world always may.

Command	What it does
`bc_export` *options*	starts an export with the defaults of `[09 BetterContinents.Export]`, changed by the options
`bc_export status`	what the export is doing, the folder of the last export, the last error, and why an export cannot start now
`bc_export cancel`	stops the running export and deletes the files it wrote, its preset included
`bc_export help`	the usage line above, and the defaults in use
`bc_export server` *options*	asks a dedicated server to export its own world (admins only). The server writes into its own save folder and answers in its log and to you, the end of the export included
`bc_export server status`, `bc_export server cancel`	the same, on the server

Option	Default	What it means
a number, or `size=`N	`Default Size`, 4096	pixels per side of every picture, 128 up to `Largest Size` (16384 unless the config says less). Above 8192 the export makes no preset: `bc_import` makes it (chapter 3)
`amount=`A	`Default Heightmap Amount`, 1	the Heightmap Amount the heights are stored for, above 0 and at most 81. On the game's height scale, where the sea is at 30 m, 1 spans -30 m to 170 m (60 m under the sea to 140 m above it), 2 spans -30 m to 370 m (up to 340 m above it) and 81 spans -30 m to 16,170 m (chapter 4)
`sealevel=`S	`Default Sea Level`, 0.5	the Sea Level Adjustment the heights are stored for, 0 to 1
`heatscale=`H	10	the Heatmap Scale the heat is stored for, above 0 and at most 100

Option	Default	What it means
`edge=auto`, `edge=on`, `edge=off`	auto	the world's edge. *auto*: as the world has it. *on*: the pictures hold the ground before Better Continents' edge drop-off, which then rebuilds the edge; the world keeps its edge. *off*: the edge is stored as it is and `Map Edge Drop-off` is switched off, which also removes the world's edge: no push back, and the sea goes on for ever
`forest=additive`, `forest=exact`	additive	how the forest is stored: *additive*, black is the game's own forest; *exact*, black is no forest
`fine=auto`, `fine=off`, `fine=4`, `fine=8`	auto	whether the export writes `heightmap-fine.png`, finer heights beside the heightmap. *auto*: with 4 bits from a Heightmap Amount of about 5 (a step of grey above 1.5 cm), none below. *off*: never. *4*, *8*: at any Amount, with 4 or 8 bits (chapter 4)
`noheight` (or `noheights`, `noheightmap`)		no `heightmap.png`
`nobiomes` (or `nobiome`)		no `biomemap.png`
`nolocations` (or `nolocation`)		no `locationmap.png`
`noforest`		no `forestmap.png`
`noheat`		no `heatmap.png`
`nocold`, `cold`	with the cold map	no `coldmap.png`; or with it (`cold` is accepted so that a line can say it)
`noaltbiomes` (or `noaltbiome`)		no `altbiomemap.png`
`nolava`, `nomoss`		no `lavamap.png`, no `mossmap.png`
`nopaint`, `paint`	with paint	no `paintmap.png`; or with it
`nosources`		no `sources/`
`nopreset`, `preset`	with a preset; none on a dedicated server	no New World preset; or make one
`perpoint`		heights one point at a time instead of the way the game builds terrain; for testing only
`editor`, `noeditor`	off	write the `editor/` folder, the ground as 32-bit floats and masks for Gaea and other terrain editors (chapter 10)
`editoronly`		`editor`, and no biome, location, forest, heat, cold, alt-biome, lava, moss, paint or sources maps and no preset; `heightmap.png`, `export.cfg`, `README.txt`, `manifest.json` and `world.json` stay
`eformat`=list	`tif`	the float files of the pack, `tif`, `r32` or `tif,r32`: `height.tif` (32-bit float TIFF), `height.r32` (headerless floats clamped to 0 to 1). `exr` is refused: not written yet
`eunit`=unit	`v`	`v` or `metres`; `metres` also writes `height-metres.tif`, the same grid in true metres
`eamount`=A	`auto`	the Heightmap Amount of the pack's values, 0.01 to 81, or `auto`: the larger of 2 and the world's own
`esize`=N	the export's size	pixels per side of the pack, 128 up to `Largest Size`; above 16384 is refused
`emasks`=list	`default`	the masks to write: `default`, `all`, `none`, or a list from `water`, `inland`, `slope`, `forest`, `biomes`, `index`, `soft`, `protect`, comma between. `default` is all but `soft`; `rivers` is refused: not written yet
`eprotectpad`=M, `eprotectfeather`=M	8, 32	metres a protected disc grows past a location's radius, and the width of its soft edge

Option	Default	What it means
`etile`, `eregion`, `epitch`, `eexrchannel`		known, and refused: "comes in a later version"

What goes wrong is said on the spot: *bc_export: unknown option '...'*, *bc_export: cannot read '...'*, or *World export cannot start: ...* with the reason, for example *the size must be 128 to 16384 pixels (got 100)*.

bc_import

```
bc_import [list | status | help | edited | <number> | <folder> | <world>] [config]
```

For every player, in the main menu and in a world. It reads folders on your own computer and writes your own presets and config, so no server has a say.

Command	What it does
`bc_import`	makes a preset from your newest finished export and selects it: the export of the world you are in; else the last export you made since the game started; else the newest export of the world you played last; else the newest of any world
`bc_import list`	numbers the export folders, newest first, with world, time and size; *(unfinished)* marks an export that did not finish, *(preset made)* one that has its preset
`bc_import` *number*	the export on that line of the list
`bc_import` *world*	that world's newest finished export
`bc_import` *folder*	a full path, or a path under `BetterContinents/` such as `Midgard/export-2026-09-24-14-05-33`. Any folder of Better Continents maps works, with or without `export.cfg`; spaces need no quotes
`bc_import` *export folder name*	for example `export-2026-09-24-14-05-33`
... `config`	added at the end: loads the folder through `BetterContinents.cfg` and selects *From Config*, instead of making a preset
`bc_import status`	what the import is doing, and how the last one went
`bc_import help`	the list above
`bc_import edited` ...	brings back the ground edited in Gaea or another terrain editor: the next part

What an import writes:

- **The preset way:** in `BetterContinents/presets/`, the preset `<name>.BetterContinents`, its 256 px picture `<name>.png`, and `<name>.import.txt`, a note of the folder it was made from. Nothing in your config changes except which preset is selected. A folder without `export.cfg` takes every setting but the folder from your own config.
- **The config way:** `BetterContinents.cfg` gets the world settings of `export.cfg`, `Directory` set to the folder, `Enabled = true` and `SelectedPreset = From Config`. The Export section, `Settings Transfer Rate`, `Map Memory`, `Max Map Size`, `Wide Sectors`, `Debug Mode`, `Debug Reset Command`, `NexusID` and `Config Version` are never copied. Your old file is kept beside it as `BetterContinents.cfg.before-import-<time>`.

bc_import edited

```
bc_import edited [<number> | <folder> | <world>] [unit=v|metres] [range=lo,hi]
  [amount=auto|<n>] [sealevel=auto|<s>]
                [orient=auto|none|flipy|flipx|rot180|transpose|antitranspose|cw90|ccw90]
  [ering=keep|gaea]
                [fine=8|4|off] [files=<path>] [size=<n>] [preset]
bc_import edited help | bc_import edited cancel
```

For every player, in the main menu and in a world (no world needs to be loaded). It reads the ground your editor saved in the export's `editor/return/` and writes a **new** folder, `<export>-edited` (`-edited-2`

and so on) beside the export, which is never changed. Chapter 10 explains every step and option; the table has them in one place.

Word	What it means
the export	found as `bc_import` finds one: a number from `bc_import list`, a folder, a world, or nothing for the newest. A folder with spaces needs no quotes
`unit=v`, `unit=metres`	how to read the returned numbers: the pack's value (the default when `editor.json` is there) or metres
`range=lo,hi`	the build was normalised to a range, in metres: metres = lo + value x (hi - lo)
`amount=auto`, `amount=`A	the Heightmap Amount the result is stored for: the pack's own by default; `auto` picks the smallest of 2, 3, 4, 5, 6, 8, 10, 16, 32, 64 or 81 that holds the highest ground (above 0, at most 81)
`sealevel=auto`, `sealevel=`S	the same for the Sea Level, 0 to 1: `auto` raises it in steps of 0.05 until the floor holds the lowest ground
`orient=`	how the editor turned the picture: `auto` (the default) finds it out; or `none`, `flipy`, `flipx`, `rot180`, `transpose`, `antitranspose`, `cw90`, `ccw90`
`ering=keep`, `ering=gaea`	the ring past the world's radius: `keep` (the default) copies it from the export's `heightmap.png`; `gaea` works it out from the returned height
`fine=8`, `fine=4`, `fine=off`	the result's `heightmap-fine.png`: 8 bits by default
`files=`path	the returned file, when it is not the only one in `editor/return/` (quote a path with spaces)
`size=`N	the side of a headerless `.r32` or `.raw` file that is not a square number of values
`preset`	make the New World preset from the result at once
`bc_import edited cancel`	stops a running one and deletes the new folder

It reads TIFF (classic or BigTIFF, one grey sample of 8, 16 or 32 bits, uncompressed or Deflate), headerless `.r32` and `.raw`, and 16-bit grey PNG, square, 16 to 16384 px; it refuses EXR and the TIFF compressions it cannot read, by name. `bc_import status` says how it is going. The messages are in chapter 8.

bc_cache

```
bc_cache [export | import <file or folder> | list | help]
```

For every player, in a world and in the main menu (`import`, `list`). Not a cheat. Chapter 9 explains what the file is for.

Command	What it does
`bc_cache export`	in a world: writes its maps as `<world>-<id>.bcworld`, with a `.txt` note beside it, into `BetterContinents/<world>/` in the save folder. On the machine that runs the world, from the world's own settings; on a player's computer, from the copy in your cache, once it has arrived
`bc_cache import` *file*	puts one `.bcworld` file into your cache. A path with spaces needs no quotes
`bc_cache import` *folder*	every `.bcworld` in that folder and its subfolders
`bc_cache list`	what your cache holds: the id, size and date of every world
`bc_cache help`	the list above, and the seed folder's path

An id you already have is skipped, never replaced, and the id is always worked out from the file's content, never taken from its name. A file that is not a settings package is refused with one line. Better Continents also imports every `.bcworld` it finds under `BepInEx/plugins/` or in `BepInEx/config/BetterContinents/seed/` when the game starts.

bc_twins

```
bc_twins [radius] | remove [radius] | remove confirm [radius] | server <the same> | help
```

For every player, in a world, without cheats. It finds and removes vegetation twins: two copies of one plant on one spot (or within the radius, 1 m by default), which a zone filled with vegetation a second time over its own objects leaves behind. Better Continents keeps new twins out of every world as zones generate; this command is for the ones a world already has. Removal runs where the world is: single player, a host, or a dedicated server through `bc_twins server ...` (admins only; the answer comes back to the admin's console).

Command	What it does
`bc_twins`	counts the twins by plant and zone; a player counts only the area around them, `bc_twins server` 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 a radius is given
`bc_twins remove confirm` [*radius*]	deletes the extra copies, 1000 a frame. The copy nearest the ground stays; nothing a player built is ever deleted; crops players grow go only when two stand on one spot. Back up the world first
`bc_twins server` ...	the same on a dedicated server (admins only)
`bc_twins help`	the usage

The bc commands (host, cheats)

`bc` is Better Continents' command for building worlds. It needs cheats (type `devcommands` first) and is registered for the machine that runs the world only. With the Server Devcommands mod installed, any player who has cheats on can type it on their own computer, but a change made there reaches only that player's own copy of the settings and their own view of the ground, never the world or the other players, and `bc regen` refuses. Make `bc` changes on the host or in single player. `bc info` on a player's computer shows the settings the server sent.

Command	What it does
`bc export run` *options*	the same as `bc_export` *options*
`bc export status`, `bc export cancel`	the same as `bc_export status`, `bc_export cancel`
`bc import run` *what* [`config`]	the same as `bc_import` *what* [`config`]
`bc import list`, `bc import status`	the same as `bc_import list`, `bc_import status`
`bc b p` [0 to 31]	the Biome precision of the loaded world (chapter 6; an older world (settings version 11) stops at 5); with no number, shows it
`bc savepreset` [*name*]	saves the loaded world's settings as a preset, with a picture of its map; the name is the world's unless you give one
`bc locs`	how many of each location the world has
`bc bosses`	pins the bosses, the start temple and the traders on the map
`bc cache` ..., `bc twins` ...	the same as `bc_cache` and `bc_twins`

Command	What it does
`bc regen` [*words*]	regenerates the world's zones now (in a world that uses Better Continents), as every change does: every generated zone is reset and generates again when somebody comes near, except the zones within one zone of something a player built, a tombstone, ground a player has worked (fields, paths, roads), or you inside a dungeon, and the zones another player's game has loaded around them. A location that reaches into more than one zone is kept or regenerated whole. Players, what players built, tombstones and tamed animals are never removed. It rebuilds no ground: nothing has changed. Words after it reset only part of the world (`dry` to count first, `biome=`, `min=`, `max=`, `pos=`, `zone=`, `chance=`); the Better Continents Guide explains them
`bc reset`	redraws the minimap (and reveals all of it), rebuilds the alt-biome grid and the loaded ground, then regenerates the zones, as every change does
`bc export`, `bc import`	alone: lists their commands
`bc` *group* `help`	lists a group, for example `bc b help`

> **Watch out:** a change made with `bc` rebuilds the loaded ground first (the game stalls for a second or two) and then regenerates the zones; `bc regen` only regenerates the zones. The ground of every zone, the zones that stay included, follows a change: one 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; one that raises it buries them. Make big height changes before you build, or away from what you want to keep.

Appendix B

export.cfg, key by key

`export.cfg` holds the Better Continents settings that rebuild the exported world from the pictures in its folder. It has the same layout as `BetterContinents.cfg`: `[sections]`, and `key = value` lines, each with a `##` comment saying why it has that value. A few keys are only written when their map was exported.

Ways B, C and D of chapter 3 use it: `bc_import` lays it over a copy of your own config to build the preset, and `bc_import ... config` copies it into your config. Since Better Continents 0.9.4, a new world made with **From Config** while `Directory` points at the folder lays it over your config as well, without changing either file (way D); its settings win over `BetterContinents.cfg`. **Every setting it does not list comes from your own** `BetterContinents.cfg`, as it is at that moment. To use a value of your own for a setting it lists, change it here.

The file starts with a short explanation for whoever opens it first:

```
## Better Continents world export of "Midgard" (seed "aB3dE5fG7h"), 2026-09-24 14:05.
##
## WHAT THIS FILE IS: the Better Continents settings that rebuild this world from the maps in
  this folder.
## You do not have to edit it. Three ways to make a new world from the export (README.txt has
  the steps):
##  1. Easiest: in the New World screen, pick the Better Continents preset "Midgard 2026-09-24
  14-05-33" and create the world.
##  2. After editing the PNGs: open the console (F5) and type   bc_import
  Midgard/export-2026-09-24-14-05-33
##     It makes the preset again from the PNGs as they are now, and selects it.
##  3. To keep editing the PNGs: type   bc_import Midgard/export-2026-09-24-14-05-33 config
##     or paste every line of this file at the END of BepInEx/config/BetterContinents.cfg (a
  later line wins) and
##     save it. Better Continents reads the change at once; create the world with the preset
  "From Config".
##     Or set only Directory to this folder: a new world made "From Config" then reads this
  file too, and its
##     settings win over BetterContinents.cfg. The maps are only right with them (Heightmap
  Amount above all).
## Only NEW worlds use these settings. A world that exists keeps the maps it was created with.
```

Its way 3 is chapter 3's ways C and D. In the tables, *old world* means the world that was exported.

[00 BetterContinents.Debug]

Key	Value written	Why	Change it?
`Enabled`	true	Better Continents on for the new world	no
`Directory`	the export folder, with `/` between folder names	loads every map in the folder by its standard name, and no map from anywhere else	no: the import always sets it to the folder it imports, and way D ignores this line and keeps the `Directory` of `BetterContinents.cfg`, so a moved or copied folder still works
`Override version`	empty	the new world gets the settings version its pictures were made for: the newest, or 11 for an export of a world resized under 0.9.x (see below)	no

> **Good to know:** a world made from an export gets the newest *settings version*, 12, when the pictures cover what the world's own size gives, 2 × (`World Size` + `Edge Size`) metres: that is `totalSize` in `manifest.json` (appendix C). Every export of a world of the normal size does, a 0.9.x export included, and so does every export of a version 12 world. An export of a world that was resized under 0.9.x, or made under Expand World Size at another size, covers another span and makes a version 11 world, which gets none of the new behaviour of Better Continents 0.10.
>
> `Override version` is for testing. Set to `11` in your own `BetterContinents.cfg`, it makes a world made with *From Config* (ways C and D) as Better Continents 0.9.4 would have made it. The preset that ways A and B use is built without reading it, but a world made from a preset is saved in it like every other world: while it is set, every world is saved in that version. This line is empty in every `export.cfg`, so way C, which copies the settings of `export.cfg` into your config (pasting the file by hand does the same), sets an `Override version` you had set back to empty. Leave it empty otherwise.

[01 BetterContinents.Global]

Key	Value written	Why	Change it?
`World Size`	the old world's, 10000 normally	the playable radius, in metres. In a version 12 world the pictures span World Size + Edge Size on each side of the middle	no: the pictures are made for it
`Edge Size`	the old world's, 500 normally	the width of the edge ring, in metres	no
`Map Edge Drop-off`	as the old world (true for a vanilla world); false after `edge=off`, true after `edge=on`	whether Better Continents drops the land into the sea at the world's edge	no
`Skip Default Locations`	true when the location map holds every location, else false	true: the game places no locations but the map's	false, to let the game add its own locations around yours
`Sea Level Adjustment`	0.5	the sea level the heights are stored for (only with a heightmap)	never, unless you change every pixel of the heightmap
`Rivers`	false	the old world's rivers are already carved into the heightmap (only with a heightmap)	no
`Mountains Allowed At Center`	true	the old world's middle is already in the heightmap: no second flattening (only with a heightmap)	no
`Mountain Noise`	0	the old world's mountain jaggedness is already in the heightmap, so the new world adds no second layer of it, whatever your own `BetterContinents.cfg` says (only with a heightmap)	no
`Mistlands Noise`	0	the same for the Mistlands crags (only with a heightmap)	no
`Ashlands Water`, `Deep North Water`	the old world's	whether the Ashlands' red sea and the Deep North's calm sea follow the world's own maps, so the new world floats and colours its seas the same way (only from a Better Continents world)	no

[02 BetterContinents.Heightmap]

Only written when the heightmap was exported, except `High Terrain`.

Key	Value written	Why	Change it?
`Heightmap Amount`	1, or the `amount=` of the export (2, for example, up to 81)	the amount the heights are stored for; `heightmap.png` records it too	never, unless you change every pixel of the heightmap
`Heightmap Blend`	1	the heightmap is the height, not mixed with generated ground	no
`Heightmap Add`	0	nothing added on top	no
`Heightmap Mask`	0	no masking	no
`Heightmap Override All`	true	every biome takes its height from the heightmap	no
`Heightmap Alpha`	false	`heightmap.png` is plain 16-bit grey	no
`High Terrain`	On or Off, as the old world; written when the old world's is not Auto, and also when it is Auto but would not get the same state from Auto at the export's Amount (its own Heightmap Amount and the export's sit on different sides of 5)	which worlds get Better Continents' changes to the game's rules for high ground. Without this line the new world takes your config's, Auto by default: the changes for a heightmap read at an Amount above 5	no

`heightmap-fine.png`, when the export wrote one, needs no line here: Better Continents finds it beside `heightmap.png` (chapter 4).

[03 BetterContinents.Biomemap]

Key	Value written	Why	Change it?
`Biome precision`	the old world's, 0 to 31	how closely the ground follows the biome map inside each 64 m zone	yes, if you like (chapter 6)

[04 BetterContinents.Forest]

Key	Value written	Why	Change it?
`Forest Scale`	the old world's; 0.5 for a vanilla world	the size of the game's own forest patches, so that its forest pattern lines up with `forestmap.png`	no
`Forest Amount`	0.5 with a forest map; else the old world's	0.5 is neutral: the old world's amount is already in `forestmap.png`	no
`Forestmap Multiply`	0, or 1 after `forest=exact`	0: forest = the game's own + the map; 1: forest = the map × (the game's own + 1)	no
`Forestmap Add`	1	see Forestmap Multiply	no
`Forest Factor Overrides All Trees`	the old world's	as the old world	no

[05 BetterContinents.StartPosition]

Key	Value written	Why
`Override Start Position`	false	the start temple is in the location map

Without a location map, a Better Continents world that fixed its start position keeps it: `Override Start Position = true` with its `Start Position X` and `Start Position Y`.

[06 BetterContinents.Maps]

Each key is written only when its map was.

Key	Value written	Why	Change it?
`Heatmap Scale`	10	heat = pixel / 65535 × scale; Ashlands wherever it is above 0 (with the heat map)	no
`Coldmap Scale`	10, or the old world's own scale when its own cold map was copied	cold = pixel / 65535 × scale; Deep North wherever it is above 0 (with the cold map)	no

[07 BetterContinents.Misc]

Key	Value written	Why
`SelectedPreset`	From Config	selects *From Config* when you paste the file into your config by hand. The import and way D ignore it: `bc_import` selects the preset it made, `bc_import ... config` selects *From Config* itself, and with way D you pick *From Config* yourself

`Compact Maps` is not written: the new world takes your config's, Auto by default, which makes a world of very large pictures or with fine heights compact (chapter 9). `Wide Sectors` is never written either: it belongs to the computer that runs the world, and a `Wide Sectors` line added here is ignored.

[08 BetterContinents.AltBiomes]

Only written when the alt-biome map was exported.

Key	Value written	Why	Change it?
`Mode`	PlantedOnly	the alt-biome map holds every alt biome as the game placed it, so the game adds none at random	Random, to let the game add alt biomes to the regions left black
`Grid`	the old world's	where alt biomes stop at the world's edge	no

What the import never takes from export.cfg

The Export section (`[09 BetterContinents.Export]`), `Settings Transfer Rate`, `Map Memory`, `Max Map Size` and `Wide Sectors` (settings of the machine), `Debug Mode`, `Debug Reset Command`, `NexusID` and `Config Version` are not world settings, so the import skips them and says so. Lines it cannot read, and names Better Continents 0.10.x does not know, are skipped and reported the same way: *World import: ignored ….* Way D skips the same lines, and the log lists them as *export.cfg: ignored ….*

Appendix C

manifest.json, field by field

`manifest.json` is written last, so a folder that has one holds a finished export. It is plain JSON, for tools and for checking that two exports line up (chapter 5). A field is `null` when its map was not exported.

The first lines of a manifest look like this:

```
{
  "format": "bc-export/1",
  "modVersion": "0.10.x",
  "gameVersion": "l-1.0.x",
  "exportedAt": "2026-09-24T14:05:33",
  "seconds": 184.2,
  "worldName": "Midgard",
  "seedName": "aB3dE5fG7h",
  ...
```

The export

Field	Example	What it is
`format`	`"bc-export/1"`	the layout of the export; it changes only if the way pictures are stored changes
`modVersion`	`"0.10.x"`	the Better Continents version that exported
`gameVersion`	`"l-1.0.x"`	Valheim's version text
`exportedAt`	`"2026-09-24T14:05:33"`	when the export started
`settingsVersion`	`12`	the settings version a world made from this folder gets: 12, or 11 when the pictures span other than World Size + Edge Size give (the rule `bc_import` uses, chapter 3)
`exportRevision`	`2`	2 for an export with `coldmap.png`, the exclusion mask, `world.json` and `settingsVersion`; a manifest without this key is an older export. A tool that needs one of them tests this, not `format`
`seconds`	`184.2`	how long it took
`worldName`, `seedName`, `seed`	`"Midgard"`, `"aB3dE5fG7h"`, a number	the old world, its seed as typed, and the seed as a number
`betterContinentsWorld`	`false`	whether the old world was a Better Continents world
`sampledOn`	`"host"`	`"host"`, `"client"` (a player on a server: incomplete locations) or `"dedicated server"`

Size and place

Field	Example	What it is
`size`	`4096`	pixels per side of every picture, 128 to 16384
`totalSize`	`21000`	metres the pictures cover per side: 2 × (World Size + Edge Size) for a version 12 world; 21000 for an older world (settings version 11), whatever its World Size said; with Expand World Size installed, the size that mod gives the world, for either
`worldRadius`, `totalRadius`	`10000`, `10500`	the playable radius, and the radius to the outside of the edge ring
`worldSize`, `edgeSize`	`10000`, `500`	the world's World Size and Edge Size settings
`layout`	`"scaled"`	`"scaled"` for a settings version 12 export, `"vanilla"` for a version 11 one
`wideSectors`	`false`	whether the exported world used Wide Sectors
`highTerrain`	`0`	the exported world's High Terrain mode: 0 Auto, 1 On, 2 Off
`metresPerPixel`	`5.1282053`	totalSize / (size − 1)
`pixelToWorld`	text	the pixel-to-world formula, as in appendix D

Heights

Field	Example	What it is
`heightmapAmount`, `seaLevel`	`1`, `0.5`	what the heights are stored for (1 unless the export was made with `amount=`); `heightmap.png` records them too
`seaLevelAdjustment`	`0`	1 − 2 × seaLevel
`heightEncoding`	text	the height formula, as in appendix D
`fineBits`	`4`	only when the export wrote `heightmap-fine.png`: how many bits its bytes use, at most 4 with auto or `fine=4` and at most 8 with `fine=8` (chapter 4)
`fineEncoding`	`"signed offset, 1/256 step, format 1"`	only with `heightmap-fine.png`: how its bytes refine `heightmap.png`, as in appendix D
`waterlineValue`	`0.3`	the value at the sea's surface (`0.15` at Amount 2)
`floorMetres`, `ceilingMetres`	`-30`, `170`	the heights of black and white, counted the game's way (the sea is at 30 m); `370` at Amount 2
`clippedLow`, `clippedHigh`	`0`, `18452`	pixels inside the world that were lower than black or higher than white
`edgeRingClipped`	`0`	the same for the edge ring, where the land falls away anyway
`heightRangeMetres`	`[-24.3, 318.9]`	the lowest and highest ground inside the world
`zoneBlend`	`true`	heights sampled the way the game builds its terrain
`edgeDropoff`	`true`	whether the new world keeps Better Continents' drop-off at the world's edge

The other maps

Field	What it holds
`forest`	`encoding` (`"additive"` or `"exact"`), `multiply`, `add`, `forestScale`, and `clippedLow`, `clippedHigh`: pixels whose forest the encoding could not hold
`heat`	`scale` (10), `ashlandsPixels` (pixels above black), `clipped` (pixels hotter than the scale)
`cold`	`scale` (10, or the world's own), `deepNorthPixels` (pixels above black; null when the world's own map was copied) and `source`: `"sampled"` or `"the world's own map"`. The whole field is null when there is no cold map
`altBiomes`	`classes` (colours), `regions`, `forcedClasses` (colours with a forced `!` name), `droppedRegions` (regions past the limit of colours, left out)
`locations`	`complete` (every location of the world is in the map), `instances`, `types`, `written`, `nudged` (moved a pixel or two to keep apart from a neighbour), `dropped`, `outsideMap`, the location map's own `pixelToWorld` formula, and `pinGrid` (`"Corner"`: the dots sit on the i / size grid of appendix D)
`biomePixels`	how many pixels each biome has, such as `"Meadows": 1520441`

The preset, the files and the notes

Field	Example	What it is
`preset`	an object	`name`, `file` (the preset's path), `made` (true or false), `error` (why not, or null), `howToUse`. Above 8192 px `made` is false and `error` says that `bc_import` makes it
`import`	`"bc_import Midgard/export-2026-09-24-14-05-33"`	the console line that makes the preset again
`files`	a list	every file the export wrote, `manifest.json` included; with the editor pack, every `editor/...` file too
`fileBytes`	an object	the size in bytes of each top-level file that existed when the manifest was built: so not `export.cfg`, `world.json`, `README.txt` or `manifest.json` themselves
`editorPack`	an object	only when the editor pack was written (chapter 10): `dir` (`"editor"`), `json` (`"editor/editor.json"`), `format` (`"bc-editor/1"`), `formats` (the float files), `size`, and `tiled` (false). An export without the pack has none of this
`notes`	a list	the same notes as `README.txt`'s *NOTES ON THIS EXPORT*

A manifest that `bc_import edited` writes

The `-edited` folder that `bc_import edited` makes (chapter 10) has a manifest of the same shape. These keys are made new: `modVersion`, `exportedAt` (now), `heightmapAmount`, `seaLevel`, `seaLevelAdjustment`, `heightEncoding`, `waterlineValue`, `floorMetres`, `ceilingMetres`, `clippedLow`, `clippedHigh` and `heightRangeMetres`; `fineBits` and `fineEncoding` follow the fine file; `heightmapSize` appears when the returned size differs from `size`; `preset` is null; `editorPack` and `seconds` are dropped; `import` names the new folder; `notes` gains a line; `files` is the new folder's list. It has one new key, `editedFrom`: `export` (the folder it came from), `returnFile`, `returnSize`, `orientation`, `orientationFound`, `unit`, `range`, `edgeRing`, `editorJson`, `clippedLow`, `clippedHigh`, `restoredPixels` and `heightRangeMetres`.

editor/editor.json

The editor pack's own description, format `bc-editor/1`, for tools. Its keys: `format`, `exportFormat` (`"bc-export/1"`), `modVersion`, `exportedAt`, `worldName`, `seed`, `source` (the manifest, the heightmap, its amount and sea level), `grid` (`size`, `totalSize`, `metresPerPixel`, `registration`, `origin`, `axes`, `worldRadius`, `totalRadius`, `edgeDropoff`, and the formulas `worldX` and `worldZ`), `height` (`files`, `dtype`, `unit`, `amount`, `seaLevel`, `seaLevelAdjustment`, `formula`, `floorMetres`, `ceilingMetres`, `waterlineValue`, `valueRange`, `heightRangeMetres`, `clippedLow`, `clippedHigh`, `clamped01`, `edgeDropoffApplied`, `metresFile`, and `crc32`, the CRC-32 of the first float file as 8 hex digits), `gaea` (`note`, `terrainHeightMetres`, `scaleMetresKeepPixelPitch`, `scaleMetresAsTotal`, `seaNodeLevel`, `buildOutputRange`, `buildFormat`, `buildResolution`; none of these settings has been checked against Gaea itself), `masks` (a list of `file`, `bits` and `definition`), `referencedMaps` (the export's own maps, not copied), `protect` (`discs`, `padMetres`, `featherMetres`), `probe`, `return`, and `tiles` and `window`, which are null and kept for a later version.

Appendix D

The formulas, for the curious

Nobody needs these to export, edit and import. They are here for tools, scripts, and the curious. `size` is the pictures' size in pixels, `totalSize` is 21000 m in a normal world (2 × (World Size + Edge Size)), and a 16-bit pixel p stands for the value v = p / 65535.

Pixel and world

Column c counts from the left (west), row r from the top (north); x grows to the east and z to the north.

```
x = ( c / (size - 1) - 0.5 ) * totalSize
z = ( 0.5 - r / (size - 1) ) * totalSize

c = round( (x / totalSize + 0.5) * (size - 1) )
r = size - 1 - round( (z / totalSize + 0.5) * (size - 1) )
```

Better Continents reads every map smoothly between these points (a biome takes the one most of the 4 nearest pixels have). Only the location map counts differently, as Better Continents' location loader does, so its dots can sit up to a pixel off the other maps:

```
x = ( c / size - 0.5 ) * totalSize
z = ( 0.5 - (r + 1) / size ) * totalSize
```

Heights

```
world height (m) = ( v * amount - 0.15 + adjustment ) * 200
adjustment       = 1 - 2 * seaLevel              (0 at Sea Level 0.5)
v                = ( metres / 200 + 0.15 - adjustment ) / amount
the sea          = 30 m world height, so height above the sea = world height - 30
```

At Heightmap Amount 1 (the default) and Sea Level 0.5 this is:

```
world height = 200 * v - 30        black -30 m, white 170 m
above sea    = 200 * v - 60        waterline v = 0.30
one 16-bit step = 200 / 65535 = 3 mm      one 8-bit step = 200 / 255 = 0.8 m
```

At Heightmap Amount 2 and Sea Level 0.5 (`bc_export amount=2`) it is:

```
world height = 400 * v - 30        black -30 m, white 370 m
above sea    = 400 * v - 60        waterline v = 0.15
one 16-bit step = 400 / 65535 = 6 mm      one 8-bit step = 400 / 255 = 1.6 m
```

In general the waterline is v = (0.30 − adjustment) / amount, black is (adjustment − 0.15) × 200 m and white is (amount − 0.15 + adjustment) × 200 m of world height. A bigger amount raises white, and a higher sea level lowers black: for example `bc_export amount=3` stores −30 m to 570 m with the waterline at 0.1. Such an export lines up only with exports and maps made for amount 3 (chapter 5), and the sea still sits at 30 m.

The amount goes up to 81, where white is 16,170 m of world height. One 16-bit step is 200 × amount / 65535 m: 3 mm at amount 1, 1.5 cm at amount 5 and 25 cm at amount 81.

An export also writes `amount` and `seaLevel` (the config value, 0 to 1, not the adjustment) inside `heightmap.png`, as a PNG text chunk with the keyword `BetterContinents` and the text `Heightmap Amount = 1; Sea Level Adjustment = 0.5`. Image editors show it and often keep it. Better Continents reads it back, and a new world that reads the heightmap at other values writes a warning to

the log.

The heights are sampled the way the game builds its terrain: every 64 m zone blends the height formulas of the biomes at its four corners, with a smooth curve across the zone. Past World Size, Better Continents' edge drop-off pulls the ground down into the sea; with the drop-off on, the heightmap holds the ground before the drop-off, and Better Continents applies it again when it builds the new world.

Fine heights

`heightmap-fine.png` (chapter 4) holds, for every pixel, a byte f read as a signed number: 0 to 127 as they are, then 255 is −1, down to 128, which is −128. It counts 256ths of one 16-bit step of `heightmap.png`, and replaces v = p / 65535 with:

```
v = ( 256 * p + f ) / 16776960        16776960 = 256 * 65535
```

With f = 0 this is exactly p / 65535, so `heightmap.png` alone reads as it always did. A 4-bit file uses only multiples of 16 (−128 to 112): a sixteenth of a step, 1.5 cm at amount 81. The export writes f = 0 under water (below 30 m) and where the ground rises a whole step or more to a neighbouring pixel. With `fine=auto` it writes a 4-bit file where one step is more than 1.5 cm, 200 × amount / 65535 > 0.015, that is from an amount of about 4.92.

The file's record, a PNG text chunk with the keyword `BetterContinents`, reads `Fine Format = 1; Fine Bits = 4; Heightmap CRC-32 = 2723AFF1`: the format, the bits the bytes use, and the CRC-32 of the `heightmap.png` file it was made for. Better Continents does not use a fine file whose CRC-32 is another `heightmap.png`'s. `manifest.json` names the same in `fineBits` and `fineEncoding` (appendix C).

Heat

```
heat = v * Heatmap Scale                    (10 by default: white is 10)
Ashlands wherever heat > 0
```

The export is exact for the game's own Ashlands test: a pixel is black wherever the game says the ground is not Ashlands. Where it is Ashlands, the pixel is at least one grey step (so that heat > 0 holds), and otherwise v = (g + 0.00001) / scale, where g is the game's Ashlands ocean gradient. Better Continents reads back scale × v − 0.00001 as the gradient. g can still be below 0 at the border, because the game's Ashlands test and its gradient use slightly different lines; that is why the map is made from the test and not from the gradient alone.

Cold

```
cold = v * Coldmap Scale                    (10 by default)
Deep North wherever cold > 0
```

The export writes a pixel black outside the Deep North. Inside it the pixel is the game's wave fade (a ramp over the 200 m at the Deep North's border, from 0 to 1) divided by the scale (10), and never less than one grey step, so that cold > 0 holds exactly where the game says the ground is Deep North. The wave fade that the calm sea follows is Clamp01 of the cold value. A world's own cold map is exported as it is, with its own `Coldmap Scale`.

Forest

The game's forest factor runs from about 1.85 (bare) to about 0.15 (densest). Call n the same thing turned into 0 (bare) to 1 (densest): n = InverseLerp(1.850145, 0.145071, factor).

```
additive (Forestmap Multiply 0, Add 1):
    stored   f = n(wanted) - n(game)
    forest     = n(game) + f
exact (Forestmap Multiply 1, Add 1):
    stored   f = n(wanted) / (n(game) + 1)
    forest     = f * (n(game) + 1)
```

A stored value below 0 cannot be kept, so the additive form cannot make a place barer than the game's own forest; the exact form can.

Lava, moss and paint

`lavamap.png` and `mossmap.png` are 8-bit: the transparency channel of the game's ground paint in the Ashlands and the Mistlands, which the game itself keeps in 8 bits, so nothing is lost. Lava wherever the lava map is above 60 %. `paintmap.png` holds the red, green and blue channels of the ground paint, 8 bits each.

The editor pack

The pack (chapter 10) stores the height in metres h as a float v, for the pack's Heightmap Amount and the Sea Level Adjustment, the same arithmetic as `heightmap.png`:

```
fh      = h / 200
v       = ( fh + 0.15 - adjustment ) / amount
file    = max( v, 0 )                          no upper limit in height.tif
metres  = ( file * amount - 0.15 + adjustment ) * 200
```

`height.r32` is also clamped to 1. At Amount 2 and Sea Level 0.5, `file` 0 is -30 m, 1 is 370 m, and the sea (30 m) is at 0.15. The ground is as the world has it, so past the world's radius it includes the edge drop-off, which the heightmap leaves out. The grid is the one above: pixel i is at `(i / (size - 1) - 0.5) * totalSize`, and file row 0 is north. The pitch is `totalSize / (size - 1)`.

The masks.

Mask	Bits	Value
`water`	8	255 where h < 30 m, else 0
`water-inland`	8	the same, where the biome (the biome the biome map would give that point) is not Ocean
`slope`	16	round(slope in degrees / 90 × 65535). The slope is atan(hypot(gx, gz)) of the float heights as written, the floor included. The differences are central over one pixel on each side (so two pitches span), one-sided at the border
`forest-density`	16	round(density × 65535). The density is the game's forest factor turned into 0 (bare) to 1 (densest), absolute, unlike `forestmap.png`, which is a difference to the game's own noise
`biome-<Name>`	8	255 where the biome at the point is that biome
`biome-index`	8	the biome's index; `biome-index.txt` lists `index: name: R,G,B`
`biome-soft-<Name>`	16	round(weight × 65535): how much of the ground's height blend the biome has. The four zone-corner biomes carry the smooth-curve weights of the blend above, and a biome that holds several corners gets the sum
`protect-locations`	8	255 within the larger of a location's two radii plus the pad (8 m), falling in a straight line to 0 over the feather (32 m); the largest value over all locations

TIFF. What the pack writes is a classic TIFF 6.0, little endian, uncompressed, one 32-bit float sample per pixel, in strips of up to 64 rows, with the Orientation tag 1 (row 0 on top). Its description reads, for example, `BetterContinents: Height Encoding = bc-v; Heightmap Amount = 2; Sea Level Adjustment = 0.5; Orientation = north-up`. A 16384 px file is 1 GiB, which the 4 GiB limit of classic TIFF allows; larger is refused. A file is `<name>.tmp` until it is whole. **R32** is `W × H` little-endian floats, top row first, no header, clamped to 0 to 1; the reader takes the side as the square root of the byte length.

The way back (`bc_import edited`). Where the unit is `v`, a returned value f is h = (f × amount − 0.15 + adjustment) × 200 with the pack's amount and adjustment. With `unit=metres`, h = f. With `range=lo,hi`, h = lo + f × (hi − lo). The result is stored as `heightmap.png` and `heightmap-fine.png` for the pack's amount and sea level, or, with `amount=auto` or `sealevel=auto`, for the smallest of 2, 3, 4, 5, 6, 8, 10, 16, 32, 64, 81 whose top holds the highest ground, and the sea level raised in steps of 0.05 until the floor holds the lowest. With the edge drop-off on, the pixels past the world's radius are taken from the export's own `heightmap.png`, which holds the height before the drop-off; the rest of the picture is the returned height.

Appendix E

GNU General Public License, version 3

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

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

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

                            Preamble

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

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

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

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

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

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

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

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

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

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

                       TERMS AND CONDITIONS

  0. Definitions.

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

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

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

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

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

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

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

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

  1. Source Code.

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

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

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

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

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

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

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

  2. Basic Permissions.

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

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

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

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

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

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

  4. Conveying Verbatim Copies.

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

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

  5. Conveying Modified Source Versions.

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

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

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

"keep intact all notices".

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

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

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

6. Conveying Non-Source Forms.

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

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

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

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

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

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

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

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

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

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

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

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

7. Additional Terms.

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

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

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

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

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

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

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

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

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

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

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

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

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

  8. Termination.

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

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

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

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

  9. Acceptance Not Required for Having Copies.

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

  10. Automatic Licensing of Downstream Recipients.

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

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

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

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

  11. Patents.

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

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

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

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

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

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

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

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

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

  12. No Surrender of Others' Freedom.

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

  13. Use with the GNU Affero General Public License.

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

  14. Revised Versions of this License.

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

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

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

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

  15. Disclaimer of Warranty.

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

  16. Limitation of Liability.

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

  17. Interpretation of Sections 15 and 16.

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

                     END OF TERMS AND CONDITIONS

            How to Apply These Terms to Your New Programs

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

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

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

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

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

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

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

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

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

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

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

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