BuildCraft Handbook
by SpaceToad

Edition 2.2
BuildCraft 2.2.x

1 Introduction

BuildCraft is a Minecraft mod that aims at providing advanced building capabilities to the game. It is currently split in four parts: Transport, Factory, Builders and Energy. Transport provides means to move items easily across the map. Factory allow the extracting of resources and automatically craft tools and materials. Builders adds automatic builder mechanisms. Energy provides advanced means of providing power to machines.

All of these mods are independents. The player may want to install one of the other depending on the user experience they're wishing to achieve, or for other technical considerations such as constraints coming from other mods. However, while independent, these mods have been designed to work together, and installing them all will provide additional interaction capabilities.

BuildCraft has been designed to be used on an advanced world. You will find that many recipes require a substantial amount of diamond, lapis or redstone, which are typically available after a few hours in survival mode.

2 Base Concepts

1   Inventories

Pipes and machines usually know how to interact with inventories. An inventory is basically anything that can contain object. In the base game, inventories can be things such as chests, furnaces or dispensers. BuildCraft adds a couple of inventories such as the Automatic Crafting Table or the Filler. Other mods may add their own inventories as well. In most cases, BuildCraft will know how to interact with them out of the box.

Certain inventories have a special status:

– Three slots inventories: These inventories have three slots, typically two inputs and one output. This is the case of e.g. the furnace.
– Two slots inventories: Although Minecraft doesn't provide any of these, certain mods have implemented a two-slot inventories, working like input / output devices.
– Special inventories: Modders can specify an inventory to have a special behavior. This is for example the case of the Diamond Pipe or the Automatic Crafting Table. There's no way to deduce from the inventory itself if it has to be considered special or not – modders have the responsibility of documenting that.
– Passive inventories: This is basically all other kinds of inventories.

2   Chunk

A chunk is a 16x16 area on the world. This is the base area size used when world generation. A chunk population event (e.g. generation of a deposit of oil) is expressed as a probability per chunk generation.

3   Smooth Blocks

Several BuildCraft machines consider two different kind of blocks – smooth blocks and solid blocks. Smooth blocks includes all blocks that the player can go through, e.g. still water, running water and air. BuildCraft devices will consider all of them as being like air, and thus will build into these blocks, and won't be able to dig / remove them as they would with solid blocks.

4   Speed

The unit used to specify speed in this document is block per cycle, or bpc. One block has the length of a regular Minecraft block, and one cycle is the timespan where Minecraft updates the game, the smallest time between two minecraft event. How many cycles in a second may depends on your actual hardware configuration, but should be roughly about twenty. So 1 bpc is a fairly high speed. Slower objects' speed will often be refered to as mili block per cycle, or mbpc.

5   Powering

Most BuildCraft machines needs to be powered, historically, prior to BuildCraft 2.0, every machine was powered by alternate redstone current (see below for explanation). Starting BuildCraft 2.0, machines can be powered with different power mechanisms, that can be selected from the global setting "power.framework".

Every machine has several properties related to its interaction with power. Power framework can take into account some of them, and ignore others.

Latency

Number of cycles between which a device can be activated again. This is mostly a legacy property used by the simple Redstone power framework, but can be used by third party frameworks.

Minimum of Energy Received

Minimum amount of energy that the device can accept from a powering machine.

Maximum of Energy Received

Maximum amount of energy that the device can accept from a powering machine.

Maximum of Energy Stored

Maximum amount of energy that the device can store at a given time

Minimum of Energy for Activation

Minimum amount of energy that the device has to have stored prior to activation.

Power Loss per Cycle

The average loss of power per cycle for a machine that has stored power.

Power is expressed in a made up unit, MJ for Minecraft Joule. Values needed to operate a machine can go from 1 to several hundreds.

BuildCraft implements two different power mechanisms, the legacy redstone power, and the pneumatic power. Third party mods can add their own powering framework relying on other devices.

Liquids

The unit used to describe liquid volumes is the bucket (B). A bucket is the amount that can be extracted from a block of liquid, such as water, lava or oil. BuildCraft mechanisms can manage bucket, or smaller amounts (mB for mili Bucket).

3 Minecraft Forge

Minecraft Forge is not a BuildCraft component per-se, but a community effort, which purpose is to standardize certain core modifications between mods. This will improve compatibility over mods using the same interfaces.

Minecraft Forge has to be installed separately from BuildCraft, straight into the jar files because it's doing modifications into the core Minecraft files. However, as other mods can include it in their code straight out, it's possible that one or several mods already installed on Minecraft already contain the Minecraft Forge. That can usually be checked on the mod distribution page.

4 Core Mod Contents

1 Gears
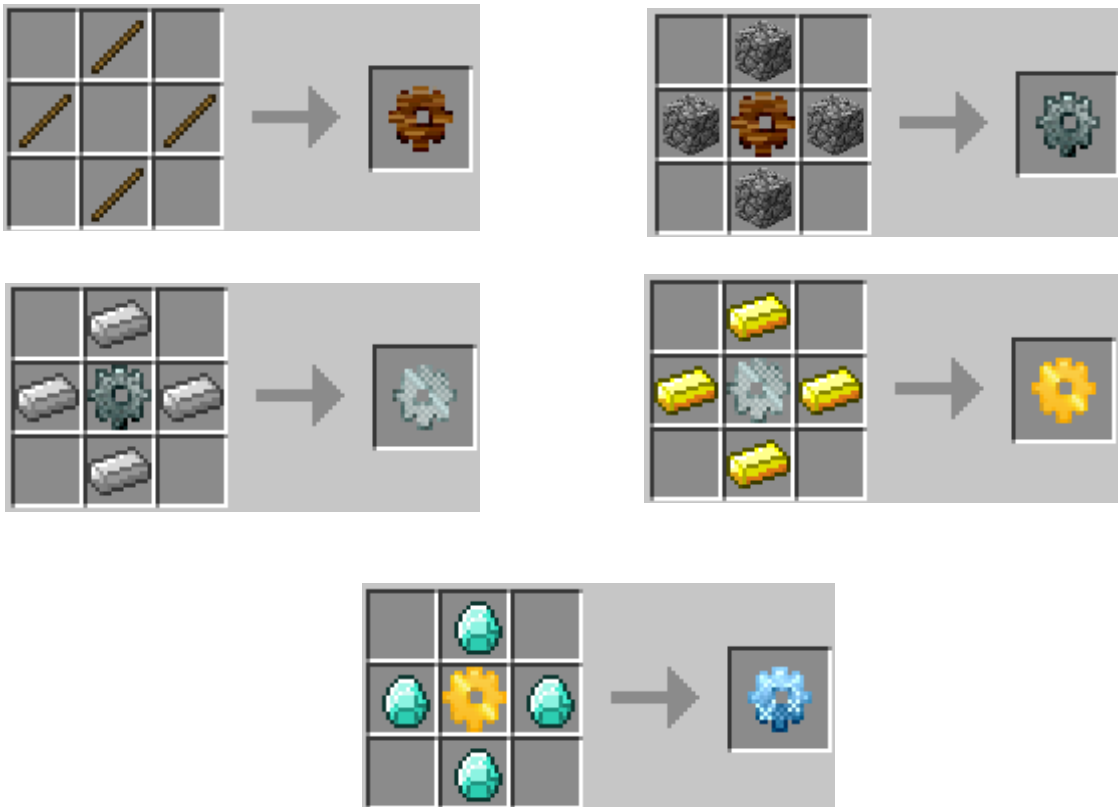
Gears are at the basis of many machinery recipes. The base gear is crafted out of four wooden sticks, and then upgraded to stone, iron, gold and diamond, so that a diamond gear costs in total four wooden sticks, four piece of cobblestone, four iron ingots, four gold ingots and four diamonds. The relative expensiveness of these items is to put in relation with the complexity and the power of the technology that they allow crafting.

2 Redstone Power Framework

This is the "historic" BuildCraft power mechanism, based on continuous or alternate current. This power framework do not take into account energy parameters such as "minimum energy for activation" or "max energy received". It does not actually transfer MCJ to devices. However, it will activate device after the specified delay.

By default, this framework relies on an alternate source of redstone current (AC) is needed, which means that the power has to be switched on / off regularly to have machines working. This can be achieved by various mechanisms know as clock, or pulsar. Several mods provide additional way of generating alternate current as well.
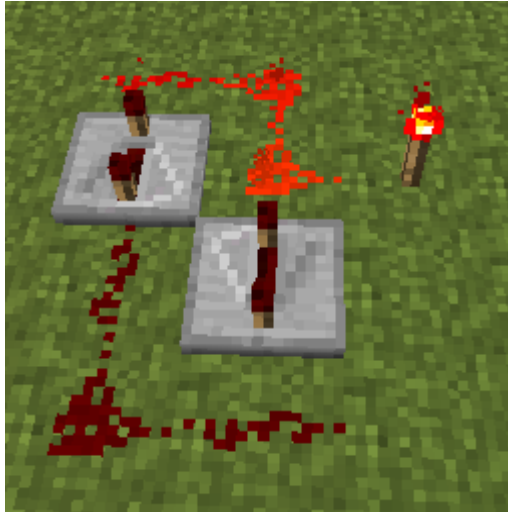
Illustration 1: Two repeaters clock - ignition phase

Illustration 2: Two repeaters clock - after ignition

In some cases, using that mechanism may be CPU-consuming. If the BuildCraft property "current.continous" is set to false, then machine will be activated upon direct current (DC), potentially saving CPU time. This is however considered as being a "cheat" mode, since it changes the mod balance.

Illustration 4: Pulsar

Illustration 3: Auto-initialized repeater

Note that BuildCraft machines are powered through indirect power. This mean that the redstone signal can go directly to the machine, or to a block next to it.

3   Wrench

The Wrench is the item allowing to change orientation of BuildCraft devices, namely wooden pipes input / output or template / builder / engine direction. It operates when right clicking on a device with the wrench in the hand.

5 Transport Mod Contents

1   *Introduction*

Pipes are the basis of the BuildCraft transportation system. They all can be connected on their 6 ends to either other pipes, inventories, machines able to push items into pipes, liquids tanks...

There are three main families of pipes, each of them able to transport three different kind of things : item pipes, liquid pipes and power pipes. Pipes are initially created as item pipes, which means that they are able to transport items. They can then be upgraded either to waterproof pipes (for liquids) or conductive pipes (for power).

Item pipes

By default, when an item enters a pipe, it will go straight to the center, and then orient itself in one of the available locations, either randomly or following a pipe-specific rule. If there's no possibility, e.g. there are no connections, or no connection can accept items (i.e. if the inventories are full), the item will be dropped. All pipes can receive objects from pipe pushing machines, and all pipes can place objects into inventories.

Items travel in the pipe as a constant rate unless accelerated. They can go in either direction – collision is not managed by the system. When a pipe is broken, it will release all objects that were traveling into it.

When connected to a three slot inventory, the top pipe will interact with the top slot, usually the place where the smeltable object is placed, the bottom pipe to the bottom slot, when the fuel is usually expected, and all pipes on the side will interact with the third one, typically the output.

When interacting with a two slots inventory, the pipes on top and bottom will interact with the first slot, typically the input, while the pipes on the bottom will interact with the second slot, the output.
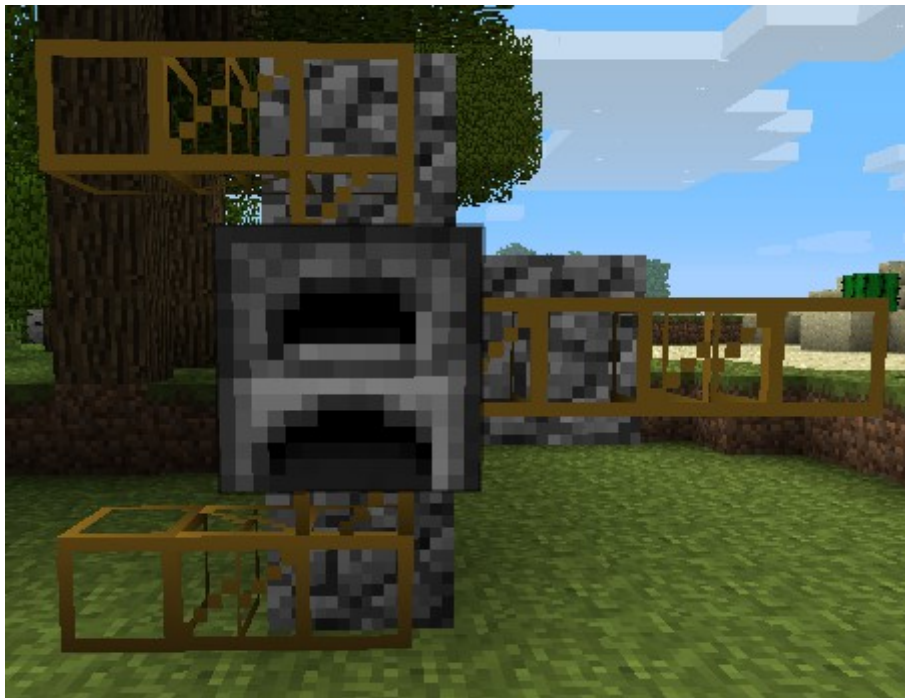
Illustration 5: Pipes connected to a three-slots inventory

Until BuildCraft 1.6.6.3, all pipes could be connected together. This is not the case anymore, but for legacy worlds, it is possible to get back that behavior by setting the configuration property "pipes.alwaysConnect" to "true".

By default, objects are moving at the speed of 10 mbpc. When they are accelerated, they will slowly decelerate, 10 mbpc each time they cross a pipe, until reaching their base speed.

Liquid pipes

Pipes can also transport liquids. As for 2.2.0, the kind of liquids that can be transported are water, lava, oil and fuel. Each section of the pipe can store up to a quarter of bucket. This means that a pipe connected to all ends can contain up to 1.75 buckets.

Pipe waterproof is required to provide to pipes the ability to carry liquids. This waterproof is crafted out of cactus greed dye.

Speed of liquids traveling in pipes varies depending on the kind of pipe. By

default, it takes 12 cycles for liquids to traverse a section, and 20 mB of liquid can be added to a section per cycle. These two properties are know as **latency** and **flow rate**.

A given section of a pipe can only contain one kind of liquid at a time. Other liquid kinds will be automatically bounced.

Illustration 6: Liquids traveling through pipes

Power Pipes

Power pipes allow to transport power from a power source (for example an engine) to a machine requiring power. They are mainly intended to be used along with the energy mod.

When connected to a machine requesting power, power pipes will automatically broadcast the energy request, and direct power already in the pipe system to them. Power and power request travels in pipes at the speed of 1 block per cycle, which explains the few instants latency between the time where a device is connected to a power network and the time where it's actually powered.

When reaching a junction, power is split between the different ends requesting power, in proportion to the amount requested by each one.
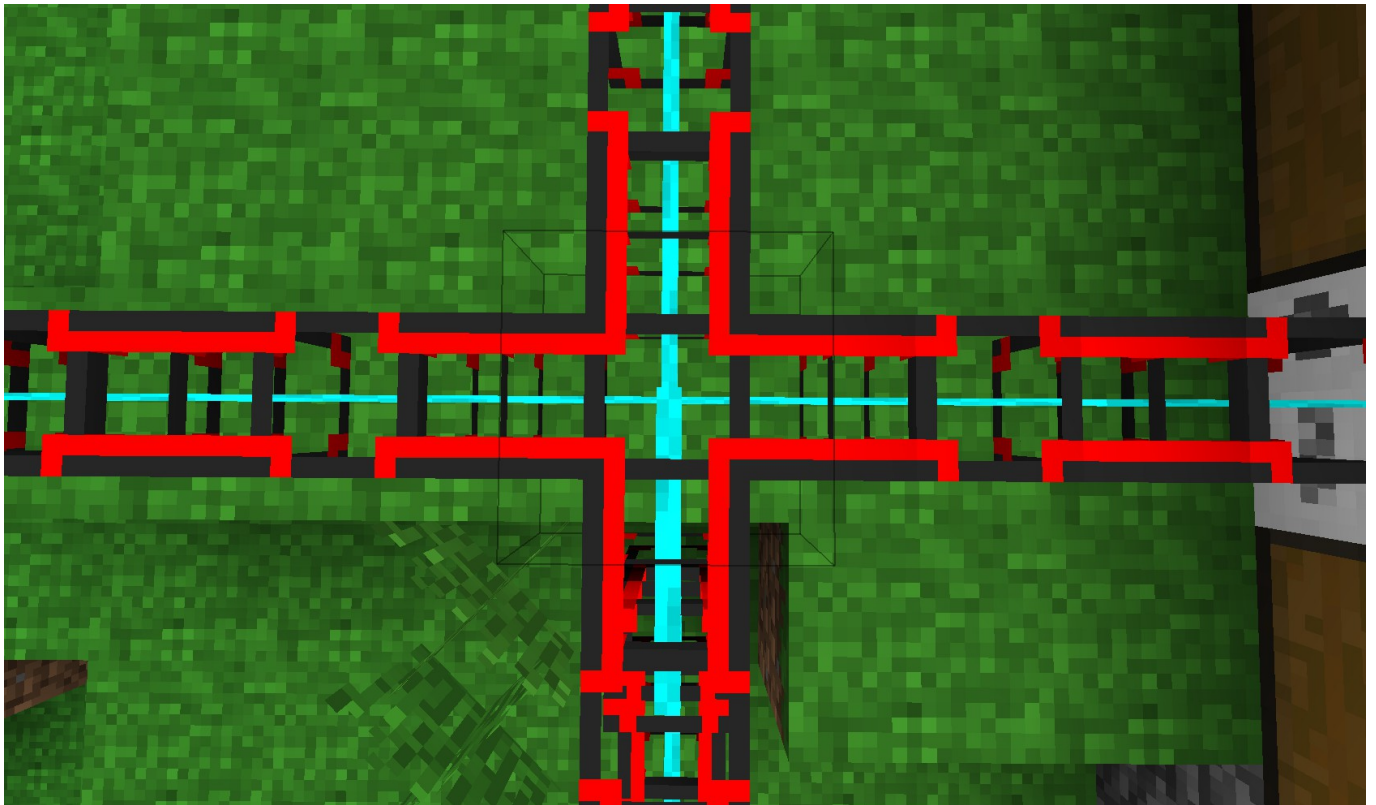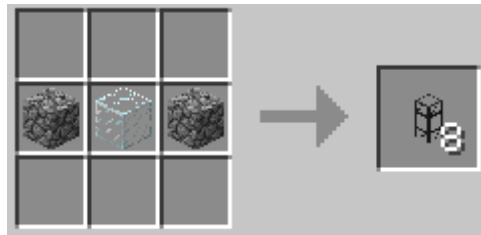
Illustration 7: Power split to three directions

A small amount of power is lost each time power travels through a pipe. The amount of loss depends on the kind of pipe. By default, this loss is of 1%, making pipes suitable to connect nearby devices but not for energy traveling across long distances. This property is know as **power resistance**.

2   Cobblestone Transport Pipe

Cobblestone pipes are the most common kind of pipes. Because they're crafted out of two block of cobblestone and one block of glass, they're also the cheapest.

The only pipe default mechanism that they override is that they cannot be connected to stone pipes. Other than that, they can interact with other pipes, put items into inventories, receive items from machines, and send items into random directions when forking.

3 Stone Transport Pipe
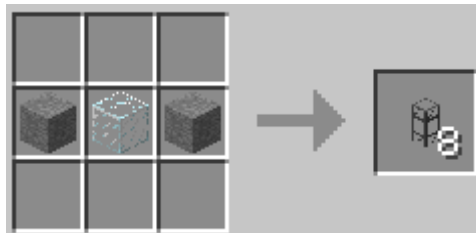
Stone pipes are very close to cobblestone pipes. Besides the crafting recipe, they have only two differences : first cobblestone pipes cannot be connected to stone pipes, which allow creation of denser unidirectional pipe networks, as shown in the illustration. They will make a better job at preserving object speed as well. While objects decelerate by 10 mbpc when entering most pipes until reaching floor speed, they will only decelerate by 5 mbpc in stone pipes, allowing to preserve speed on a longer distance.
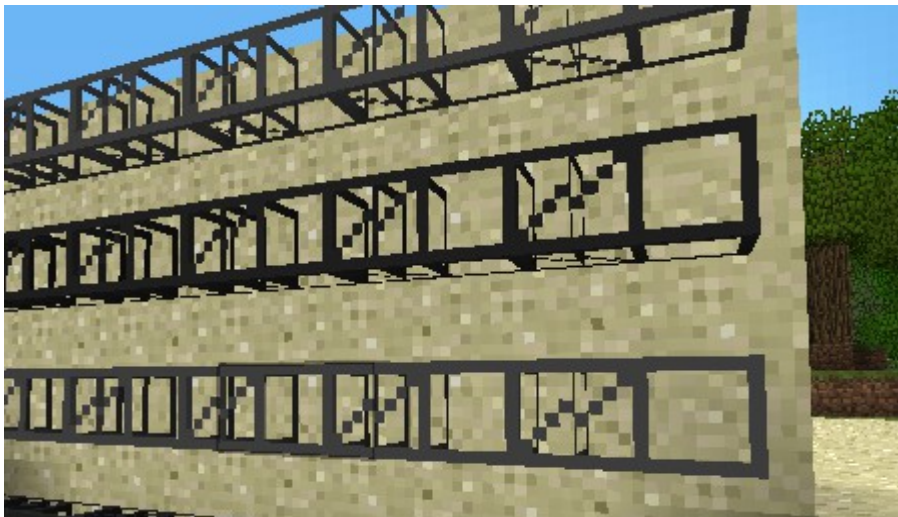

Illustration 8: Stone and cobblestone pipe network

4 Wooden Transport Pipe

Power Properties	
Latency	50
Minimum of Energy Received	1
Maximum of Energy Received	64
Maximum of Energy Stored	64
Minimum Energy for Activation	1
Energy Loss per Cycle	64

Wooden pipes allow the extraction of objects from inventories sharing all the common pipes properties. They and can be connected to any kind of pipes, except to other wooden pipes. When provided with a source of power, they can extract items from a chest.

Wooden pipes will only extract one stack at a time. They need 1 MJ per item of the stack to be moved. If they're powered with less energy, they will extract only as much item as they can, leaving part of the stack in the chest.


Illustration 9: Right-to-left wooden pipe extraction

Wooden pipes are polarized. They will only extract objects from their plain side so that it's possible to use just one pipe to connect two chests. Right-clicking on the pipe with a wrench will switch the extraction end.

It is possible to explicitly list inventories not to be considered by wooden pipes using the setting "woodenPipe.exclusion" in the buildcraft.cfg file. See the related section later in this document.

5   Iron Transport Pipe
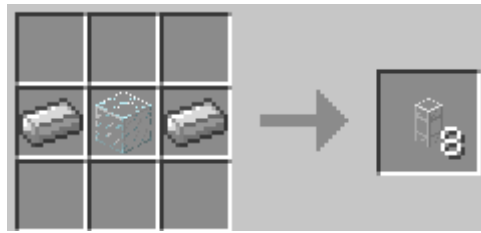
Iron pipes allow to lead objects always in the same direction, as opposed to a random one which is the case for most other pipes. Iron pipes have only a transparent end, all others are solid. Objects can come from any end of the pipe, but they always will be directed towards the transparent end. As a matter of fact, if an object is sent through the transparent end, it will bounce back. This end can be modified by either right-clicking on the pipe with a wrench, or by changing the power status next to the pipe with e.g. a lever.

Liquids traveling in iron pipes are following the same rules as objects – that is, they will only travel to the transparent end.

Iron pipes can never send objects to machines that are only in output mode (e.g. quarry or mining well) or to wooden pipes. This last characteristic allow to design a system with three pipes going to the iron pipe, two of witch being non-wodden. Switching the power in that situation will switch the direction back and forth between the two non-wooden pipes.

6   Golden Transport Pipe

When powered by redstone, golden pipes accelerate objects to 200 bpc. Items will slow down by 10 bpc each time they cross a pipe after that, except for stone pipes, which only decelerate by 5 bpc. You can check that a golden pipe is properly powered by looking at the texture which will be a little bit darker. Note that two golden pipes cannot be connected together.

Illustration 10: Unpowered and powered golden pipes

7   *Diamond Transport Pipe*

Diamond pipes allow to send objects into specific directions depending on their kind. A diamond pipe has a different color at each end, each of one corresponding to a different filter. Right clicking on the pipe will open the filter inventory. In this view, each row correspond to an end of the filter pipe, the correspondence between rows and ends being identified by the colors.

By default, diamond pipes behave like regular pipes – items will be sent randomly to each direction. As soon as an item is added to a filter, only items of the corresponding kind will pass through that filter, other items will go to other ends if available, or be dropped if there's no available filter. Unlike with iron pipes, items in diamond pipes do not bounce back.

It is possible to let several kind of items pass through a given filter by adding them one after the other to the corresponding inventory row – making it possible to filter up to nine item kinds. If an item is present in several filters, then the direction will be chosen randomly amongst the possible filters. Adding more than one occurrence of an item will increase the probability of an object to take that direction. If an item has the choice between two directions, one filtered by one occurrence, and the other one by two, it will have twice as much chance to go into the second one than the first one.

Diamond pipes are special inventories. Pipes cannot extract or place objects from them as they would with regular inventories.

8   Obsidian Transport Pipe
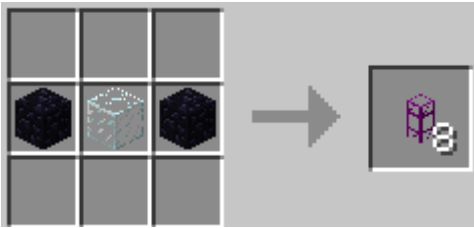
Power Properties	
Latency	25
Minimum of Energy Received	1
Maximum of Energy Received	64
Maximum of Energy Stored	256
Minimum Energy for Activation	1
Energy Loss per Cycle	1

Obsidian pipe vacuum objects from the ground. By default, when not powered, they will only get the object when sent right to them. However, if powered, obsidian pipes will suck items placed objects placed up to 5 blocks in front of them. The number of objects that they can extract per unit of time is proportional to the power that they have at their disposal, objects 1 block ahead costs 1 MJ, 2 blocks costs 2 MJ etc. They can suck stacks partially or in totally if they have enough energy. They will be able to suck arrow on the walls as well.
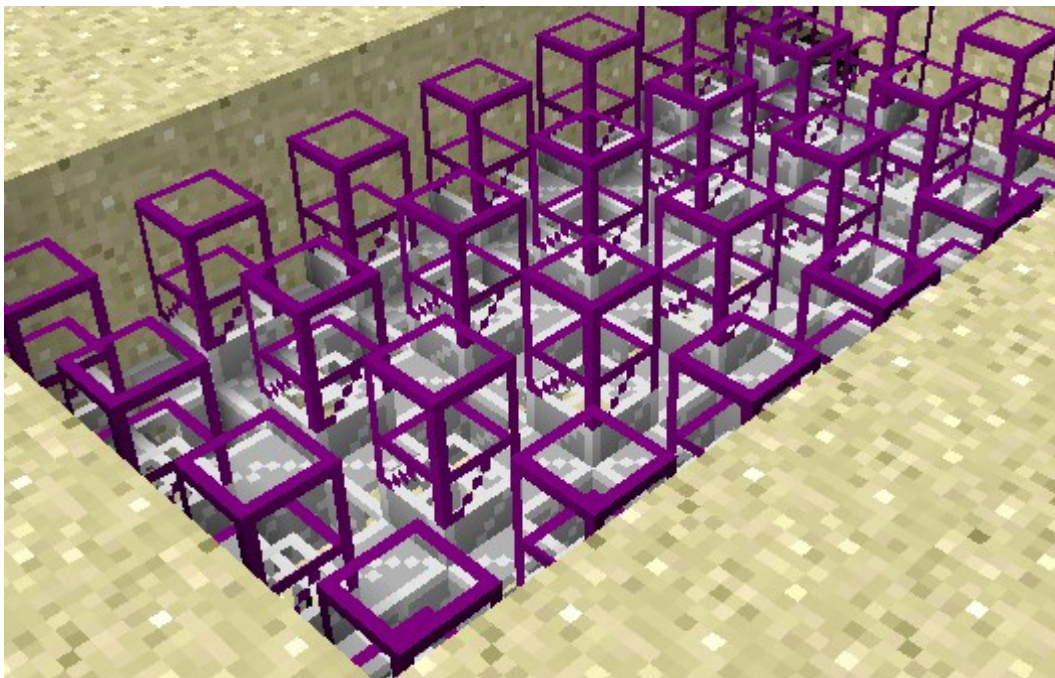

Illustration 11: Vacuum floor with obsidian and iron pipes

When powered, obsidian pipes will be able to suck objects from the minecart right in front of them, one MJ per item.

Illustration 12: Open junction with obsidian pipes

Obsidian pipes cannot be connected together. When an item gets out an obsidian pipe, it will not be sucked again for some time.

9   *Cobblestone Waterproof Pipe*

This is the base of liquid pipe system. Doesn't have any particular property except for the fact that it cannot be connected to a stone pipe.

10   Stone Waterproof Pipe

   As for the stone transport pipe, the stone water proof pipe is similar to the cobblestone waterproof pipe, except that it cannot be connected to cobblestone pipes.

11 Wooden Waterproof Pipe
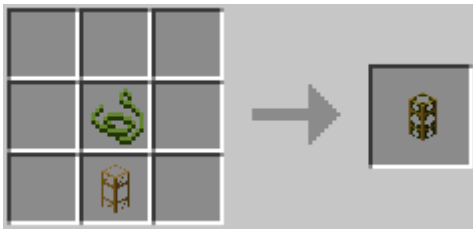
Power Properties	
Latency	50
Minimum of Energy Received	1
Maximum of Energy Received	64
Maximum of Energy Stored	64
Minimum Energy for Activation	1
Energy Loss per Cycle	64

The wooden waterproof pipe has the ability to extract liquids out of liquids containers such as tanks or refineries. Each time it receives power, it will extract one bucket out of the container.

12   Iron Waterproof Pipe

Similar to the regular iron pipe, this pipe will direct liquid to a certain direction depending on its transparent end. This direction can be switched using a wrench.

13 Golden Waterproof Pipe

Like golden transport pipes, golden waterproof pipes accelerate the travel of objects. However, the acceleration is only effective while liquid is in the pipe. Unlike transport pipes, golden waterproof pipes can be connected together, and do not require redstone to operate.

Golden waterproof latency is reduced to 4 and flow rate increased to 80, making them suitable when fast liquid delivery is important, for example as an engine coolant dispenser.

14 Wooden Conductive Pipe

When place above an engine, this pipe has the ability to gather its power, and convert it into a power stream to be send into a conductive pipe network. Only the conductive end of the engines can send power to these pipes. Power can then be gathered through the system.

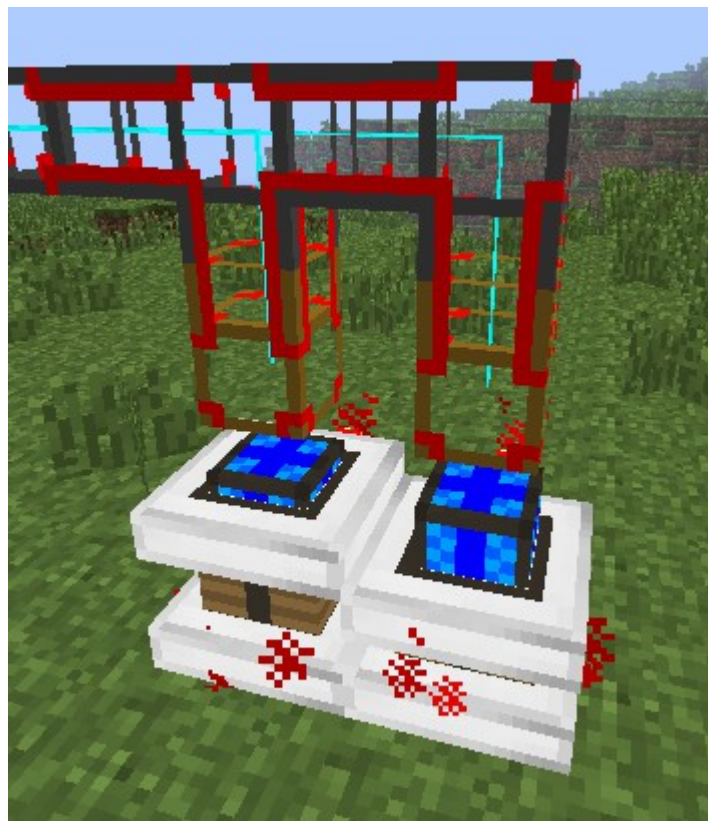Like the other kinds, conductive wooden pipes cannot connect to one another.
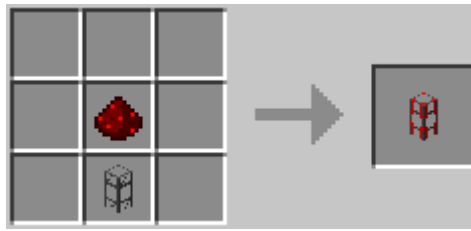

Illustration 13: Engines sending power

15   Stone Conductive Pipe

This is the base of a power network. Properties of these conductive pipes are the default ones. Note that still there is a 1% power loss per pipe, this is appropriate for small circuits but not to transfer power on long distances.
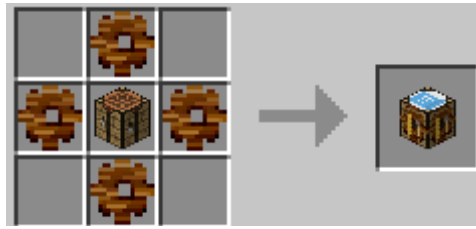
16   Golden Conductive Pipe

These pipes have a much better ratio when transporting power. Their power resistance is only of 0.1%, so a tenth of regular pipes, which allow them to efficiently transport power on a much longer distance.

6 Factory Mod Contents

1   *Automatic Crafting Table*

From the player interaction point of view, the automatic crafting table acts like a regular crafting table, except that it stores objects. So it is possible to exit the table without dropping the contents that were in the process of being crafted. However, leaving a working recipe in the table will allow pipes to activate the automating crafting mechanisms and pull out objects.
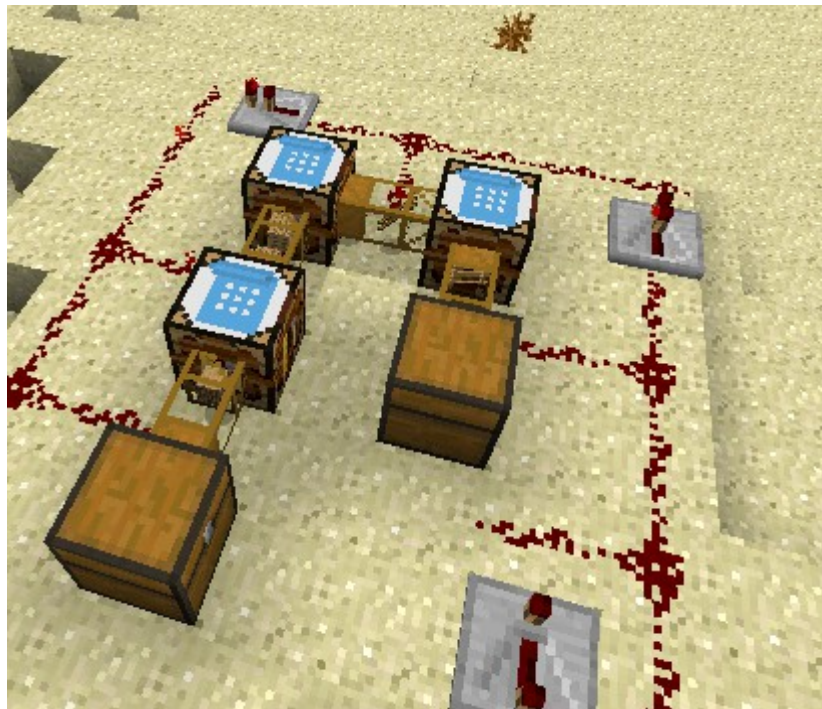

Illustration 14: Compact ladder craft chain

Automatic crafting tables are special inventories. Pipes can only place object on top of items stacks that are already there, in order to preserve the recipe. When wooden pipes extract objects out of chests, they never extract the object in the inventory. Instead, if all stacks contains at least two objects, they will decrement each stack, and extract the recipe results only. Wooden pipe extraction will make sure that there is always one item of each left in the recipe, in order to keep it.

If there is not enough item in the crafting table inventory, the automatic crafting table will look at nearby inventories to get items to craft. It will not

consider special inventories, like other automatic crafting tables. This is particularly useful when the crafting recipe is based on non-stackable items. In this case, the only option is to provide adjacent chests with items to be used in the crafting recipe, like bucket for example.

2   Mining Well

Power Properties	
Latency	50
Minimum of Energy Received	25
Maximum of Energy Received	25
Maximum of Energy Stored	25
Minimum Energy for Activation	1000
Energy Loss per Cycle	0.01

Mining wells are devices allowing to dig vertically in the ground. They will extract block underneath one by one, replacing them by a special kind of pipe that cannot be crafted manually. They will consume 25 MJ per block to remove. Digging will go down until bedrock or lava is encountered. After being placed, mining wells can be dug by e.g. an iron, a golden or a diamond pickaxe. Using any other kind of item will break the well.

Illustration 15: Mining well digging

If there is an inventory adjacent to the well, like a chest, objects will be placed into the inventory in priority. In there is no inventory, or all inventories are full, then the mining well will try to place items in a adjacent pipe. If there's no pipe either, the dug item will be thrown in the air.

In principle, mining wells should be used after some time of manual digging, but before getting into some more serious mining with a quarry. Their main advantage is that they can dig very fast to bedrock, hence extracting valuable ores

in small quantities quickly.

3   Quarry

Power Properties		
Latency	20	20
Minimum of Energy Received	25	20
Maximum of Energy Received	25	200
Maximum of Energy Stored	7000	7000
Minimum Energy for Activation	25	20
Energy Loss per Cycle	0.01	0.01

The quarry is the most advanced mining mechanism in BuildCraft as for now. It is able to mine a large area on the ground, all the way to bedrock. The energy consumption depends on the quarry phase – while building, it requires 25 MJ to either remove or add a block. While digging, it can receive power between 20 MJ and 200 MJ, and will consume between 2 and 8 MJ per cycle, depending on how much power is currently stored in the machine. Like the mining well, block under lava or bedrock will not be mined, and only iron, golden or diamond pickaxe can pick up a mining well that has been placed on the ground.

*Illustration 16: A quarry sending objects in two chests*

By default, when placed on the ground, a quarry will define a 9x9 area where it's going to dig. During the first stage of operation, an orange frame will be build around, and the area withing this perimeter will be excavated. Building this yellow frame, or placing a block in this area will stop the mining process up until the quarry repairs itself. Once set, the quarry will create a drill which will slowly dig objects in the ground.

As for the mining well, if there is an inventory adjacent to the quarry, objects

will be placed into the inventory, otherwise in pipes, and lastly thrown in the air.

Quarries are area-sensitive. Placing them to a land mark selecting an area (see the next chapter) will define an new mining area, provided that its size is at least 3x3. The height of the frame will be at least 5 blocks, or more if the area defines a higher height.

4   Tank
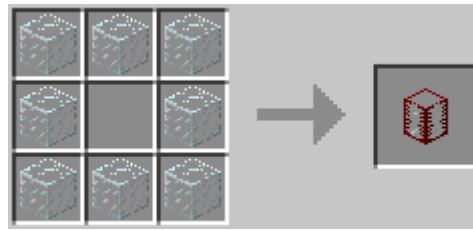
Tanks are used to store liquids. As of BuildCraft 2.1.0, they can only store oil, but will eventually be able to store any kind of liquid such as water or lava.

Each tank can store up to 16 buckets of liquids. The only way to put liquid in a tank is through a pipe, from any side. All pipes can send liquid to a tank. When powered, wooden pipes can extract liquid from tank, at a rate of one bucket per MJ.

Tanks can be place one of top of the other. When so, liquid arriving through any stacked tank will initially fill the bottom one, and any pipe extracting liquid from the tank will start from the top one.

Illustration 17: Oil moved from one tank to the other

When right clicking with a bucket containing liquids, liquid will be poured in the tank. If the bucket is empty, then it will be filled with liquid contained in the tank.

5   *Pump*
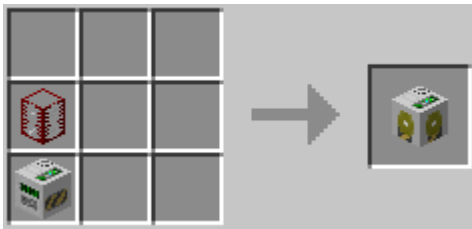
Power Properties	
Latency	20
Minimum of Energy Received	10
Maximum of Energy Received	10
Maximum of Energy Stored	100
Minimum Energy for Activation	10
Energy Loss per Cycle	0.01

Pumps are "upgrades" for the mining well, allowing the machine to extract liquids underneath. As of 2.1.0, pumps can only extract oil.

Pumps have the ability to extract an entire contiguous volume of liquids. It will extract layer by layer, sending the tube down to extract deep deposits. A bucket of liquid can be extracted per 10 MJ provided to the pump, which will then be sent to the adjacent pipes. Several pipes placed at the output of a pump will increase its global flow.
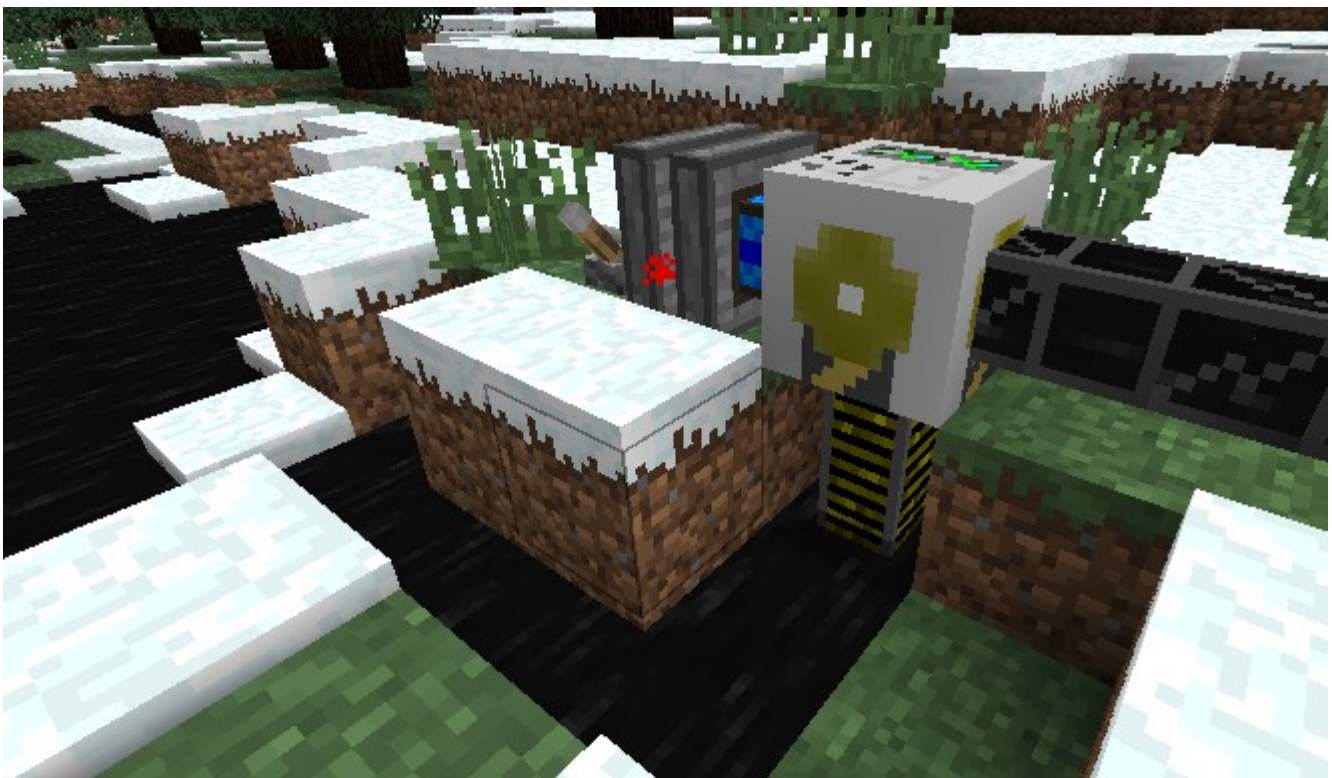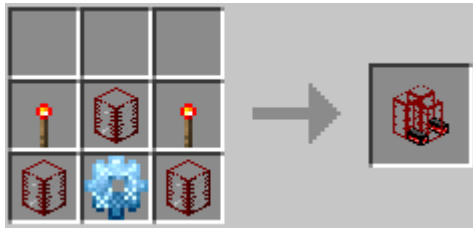

Illustration 18: A pump operating on a oil deposit

6 Refinery

Power Properties	
Latency	20
Minimum of Energy Received	25
Maximum of Energy Received	25
Maximum of Energy Stored	1000
Minimum Energy for Activation	25
Energy Loss per Cycle	0.01

The refinery is a new crafting block able to produce new liquids out of various sources. It can be filled either by right clicking on it with a bucket, or through waterproof pipes, and will require energy to operate the transformation. The amount of power required, and the time needed to convert input to outputs depends on the actual recipe.

Here is the table of various things that can be produced in the refinery in BuildCraft as of 2.2.0:

Source 1	Source 2	Energy	Time	Result
1 mB Oil	-	10	1 cycle	1 mB Fuel

You will probably notice how small the list is for now. Stay tuned for new recipes coming with next releases!

7 Builders Mod Contents

1   Land Marks

Land marks are special redstone torches that allow to define a line, a 2D or a 3D area in the world. They can be placed on blocks like torches, including ceiling blocks. When right-clicked, a mark will look at all directions to find an other mark, up to 64 blocks. If it does, and if that mark is available, if will use that mark as a line selection, identified as a red laser. Connecting this mark to an other one on an other axis will define a square, and if connected to yet an other one on the last remaining axis, it will define a cube. When a mark of an area is removed, the entire area selection is canceled.

Illustration 19: Ways of placing land marks

When defining large areas, it's often difficult to place a mark straight on the correct axis. For that reason, when powered by DC, marks will send a blue laser, showing the directions where a complementary mark can be placed. The blue laser will be removed either when the mark gets unpowered, or when replaced by a red laser.

Marks are used to define areas in BuildCraft for certain machines, such as the quarry, the filler or the template drawing table. In order to take into account the area selection, the machine has to be placed right next to one of the marker defining the area. When doing so, the markers will be dropped, and the red laser will be replaced with stripes, showing the area of effect.
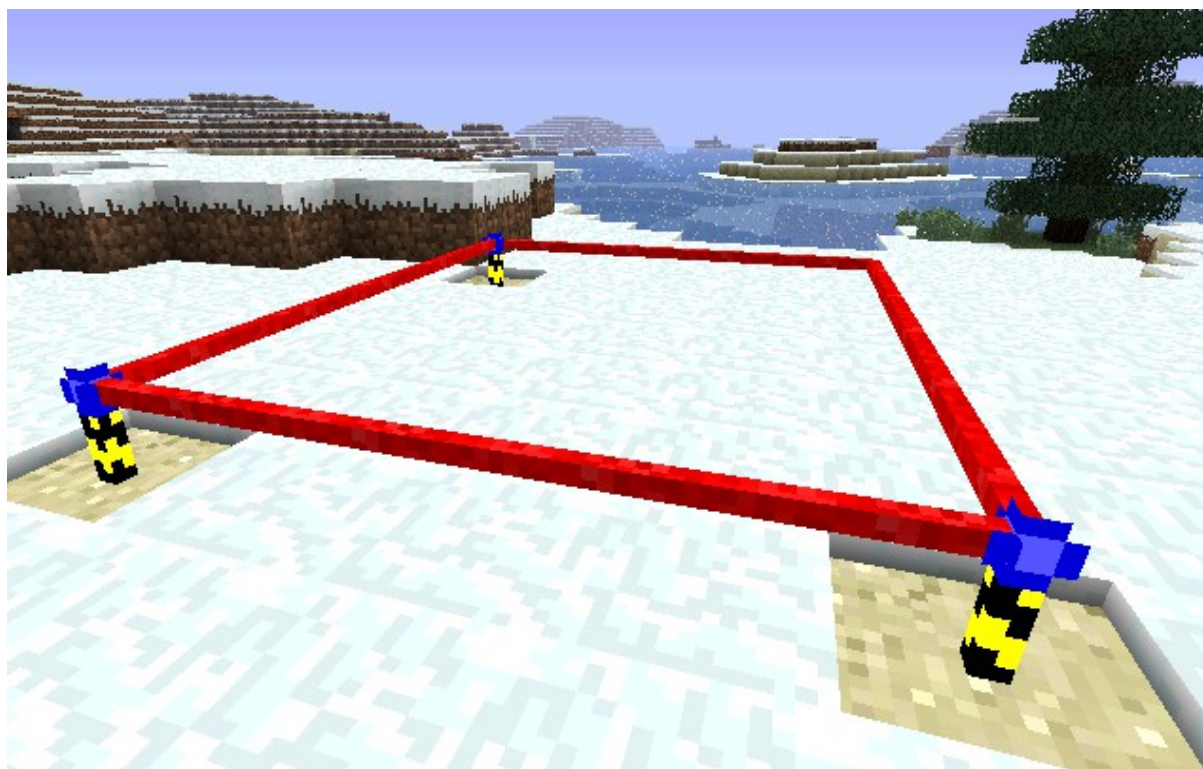

Illustration 20: Square defined by three land marks

2   Filler

Power Properties	
Latency	10
Minimum of Energy Received	25
Maximum of Energy Received	25
Maximum of Energy Stored	25
Minimum Energy for Activation	25
Energy Loss per Cycle	25

The filler is the first actual building block. It allows players to automatically fill an area, using a specific pattern. When placed next to a marker part of an area, it will take that area as the area to fill, replacing the red lasers with stripes. Each block to remove or add will require 25 MJ.

Fillers is a special inventory, containing both a "pure" inventory part, at the bottom, and a recipe part, at the top. This recipe is connected to a special crafting book, allowing to define filling pattern. At the moment, these recipes are based on brick and glass. They do not actually create items, but define the way the filler work, and will update the pattern display on the sides.

The filler will use blocks located in the lower end of its inventory to fill the area. If there's already blocks correctly located, they will be left as-is. Depending on the recipe, the filler may or may not remove blocks that don't fit in the pattern. Once done, the green indicator on the top turns red, but the stripes stay, allowing to place an other recipe for an other filling operation.

Full Filler

This pattern will fill the entire area with the given blocks.

Clear Filler
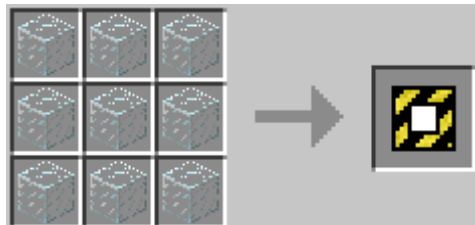
This pattern do not use blocks per see. It will however remove all blocks that are within the defined area, dropping the corresponding items.

Flattener Filler
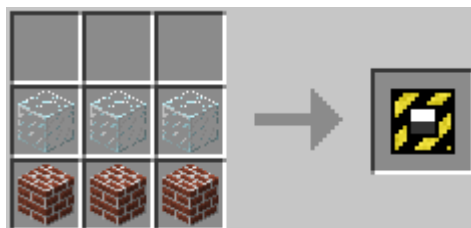
This pattern is the only one working outside of the area defined in by the stripes. It will fills all block below the stripes area, and then remove all blocks from the bottom of that area to the top of the world.

Walls Filler

This pattern will fill only the sides of the area (walls, floor and ceiling), leaving the center untouched.

Pyramid Filler

This pattern will fill a pyramid on the area, leaving the other blocks untouched. The way the pyramid is built depends on where the filler is placed – when on the bottom of the area, it will go from the bottom to the top. If the filler is placed at the top of the area, the a reverse pyramid will be built, from the top to the bottom.

Stairs Filler

This pattern will fill the area with stairs, depending on when placed relatively to the markers. If placed on the same level as the marker, it will create regular stairs, starting from the level where the filler is placed (top or bottom of the area). If placed underneath or above a marker, it will create radius stairs.

Illustration 21: Radius stairs, from the top of the area

3   Template Drawing Table

The template drawing table is a device allowing to define a precise pattern to fill. When placed to an area, it will replace the red lasers with stripes. Whatever blocks in that set of stripes will be captured as the pattern.

Once set, a template has to be placed on the top left slot of the inventory. This template will then be used to write the template, and used later on. Each template gets assigned a number. If no block has changed, creating twice a template out of the same template drawing table will issue the same template, with the same number.

Illustration 22: Template made with dirt

When powered by DC, the template drawing table will inverse the pattern set in the area, recording a plain block for every block of air and vice-versa.

There is a restriction currently that forbids templates numbers to go beyond 65 025. When reaching that limit, it is possible to remove old templates by going to the .minecraft/buildcraft/bpt, and remove the old .bpt files. Note that if a template is associated to a removed number, it will not be usable anymore.

The orientation of the template can be changed with a wrench.

4   Builder

Power Properties	
Latency	10
Minimum of Energy Received	25
Maximum of Energy Received	25
Maximum of Energy Stored	25
Minimum Energy for Activation	25
Energy Loss per Cycle	25

The builder blocks act much like the filler block, in the sense that it will fill an area with the blocks it's provided with. The pattern, however, isn't defined from an recipe, but from a template coming out of a drawing table. As a matter of fact, the filler doesn't need to be placed next to an area. The area of construction will be automatically set when receiving a template in the top part of the inventory. As for the filler, each block added or remove requires 25 MJ.

Builders will place and remove blocks so that the shape set by the template is respected. Once done, it will remove the stripes around it.
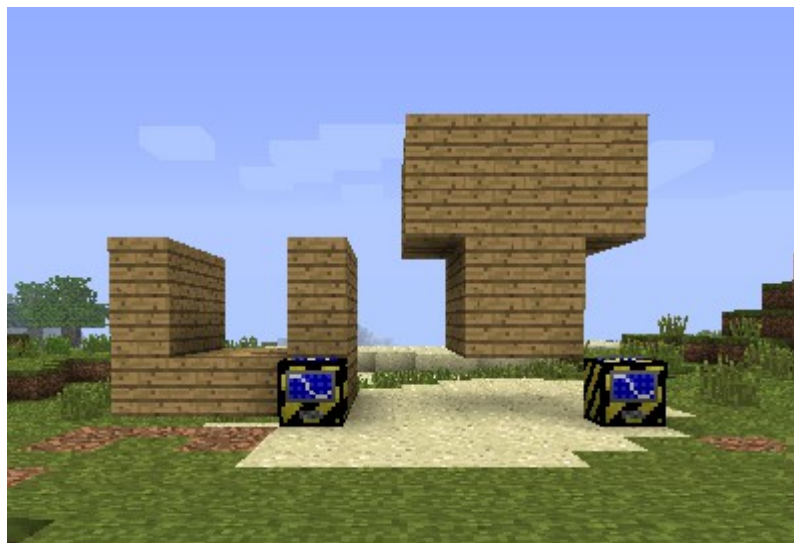

Illustration 23: Building from template and anti-template

The orientation of the builder can be changed with a wrench.

8 Energy Mod Contents

1 Introduction

As for now, this Energy mod and its contents are in beta mode, in order to activate its capabilities, the power framework has to be set as follows in the buildcraft.cfg file:

power.framework=buildcraft.energy.PneumaticPowerFramework

Engines provided by this mod allow to provide more power to devices than the legacy redstone power mechanism. However, most of them require additional material to operate.

All engines need to be activated through redstone power. When they are, they will both smelt the fuel that they're using, if any, and release their power to the nearby device. When unactivated, they will stop releasing the power. However, certain engines cannot stop the smelting process right away, and will carry on generating energy for a little while. Note that when an engine is activated, it will start generating and storing energy, whether or not it is connected to a device to release it.

When an engine is generating more energy that consumed, it stores the extra amount. When reaching a certain threshold, this will accelerate the mechanical part, accelerating the energy release. Engines have four level of internal energy, blue, green, yellow and red, which are visible by the color of the central area. These levels correspond to different quantity of energy depending on how much total energy the engine is able to store. When not powered with redstone, engines will progressively cool down, at the rate of one MJ per cycle.

If the power stored by an engine goes beyond the maximum limit, the engine will explode. The range of the explosion depends on the kind of engine.

When placed on the world, engines are automatically oriented towards a device requiring power, if any. This orientation can be switched, if there are other nearby devices, by using a wrench.

Engine can transmit power to other engines, allowing to create a chain that ultimately provides more power to the final device. However, in the process or transmission, there is a 5% loss of power.

Illustration 24: Four engines around one quarry

Note that engine ignore the latency property of devices.

BuildCraft engines are associated with all the regular power receptor properties, plus additional ones:

Energy Stored per Cycle

Amount of energy stored by the engine per cycle when activated.

Maximum of Energy Released

Maximum amount of energy that can be released by an engine to a device.

2   Mechanical Engine

Power Properties	
Latency	1
Minimum of Energy Received	1
Maximum of Energy Received	50
Maximum of Energy Stored	1000
Minimum Energy for Activation	1
Energy Loss per Cycle	0
Engine Properties	
Energy Stored per Cycle	0.05
Maximum of Energy Released	1

Mechanical engines are the least powerful engines available in the game. They can only release 1 MJ per contact, and need 20 cycles to store one MJ of energy. For this reason, they cannot transmit power to an other engine, because the transmitted power would be below 1 MJ.

However, mechanical engine only require redstone activation to work. For this reason, they are very suitable to operate simple devices such as wooden or obsidian pipes.

3   Steam Engine

Power Properties	
Latency	1
Minimum of Energy Received	1
Maximum of Energy Received	200
Maximum of Energy Stored	10000
Minimum Energy for Activation	1
Energy Loss per Cycle	0
Engine Properties	
Energy Stored per Cycle	1
Maximum of Energy Released	100

Steam engines are much more powerful than redstone engines. They require however material to smelt in order to store energy and operate. The smelting for items is the same as for the furnace. For example, coal last 1 600 cycles. A stack of 64 coal will last about 1 hour and 20 minutes.

Here are details on power provided by various things that can be fed to the engine:

Item (1 unit or 1 B)	Total Energy (MJ)	Time (cycles)
Coal	1,600	1,600
Lava	20,000	20,000
Stick	100	100
Wood	300	300

4   Combustion Engine

Power Properties	
Latency	1
Minimum of Energy Received	1
Maximum of Energy Received	2000
Maximum of Energy Stored	100000
Minimum Energy for Activation	1
Energy Loss per Cycle	0
Engine Properties	
Energy Stored per Cycle	variable
Maximum of Energy Released	1000

At this stage in the game, the combustion engine is the most powerful of all. It requires some kind of fuel to operate, and can store up to 10 bucket of it. For example, a bucket of oil will last 10 000 cycles, which makes it around 8 minutes of Minecraft time. An engine can be filled with up to 10 bucket of oil will last about 1 hour 20 minutes of work.

Here is the list of all different kind of materials that the iron engine can use to produce energy:

Item (1 unit or 1 B)	Total Energy (MJ)	Time (cycles)
Fuel	600,000	20,000
Lava	20,000	20,000
Oil	200,000	10,000

Like other engines, iron engines warm up over time. Their warming behavior is that each time a unit of energy is produced, a unit of heat is stored, which makes warming up potentially much more important than on e.g. stone engine which only warm up on unused power.

Iron engines can be cooled down using water. Water can be filled either using a bucket, or through pipes connected to the engine. The destination – fuel or coolant tank – will be selected automatically.
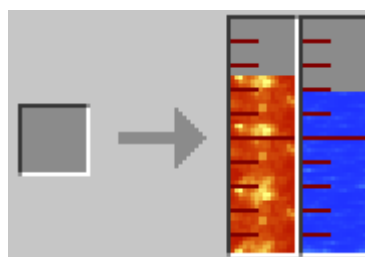
Illustration 25: An engine powered by lava and cooled by water

One mB of water can cool down the warming that has been produced by 1 MJ of energy.

5   Oil

The combustion engine is the most powerful, it's also the one that relies on one rare resource oil. Like laval, oil is a finite resource and getting an oil source in a bucket will remove that source from the world.

Small oil deposits will get generated randomly on the surface of the desert biomes, up to 16 units of oil. The probability for a small deposit of oil to generate on a desert chunk is 3%, which makes it a likely event. Large deserts will typically contain lots of these.

Illustration 26: A small oil deposit in the desert

However, to sustain an industry based on oil resource, more oil is likely to be required. Medium deposits containing between 100 and 1000 buckets of oil may appear on any chunk with a probability of 0.1%. Massive deposits containing between 2 000 and 17 000 buckets of oil are spread randomly around the world as well, with a generation probability of 0.005% (which makes it an event that has 95% of happening on a 1000x1000 blocks surface).

Medium and large deposits consist of an oil bubble which center is located between depth 20 and 30. They can be spotted from the surface through an oil deposit looking like a regular deposit, except that it can be located on any biome, and contain a chimney going from to the actual bubble up to a geyser when reaching the air.

Illustration 27: Medium size oil deposit

While such massive oil deposits might seem overkill at present, if you're lucky enough to find one, mark its location and keep it! There's lot power and oil usage to come for further versions of BuildCraft.

6   *Fuel*

Fuel is produced by refineries (from the Factory mod) when processing oil. It does not appear naturally in the world, nor can be dropped as other liquids. It only use so far is to feed iron engines with something more efficient than crude oil or lava.

9 Configuration

1 Buildcraft Folder and buildcraft.cfg file

All BuildCraft-specific content is located under .minecraft/buildcraft. The file .minecraft/buildcraft/config/buildcraft.cfg contains a property file allowing users to adjust the gameplay or fix other plugin compatibility issues. This file is read and resynchronized at each launch of Minecraft. It contains three sections, "general" for setting related to the general gameplay, "block" for settings related to blocks (mainly block ids) and "item" for settings related to items (mainly item ids). In case of SMP, client and server need to have the same configuration file.

Worlds created with BuildCraft depend on information contained in the .minecraft/buildcraft, in particular templates and configuration properties. To share a world with an other user, it is important to share the entire buildcraft folder as well.

Note that worlds created with a version of BuildCraft prior to 1.6.6.4 will have their legacy BuildCraft.cfg file automatically imported and removed.

2 General Settings

current.continous

Can be either "true" or "false". Allow to enable the cheat mode, where only continuous current is required to have machines working. This setting is only used along with the legacy redstone power framework.

mining.enabled

Can be either "true" or "false". Allow to deactivate auto-mining machines.

modifyWorld

Can be either "true" or "false". Allow to deactivate buildcraft-specific world generation things, such as oil deposits.

powerFramework

The current power framework activated in the game. By default, BuildCraft allow to select between two different ones:

buildcraft.core.RedstonePowerFramework

buildcraft.energy.PneumaticPowerFramework

pipes.alwaysConnect

Can be either "true" or "false". When "true", enables pipes connections rules implemented until 1.6.6.3 where all pipes could be connected to any other kind of pipe.

3   Block Settings

<block>.id

All blocks are listed in the form "<block name>.id". When the file buildcraft.cfg is generated, all blocks are provided with a default value. If, at the time of the generation, there is already one block at the default id location, BuildCraft will automatically compute the next available slot.

This mechanism allow to avoid block conflict most of the times. However, when such a problem arises, it is possible to modify the block ids manually.

Note that as soon as a world has been created with a set of block ids, these ids have to stay unchanged, otherwise the world will not be loaded properly. It's wise to save your versions of buildcraft.cfg from one update to the other.

woodenPipe.exclusion

This is a comma-separated list of blocks ids not to be considered for pipe extraction. Can be useful to prevent e.g. extraction from private chests in SMP, when the developers of the private chest did not take into account BuildCraft pipes. The list can be made with internal block names as well, usually provided by the developers, in the form of "tile.<blockname>".

4   Item Setting

Similar to blocks, items ids are listed in the form "<item name>.id". Although conflicts are less likely to happen, given the number of available ids, it's still possible to modify then through the buildcraft.cfg file.

5   Blueprints Folder

The folder .minecraft/buildcraft/bpt contains blueprints for template, named <template number>.bpt. The total number of templates that can be managed by the game cannot exceed 65 025, shared between all the worlds created in the current Minecraft installation. The player can remove manually files that he knows are not in use anymore.

10 Recipe Book