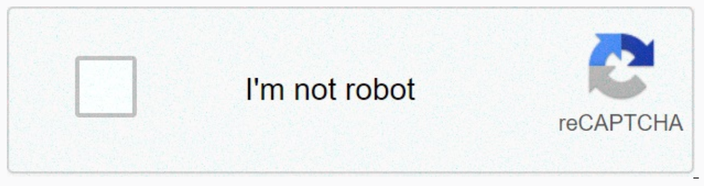

Continue

Minecraft biome bundle list

Author: MC_Pitman, team_otg | July 18, 2019 | 195,777 views Biome Bundle Mod 1.12.2/1.11.2 is a completely unique world generator containing a huge package of more than 400 original bioms with over 2,000 structures, including custom trees, rocks, caves, dungeons, villages and much more. Features: Unique biomes and terrain. Unique objects (trees, rocks, etc.) Custom structures (Dungeons, ruins, etc.). Individual ore generation. Custom mob spawn rules (selected biomes). Screenshots: Alpine: Half covered in snow with huge and varied mountains and rare coniferous trees. Autumn forests: Rolling hills and grassy plains covered with oak trees with seasonal leaf colors. bald cypress forest enchanted forest extreme hills jungle: hilly, mountainous and overgrown, difficult lifelong, but perfect for those who prefer to live in those secessive difficult to reach places. Frozen forest hides in the mind of High Cliffs: A picturesque but difficult to pass through a combination of cliffs, valleys, covered forests and high peaks. Mangrove Forest: Dark green waters are home to tall Mangrove trees, lily pads and small volcanoes. It is one of several variants of Swampland. Random mountain between biomes Sandstone Mesa: Vivid sandstone and clay is a combination of baobab trees and surface stalagmites. Tropical savanna of the tropical rainforest of Mesa: the hot savanna is broken up with clay-coated canyons, high plateaus and unique rock formations. Tundra: Mostly flat, snowy and treeless patches of coniferous forests and mountains. Webbed Forest wet forest: covered with cliffs, rivers and lakes and lush unique foliage on the occasional mountain surrounded by covered woods. Required: Minecraft Forge Open Terrain Generator Mod How to Install: Make sure you've installed Minecraft Forge and Open Terrain Generator Mod. Find the Minecraft app folder. Under Run from Start, type %appdata%, and then click Run. For Mac open finder, hold down ALT and click go then library in the top menu bar. Open the App Support folder and find Minecraft. Place the mod you've just downloaded (.jar file) into the Mods folder. Now that everything is installed, you can create a new world, simply select Biome Bundle on the normal world creation screen. Biome Bundle Mod 1.12.2/1.11.2 Download links: Minecraft 1.10.2 Download from server 1 – Download from Server 2 Minecraft 1.11.2 Download from Server 1 . You can find biome config files from around the world: Spigot: /plugins/OpenTerrainGenerator/worlds/_worldname_/WorldBiomes Forge: /mods/OpenTerrainGenerator/worlds/_worldname_/WorldBiomes There's also a GlobalBiomes folder which allows you to share biomes with multiple worlds on the same server. Each Biome configuration file is divided into sections, each representing a specific aspect of the production process. Here we list them all with a brief explanation of usage, remember that all Biome configuration files can already make a lot of comments. Biome inheritance Edit Defined by BiomeExtends: If you enter biome here, OTG will use the specified biom settings for those missing in the current Biome configuration. When you work with certain worlds, you can often create more, very similar, life lifes. If you want to make a change to all of these bioms, you'd have to go through each configs and change your settings. Biome inheritance solves this problem by introducing the mechanic of the child biomes inherits settings from the parent to one. For example, if you created MyForest Biome (with all the specified settings) and want to make a hillier version, you can only create a new biom called MyForestMountains, where you can specify BiomeExtends: MyForest and set only BiomeVolatility to 0.5. All other missing settings in MyForestMountains are inherited from the MyForest forest. From now on, all changes to MyForest will be transferred to MyForestMountains, except BiomeVolatility: 0.5. Resource InheritanceEdit: ResourceInheritance: A logical value, if true, all resources in the parent community (if any) are transferred to the biom resource queue. If a resource in the parent community is very similar to a child biome (for example, two ores of the same type), it is not copied. Biome Placement Edit BiomeSize: From 0 to GenerationDepth, adjust the size of the life material when created by a BiomeGroup. See also BiomeSizeWhenIsle and BiomeSizeWhenBorder below. BiomeRarity: Between 0 and 100, the probability of a life library when a BiomeGroup is created. See also BiomeRarityWhenIsle below. BiomeColor: This is the internal (hexadecimal) color that should be used for this theme of life in FromImage mode and on the map of the life service. ReplaceToBiomeName: Only available in custom bioms. This biom will be replaced by the one specified after the terrain is established. Isle Biomes Only Edit IsleInBiome:String, another life material name. It determines which biom this biom will breed as an island. BiomeSizeWhenIsle: from 0 to GenerationDepth, the size of this biom when an isle was born as biom. Larger numbers give smaller islands. The biom must be smaller than the biom in which it will reproduce, so the BiomeSizeWhenIsle number must be greater than the BiomeSize of the other life. BiomeRarityWhenIsle: A rarity in biom when it gave birth as an isle biom. It seems to have a logarithmic scale: the rarity of the 100 makes the biom spawn everywhere from the parent biom, 99 only 50% of the biom, and so on. Biomes Only Edit BiomeIsBorder:String, another life material. It determines which life-wanter will have a limit. NotBorderNear: String, another life material name. This setting determines which life material cannot reproduce. BiomeSizeWhenBorder: from 0 to GenerationDepth, the size of this biom if it appears as a border life. Larger numbers provide smaller boundaries. The biom must be smaller than the biom in which it will reproduce, so the BiomeSizeWhenBorder number must be greater than the BiomeSize of the other life. Terrain height and volatility Edit BiomeHeight:Float -10.0-10.0, the height of the biom, the table below shows the resulting Y level: BiomeVolatility: Float, how hilly the biom is, higher values make the terrain more hilly. SmoothRadius: This setting must be in the range 0-32 and its scope facilitates the transition between the two neighboring partners. The formula is as follows: [SmoothRadius (from this biom) + 1 + SmoothRadius (from the neighboring world)]*4 Therefore, two bioms with SmoothRadius 3 and 5 result in a 36 block wide transition ((3+1+5)*4). Rivers Edit This section rules how rivers are formed, and it is divided into two mutually excluded configurations based on ImprovedRivers: option WorldConfig.ini. When You read ImprovedRivers: fake The RiverBiome: {biom name} option and consider it, here you can choose which life material to use as a river, usually using the Vanilla River or frozenriver biom (here you can check them out). This is likely to cause a non-homogeneous transition to the landscape, here's where the ImprovedRivers: true comes in handy. When ImprovedRivers: true these options become available: RiverHeight: float behaves like BiomeHeight (see Biome_Configs #Terrain_Height_and_Volatility) RiverVolatility: float acts as biomevolatility (see Biome_Configs #Terrain_Height_and_Volatility) RiverWaterLevel: {Y coordinate in the world} acts like a filler, all empty space between the surface and this quote will be filled with water. RiverCustomHeightControl: acts like CustomHeightControl (see Biome_Configs #Terrain_Height_and_Volatility) Blocks Edit Water/Lava & Frozen States Edit This section is a main switch that uses the UseWorldWaterLevel option, which is true by default, which means that OTG ignores all other settings in this section and by default in the global world as well. Changing the option to fake allows for options that are almost self-evident: WaterLevelMax: WaterLevelMin: Across from 0 sky limit, these options set higher and lower Y limits where water (or a custom block) will be spawned when the surface Y height is lower than WaterLevelMax and higher than WaterLevelMin. WaterBlock: The string name. What to spawn when the above criteria are met. IceBlock: The string name of the block. Which is which must be spawned as ice instead of the water whenever the BiomeTemperture is below the default Vanilla Minecraft 0.15. CooledLavaBlock: Same as IceBlock, but lava replacement when too cold for liquid lava exists. Visualizations and Weather Editing To install OTG server, this setting will only affect customers who have OTG installed. BiomeTemperature: A value between 0.0 and 2.0 determines the Biome temperature according to Vanilla standards. If it is set in the Vanilla Biome replacement will only affect the foliage and grass color. BiomeWetness: A value between 0.0 and 1.0 gives you the chance that rain may occur. SkyColor: Color: GrassColor: FoliageColor: It goes without saying, it's a color expressed in hexadecimal value, here you can read a list of common colors, otherwise Google is your friend =). Remember to prepend your #. GrassColorIsMultiplier: FoliageColorIsMultiplier: Boolean true or false, simpy up tells OTG if the grass and foliage colors are always exactly as defined above (false) or if the color can change as a blend of the above set color in the BiomeTemperature and BiomeWetness settings (true). Resource Queue Edit: This section controls all resources that are created after terrain generation. The sources are placed in the order in which they are determined, here's an example in which Ore Dirt is added before Ore pebbles. Keep in mind that high resource frequency or rarity can slow down field production. Ore(DIRT,33,10,100.0,0,255,STONE) Ore(PEBBLES,33,8,100.0,0,255,STONE) These are possible sources like OTG 1.12.2+-+v6 DoResourceInheritance(true|fake) SmallLake(BlockName, Frequency, Ra rity,MinAltitude,MaxAltitude) dungeon(Frequency,Rarity,MinAltitude,MaxAltitude) UnderGroundLake(Minsize, MaxSize, Frequency,Rarity,MinAltitude,MaxAltitude) Ore(BlockName,Size,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,BlockSource3.....]) UnderWaterOre(blockname,size,frequency,rarity,blockSource[,BlockSource2,BlockSource3.....]) CustomObject(Object[,AnotherObject[,...]]) CustomStructure([Object,Object_Chance[,AnotherObject,Object_Chance[,...]]] SurfacePatch(BlockName,DecorationBlockName,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,BlockSource3.....]) Tree(Frequency, TreeType,TreeTypeChance[,AdditionalTreeType,AdditionalTreeTypeChance.....]) Plant(PlantType,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,BlockSource3.....]) Grass(PlantType,Grouped/Ungrouped,Frequency,Rarity,BlockSource[,BlockSource2,BlockSource3.....]) Reed(BlockName,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,BlockSource3.....]) Cactus(BlockName,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,BlockSource3.....]) Liquid(BlockName,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,BlockSource3.....]) Nain(BlockName,MinRadius,MaxRadius,Rarity,OreSize,OreFrequency,OreRarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,..]) Well(BaseBlockName,HalfSlabBlockName,WaterBlockName,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,..]) Boulder(BlockName,Frequency,Rarity,MinAltitude,MaxAltitude,BlockSource[,BlockSource2,..]) IceSpike(BlockName, IceSpikeType,Frequency,Rarity,MinAltitude,MaxAltitude,Blocksource[,BlockSource2,...]) ====Explaination/Example usage of the resources: BlockName: must be the name of a block. It can contain block data, such as WOOL:1. BlockSource: a list of blocks that the resource can bring in/enter. You can also use Solid or All or All. Frequency: The number of attempts to place the resource in each chunk. Rarity: chance for all experiments, Rarity: 100 - 100% average, Rarity: 1 - 1% passage. MinAltitude and MaxAltitude: height limits. BlockSource:average where or where resource is placed TreeType: Tree (original oak tree) - BigTree - Birch - TallBirch - SwampTree - HugeMushroom (randomly red or brown) - HugeRedMushroom - HugeBrownMushroom - Taiga1 - Taiga2 - HugeTaiga 1 - HugeTaiga2 - JungleTree (the huge jungle tree) - GroundBush - CocoaTree (smaller jungle tree) DarkOak (a covered forest biom) - Acacia You can also use your own custom objects, as long as as long as they set Tree true to the settings. TreeTypeChance: Similar to Rarity. Example: Tree(10, Taiga1,35,Taiga2,100) - plugin tries 10 times, each attempt tries to place Taiga1 (35% chance), if it fails, tries to place Taiga2 (100% chance). PlantType: one of the plant types: Allium, AzureBluet, BlueOrchid, BrownMushroom, Dandelion, DeadBush, DoubleTallgrass, Fern, LargeFern, Lilac, OrangeTulip, OxeyeDaisy, Peony, PinkTulip, Poppy, RedMushroom, RedTulip, RoseBush, Sunflower, Tallgrass, WhiteTulip or simply a BlockName IceSpikeType: one of the ice spike types: Basement, HugeSpike, SmallSpike Object: can be any custom object (bo2 or bo3), but without the file extension. You can also use UseWorld to create one of the objects in the WorldObjects folder, and usebiome to create an object in the BiomeObjects setting. If you are using BO2s for UseWorld, BO2 must have this biom in the spawnInBiome setting. Plant and grass resource: also a block resource. The plant can place the blocks underground, the grass can't. Liquid resource: a block of water or lava source smalllake and undergroundlake springs: small lakes about 8x8 blocks vein resource: no It triggers an area where ores multiply. It can be slow, so use low Rarity (less than 1). CustomStructure resource: starts a bo3 structure in the chunk. Herding Resource Edit OpenTerrainGenerator allows you to grow custom objects instead of tree plants instead of vanilla

trees. Add one or more of the herd functions to override the spawning of vanilla to the tree. Syntax is: Tree(Tree Type,TreeType_Chance[,Additional_TreeType,Additional_TreeType_Chance.....]) It works as the Tree resource instead of the first parameter. Groin types: All, Oak, Redwood, Birch, SmallJungle, BigJungle, RedMushroom, BrownMushroom, Acacia, DarkOak, HugeRedwood All - makes the tree spawn all the groceras, but not the mushrooms. BigJungle - because when 4 jungle children grow at the same time. RedMushroom/BrownMushroom - only grows when bone meal is used. Custom Objects Edit: These objects appear when you use the UseBiome keyword. BiomeObjects: Structures Edit Here you can change, enable or disable stuctures. If you have disabled the structure of WorldConfig, it will not spawn, regardless of these settings. It disables the forts of life in life. If you don't have a proper life nearby, Minecraft ignores this setting. StrongholdsEnabled: Fake that a Woodland Mansion can be placed in this biom. WoodLandMansionsEnabled: fake that an Ocean Monument can be placed in this biom. OceanMonumentsEnabled: it's false whether an Underworld Fortress can start in this biom. It may extend to neighboring people. NetherFortressesEnabled: true The type of village in this biomban. It can be wood, sandstone, taiga, savanna or disabled. VillageType: Disabled the mine type in this biomban. It can be normal, mesa or disabled. MineshaftType: normal Mine rarity from 0 to 100. 0 = no eneka, 1 = default rarity, 100 = wooden chaos. Keep in mind that mines will never spawn, regardless of this setting, if MineshaftType was set to Disabled MineshaftRarity: 1.0 The type of above-ground rare building in this biom. It can be desert, jungle Temple, swampHut, igloo or disabled. RareBuildingType: Disabled mob spawning Edit Can't change mob spawns vanilla biomes. These are the values that vanilla uses for this biom. They are read-only: # changes to this setting are ignored and overwritten. The monsters (skeletons, zombies, etc.) that spawn in this biom Example [{mob: Spider, weight: 100, min: 4, max: 4}, {mob: Zombie, weight: 100, min: 4, max: 4}] Use the / OTG organizations console command to get a list of possible mob and mob types. Use /otg biom -m console to get a list of registered mobs for a civil partnership. SpawnMonsters: The friendly creatures (cows, pigs, etc.) that spawn in this biom Example [{mob: Sheep, weight: 12, min: 4, max: 4}, {mob: Pig, weight: 10, min: 4, max: 4}] Use the / otg organisms console console make a list of possible mob and mob types. Use /otg biom -m console to get a list of registered mobs for a civil partnership. SpawnCreatures: [] Aquatic creatures (only squid vanilla) that spawn in this biom Example [{mob: Squid, weight: 10, min: 4, max: 4}] Use the /OTG organisms console command to make a list of possible mob and mob types. Use /otg biom -m console to get a list of registered mobs for a civil partnership. SpawnWaterCreatures: [] Environmental creatures (only bats vanila) that spawn in this biom For example [{mob: Bat, weight: 10, min: 8, max: 8}] Use the /otg organisms console command to make a list of possible mob and mob types. Use /otg biom -m console to get a list of registered mobs for a civil partnership. SpawnAmbientCreatures: [] Forge Biome Dictionary ID is used by other mods to identify life people and place modded blocks, objects and mobs in it. It only works with modded elements/blocks/mobs that are placed in bioms while the pieces are being created. Most mods that add mods add to the mob's biomes internal mob list when MC starts up and let MC mob spawning mechanics handle the actual spawning. This means that when TC creates new bioms, when it creates a world, other mods don't add the mob to these bioms. This can be done by using InheritMobsBiomeName to inherit a mob list from a vanilla living area. Note: Only works with biomes identification conclusions < 255 (non-virtual biomes). The virtual bioms of BiomeDictId are inherited via ReplaceToBiomeName. BiomeDictId: See list. Inherits the inner mob list of another lifeman. Legacy mobs can be overridden with SpawnMonsters, SpawnCreatures, SpawnWaterCreatures, and SpawnAmbientCreatures. The mob types defined by these settings override the legacy mob settings of the same mafia. Use this option to inherit modded mobs from vanilla bioms (see also BiomeDictId) InheritMobsBiomeName: A list of . Modded creatures work the same as vanilla creatures, but with different EntityNames. Where first sheep {mob: Sheep, weight: 12, min: 4, max: 4} you need to use entityclassname. They are easy to find using mods tellme. To query the entityclassnames list, use /tellme dump entities. Entities.