

# Ancient Warfare 2

MAN YOUR DEFENCES

MAN YOUR MACHINES



## General Information

Ancient Warfare is a modification for Minecraft that adds a variety of game enhancements; new blocks, new entities, and a template based world-generation structure generator.

Beginning with the 2.x releases Ancient Warfare has been split into a multi-module release. There is a single Core module that all other modules rely on for shared functionality. Each other module may be installed individually, or in any other combination desired. The non-core modules are – Automation, NPCs, Structure Generation, and Vehicles. Each of these modules focuses on a single aspect of the mod and may be used in any combination.

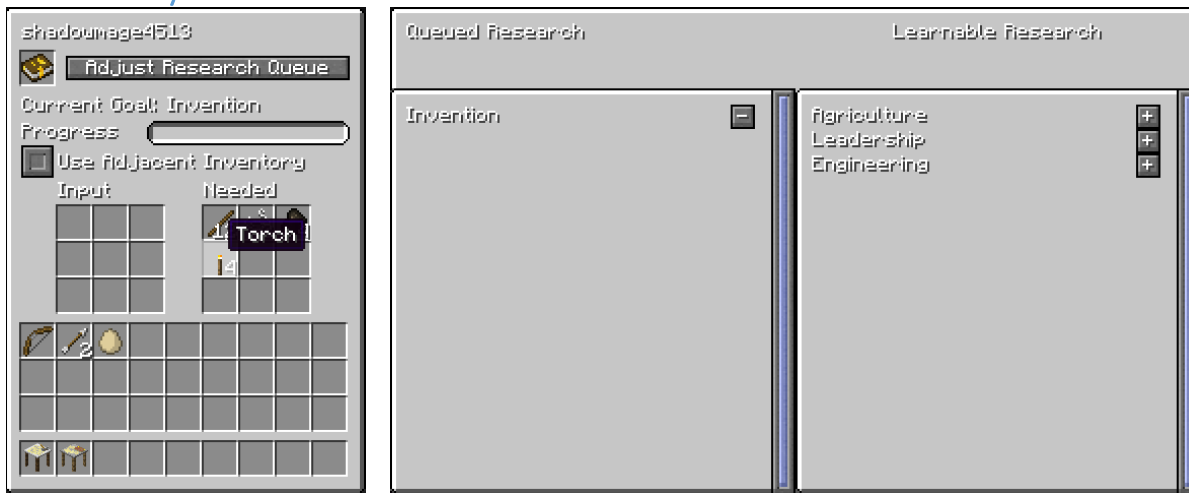
## Section I - Core Module

### General

The Core module in Ancient Warfare offers very little functionality on its own. It merely contains the framework that the rest of the modules use to communicate between one another, and any code shared amongst multiple modules. The Core module is mandatory if any of the other modules are to be installed.

The Core module adds the Research Station and Engineering Station blocks, the Research Book, Backpack, Quill (research), and Hammer (work/block-rotator/weapon) items. It additionally contains many of the frameworks and systems used by the rest of the modules, such as the research and crafting systems and improved item use/interaction systems.

### Research System



Research Station GUI

Research Queue GUI

The Research Station may be crafted in a normal vanilla workbench. For the precise recipe, see NEI or similar. Once crafted, it may be placed into the world like any other block. It has a single externally-accessible inventory – the slots labeled as ‘Input’.

To begin researching, place your research book into the top-left slot (below the name) (research book is crafted in a normal crafting bench). Press the ‘Adjust Research Queue’ button to view the research-queue GUI, where you can add or remove goals from your research queue. Resources needed for your current goal (if any) will be shown in the slots labeled ‘Needed’. Place your resources into the slots labeled ‘Input’. Once all resources have been met, the resources will be used and the current goal will begin researching. Research progress will continue for as long as you have the GUI open, but will cease as soon as you close the GUI. You must be present at / have the GUI open on one of your research stations, or use a researcher NPC in order to make research progress.

Once a particular goal has been completed, the goal at the head of the queue will be moved into the ‘active goal’ position. As long as the required resources are already present in the ‘Input’ boxes, the goal will start, and research will progress on the new goal. A research goal in the ‘active’ position may only be cancelled if it has not yet been started (go to the research queue, and remove it from the queue).

Goals that may be learned but are not yet queued will be shown on the right-hand side of the Research Queue GUI. Goals that are currently queued and not yet in-progress will be shown on the left-hand side of the GUI.

Goals may be added to the research queue by pressing the '+' button located to the right of the goal name. Goals may be removed from the research queue by pressing the '-' button located to the right of the goal name. In order for a goal to be added to the queue, any pre-requisite goals must already be learned or queued. If a goal is removed from the queue that has other goals dependent upon it, those goals will also be removed from the queue.

The research station may be optionally configured to use adjacent inventories to withdraw resources for research goals. To enable this functionality, check the checkbox marked 'Use Adjacent Inventory'.

Other notes:

- Most recipes in Ancient Warfare require research before they are craftable. The entire research system may be disabled in the core configuration file – in which case, no research will be required and all Ancient Warfare recipes will be craftable in the normal workbench.
- Research is done only at the Research Station, which may be crafted in the normal workbench or in an Ancient Warfare engineering station.
- Research requires a Research Book, which is also craftable in the normal workbench.
- Research is stored per-player. Any player may contribute to any others' research by obtaining a copy of their research book and using that book in any research table.
- You may use other players research for crafting by obtaining a copy of their research book and using it in your Engineering Station.
- You may only progress on a single research goal at a time. All research stations with a copy of your research book will all contribute towards your selected goal.
- Research goals have both a time and materials component. The needed materials may be viewed on the tooltip of the goal in the research-queue GUI, or in the 'needed' box on the main research progress GUI. Once these materials are present in the 'input' box, the materials will be used and research will automatically start progressing.
- In order to make progress on a research goal you must do one of three things:
  - Leave the research station GUI open.
  - Assign an NPC researcher to work at the station.
  - Right click on the research station with a 'Quill' item equipped.
- Multiple Research Stations may be setup to make faster progress towards a single goal. They will each need researcher NPCs assigned to them, or have players using quills / viewing their GUIs.
- The Research Book may be right-clicked after it is assigned to view a GUI with details about each research goal and researchable recipe.

## Crafting System



Engineering Station GUI

Crafting in Ancient Warfare 2 has been simplified from the prior versions to use a single crafting block. All Ancient Warfare and vanilla recipes may be crafted at the Engineering Station. The Engineering station may be crafted in a vanilla crafting table. See NEI or similar for the recipe.

Once crafted and placed in the world, the Engineering Station crafting GUI may be accessed via interacting with the block (default: right-click). The Engineering Station has a Research Book slot, and a Research Book must be placed into this slot in order to craft any recipes that have research requirements – the recipes available will depend upon the completed research for the player owning the Research Book. If no Research Book is in the slot the Engineering Station will only be able to craft vanilla recipes or the few Ancient Warfare recipes that do not have research requirements.

To craft using the Engineering Station, lay out your recipe in the crafting grid as normal. Results may be removed from the output slot one at a time, or shift-clicked to craft as many as possible (placed directly into player inventory). The additional slots below the layout grid can be used for extra raw materials storage. The crafting-grid will auto-refill from these components if possible. All slots in the Engineering Station are persistent – unlike the vanilla crafting bench they will not be ejected into the world when the GUI is closed; they will still be there whenever you return, enabling you to grab forgotten materials or extra materials without losing your recipe layout.

## Crafting Recipes



[NEI Recipe lookup for researched recipe](#)

To view crafting recipes, use NEI or other similar mod. As the recipes may change at any time (and some are user-configurable), they are not listed here.

There is a built-in NEI plugin that should be compatible with recent releases of NEI. It adds recipe lookup and item use search capability for all researched recipes. It does not currently list the required research for viewed recipes (that capability is planned for a future update).

## Teams

Ancient Warfare 2.x versions use the vanilla Scoreboard system for team management. Teams may be created/joined/left by using the vanilla scoreboard chat commands. Please see the Minecraft documentation for a list of these commands.

Most entities and tile-entities in Ancient Warfare are owner and team-aware. These entities and tiles store a reference to their owners **name** (not UUID), and use this reference to look up their owners team. Any time a player switches team, **all** of his owned entities (npcs/vehicles/gates) and tiles (crafting blocks, worksites, others) that are team-aware will switch team with him.

Any assigned team colors will be used to display the nameplates for entities owned by other teams (npcs/vehicles) (only applicable if enemy nameplates are enabled in the client-side configuration).

Interaction with many entities and tiles is limited by owner and/or team. You will not be allowed to interact (open GUIs, send commands, etc) with entities or tiles that you do not own or are not on the same team as the owner.

## Options Menu and Keybinds

In game options and keybinds may be adjusted through the Forge Config GUI system. To access these GUIs you must be at the main menu (before joining a world). Click on the Mods button to examine the mods list. Scroll down to the Ancient Warfare Core entry and select it. Below the scroll area a 'Configuration' button should become selectable. Click on this button to access the Ancient Warfare in-game configuration options.

Each module that has configurable options will have one or more configuration sections listed in the main Ancient Warfare configuration screen. Click on the individual buttons to see the configuration options for that module.

Keybind options use the MC / LWJGL key-code mappings. These may be found at: [http://minecraft.gamepedia.com/Key\\_codes](http://minecraft.gamepedia.com/Key_codes) . These keybinds may share assignments with vanilla / other mods keybinds – conflict resolution is up to the user.

## Configuration Files

Each module has its own configuration file(s) for module-specific options. The core module also has its own configuration file for global/shared options. Please pay attention to the default values and comments in the configuration files, as these should explain everything necessary to make adjustments. All configuration files for the mod are located in the /config/ancientwarfare/ directory in your Minecraft folder.

More configuration options will be added to these files as development continues. If there is a specific option you would like added, please open up an issue ticket on GitHub including information regarding the option you would like added (see the Features Requests section for more information).

## Item Use

Most useable items have been reworked to use a key-driven activation scheme in addition to the existing right-click action. This additional interface is needed due to the complexity of several of the items, as well as for a workaround for item/block interaction conflicts (e.g. cannot right-click activate an item when a block has a right-click action).

The keys used for item interaction can be changed/reassigned in the Options->Keybinds GUI. The default assignments for these keys are 'Z', 'X', 'C', 'V', and 'B'. Any particular item may use one or more of these keybinds; a few of the more complex items from other modules use all of them. Many actions for items that previously used right-click will now use item-key 1 (default 'Z'). Please see the tooltip on specific items to see what key is bound to which action for that item.



Example tooltip for an item with key based interaction

## Chat Commands

Ancient Warfare adds a few new chat commands for manipulation of advanced game data. These commands are only available if the module they interface with is also installed (e.g. structure command needs structure module).

`/awresearch <player_name> <add/remove/fill/clear> [research_to_add/remove]`

This command allows for manipulation of a players' research data. You may add techs, remove techs, or completely reset a players' research progress.

To add research to a player the syntax is `/awresearch add <player_name> <research_name>`.

To remove research from a player the syntax is `/awresearch remove <player_name> <research_name>`.

To complete all research for a player the syntax is `/awresearch fill <player_name>`.

To remove all research from a player the syntax is `/awresearch remove <player_name>`.

`/awstructure <delete/build> [<structure_name> <x> <y> <z> <face>]`

This command allows for construction of a structure through the chat interface. It may be used in command blocks for scripted structure creation. It is only available if the Structures module is installed.

To delete a structure from the available structures list, the syntax is `/awstructure delete <structure_name>`.

To build a structure from the loaded template list, the syntax is `/awstructure build <name> <x> <y> <z> <facing>`. Valid facing directions are 0-3 (south, clockwise through east).

`/awfaction <set/setall/get> <player_name> [faction] [amount]`

This command allows for viewing and manipulation of a players' faction standing with the NPC factions. It is only available if the NPCs module is installed.

To view a players standing with all factions the syntax is `/awfaction get <player_name>`.

To set a players standing with a single faction the syntax is `/awfaction set <player_name> <faction_name> <new_standing_value>`.

To set a players standing with all factions the syntax is `/awfaction setall <player_name> <new_standing_value>`.

## Core Module Items

### Backpack



Backpack GUI

The backpack serves as a portable chest. It comes in four sizes – 9 slot, 18 slot, 27 slot, and 36 slot. Larger sizes are more expensive to craft but have no other additional features.



Right-click with a backpack equipped to open the Backpack GUI. From here you can move items into or out of the backpack. The inventory slot that the backpack occupies will be locked and unavailable for manipulation – this is to remove any chances of doing something silly (like putting the backpack inside of itself, thus deleting it when you close the GUI). To close the GUI, press ‘esc’ or ‘e’ (or whatever your inventory keybind is).

### Hammer



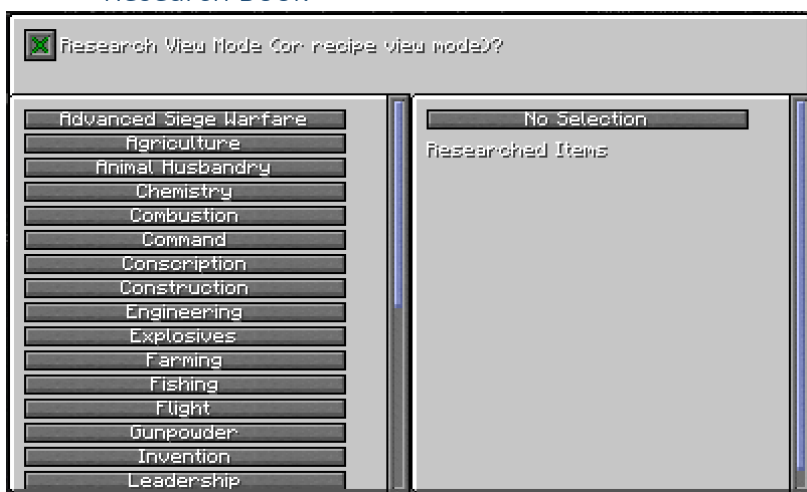
Hammer tooltip

The Hammer item has three functions: weapon, block rotation tool, and a tool for allowing players to work at Automation Module Worksites. To use it as a weapon simply attack an entity with it; it should have damage equivalent to a sword of the same material, but lacks a block ability. To use it as a block-rotator or work-tool, right click on the target block. To switch between ‘rotate’ and ‘work’ modes press item-interaction key 1 (default ‘Z’). The Hammer is also used as a class equipment item for some NPCs in the NPC Module.

### Quill

The Quill item may be used as a mid-grade weapon (slightly less than same-material sword), or as a tool to allow players to ‘work’ at a research station (similar to working at a worksite with a Hammer). To manually add research to a research station simply right-click on the block – one seconds worth of research time should be added to the progress of the currently researching goal. The Quill is also used as equipment for some NPCs in the NPC Module.

### Research Book



Research Book GUI



The Research Book serves to denote what player's research progress should be used in Research Stations, Engineering Stations (and other AW crafting stations). Exact research progress is not stored in the Research Book – losing your Research Book or having it destroyed will not result in losing any research progress.

After a Research Book is first crafted it must be assigned to a player to be of any use. Right click to assign a new research book to the player using it. After it has been assigned to a player, the Research Book can be used to view the Research goal tree as well as any recipes that are unlocked from specific research.

After a research book is assigned to a player, that player may right click to open the Research view GUI. The Research view GUI allows for viewing either a list of research goals and what they unlock, or a list of unlockable items and the research necessary to unlock them. Click the checkbox at the top to toggle between these modes.

Once a Research Book has been assigned to a player it may not be re-assigned. You may give a copy of your Research Book to other players to allow them to use your Research progress or allow them to contribute progress towards your current Research Goal.

### Research Notes

Research Notes items. These may be right clicked to either learn the denoted research directly, or add progress towards your current research goal (depending on if you already know the research from the note). These will be added to the dungeon-chest loot generation system in the near future, with configurable options for rarity/enable/disable. They are currently only available via the creative menu.

## Core Module Blocks

### Engineering Station

The Engineering Station is the main crafting block for Ancient Warfare. To interact/open the crafting GUI simply right-click on the block. See the Crafting System section for more details.

### Research Station

The Research Station is necessary for progressing on the Ancient Warfare Research tree. To interact/open the Research GUI simply right-click on the block. See the Research System section for more details on its exact use.

## Bug Reports / Issues

Please report any bugs or issues on GitHub. The issue tracker is located at: <https://github.com/shadowmage45/AncientWarfare2/issues>. Issues reported in any other way will be ignored. Please browse the existing tickets to ensure it is not a duplicate. If an issue exists that covers your problem feel free to add any additional info that you may have.

If your issue is not a duplicate, feel free to open up a new ticket. Please describe the problem in as much detail as possible. If it is crash related, include the crash report. If it is a repeatable issue, describe the steps necessary to duplicate it. All of this information will come in handy tracking down and solving the issue – the more information you provide, the quicker the issue can be found and dealt with.

## Feature Requests

Is there a new feature you would like to see added? Think the balance on something could use adjusting? Want more options for configurability?

The steps for feature requests are as simple as opening an issue ticket on GitHub. Navigate to <https://github.com/shadowmage45/AncientWarfare2/issues>, click on the 'New issue' button, and include 'balance' or 'enhancement' in the subject line. Fill in as much information as possible regarding the feature you would like. Screenshots, links, diagrams, etc. are welcome (and encouraged) for explaining your idea.

## Contributions

Ancient Warfare is an open source project. Contributions to the code are welcome. More specific details can be found at the source repository, located at: <https://github.com/shadowmage45/AncientWarfare2>.

## Section II – Automation Module



### General

The Ancient Warfare Automation Module adds many new blocks to the game with the intent of adding tools for automation of resource gathering, processing, and crafting. It also adds a fully RF (Redstone Flux) compatible power-framework to power the automation blocks. A few other utility oriented blocks are also included.

Many of the features of the Automation Module are intended to be used in combination with the NPC Modules ‘Worker’ type NPCs – these NPCs provide the energy for operation of many of the automation machine blocks. Alternative means of power-generation are available in the Automation Module allowing it to be used as a stand-alone/without the NPC module if wanted.

### Worksites

Worksites form the heart of the Ancient Warfare automation system. A Worksite is a block that requires energy input in order to process its work. Many different types of worksites and work exist. Each worksite is devoted to a single task or type of work. Many worksites have work-bounds that denote the area over which they have control / will work in.

In order to place a worksite, simply right-click with the worksite block in-hand. The worksite will be placed with default bounds if applicable. To adjust the work-bounds, open the GUI (right-click) and hit the ‘Bounds’ button at the bottom. Many of the worksites with bounds also have configurable target/work blocks; these blocks may be set in the ‘Bounds’ GUI as well; red=work target, black=not a work target. For non-configurable worksites, the blocks will always be black.

Most worksites can have various properties upgraded through the use of ‘Worksite Upgrade’ items. Please see the Worksite Upgrades section for more details.

Most worksites have configurable block side/inventory assignment. Open the main GUI and press the 'Inv' button to adjust the block side/inventory mapping. Each side of the block may be mapped to a single internal inventory area (or none). Multiple sides on the block may be mapped to the same inventory area.

Players may manually 'work' at a worksite by right-clicking on the control block with a Hammer equipped.

Worksites may be 'turned off'/disabled by applying a redstone signal to the control block.

Worksites must be powered by Worker NPC types. Each worksite has a specific work type, and the workers work-type must match that of the worksite. Workers add energy to a worksite at a set interval (configurable via config file). The amount of energy added by a worker depends upon the workers equipment and level for that work-type. Anytime the stored energy is greater than activation energy, the worksite will process a single block or item.

Alternatively worksites may be provided power through either the Torque power system or by any tile capable of emitting Redstone Flux. They will accept power from any side.

The following types of worksites are currently available:

#### Crop Farm

The Crop Farm plants and harvests Wheat, Carrots, Potatoes, Melons and Pumpkins. The blocks to be tilled / planted / harvested may be adjusted in the 'Bounds' adjustment GUI. Harvested resources will be placed into the upper inventory area. Seeds to be planted should be inserted into the middle inventory area. The lower area is reserved for Bonemeal, which will be used to speed the growth of any crops after tilling/planting has been done. Can use bounds, enchanted tools, and quality tools upgrades.

#### Tree Farm

The Tree Farm plants saplings and harvests logs from grown trees. The blocks to be planted and harvested may be adjusted in the 'Bounds' adjustment GUI. Harvested resources will be placed into the upper inventory area. Saplings to be planted should be inserted into the middle inventory area. The lower area is reserved for Bonemeal, which will be used to speed the growth of any saplings after planting has been done. Can use bounds, enchanted tools, and quality tools upgrades.

#### Mushroom Farm

The Mushroom Farm plants brown and red mushrooms as well as Netherstalk. The blocks to be planted may be adjusted in the 'Bounds' adjustment GUI (all non-planting blocks will be 'harvest' blocks). Harvested resources will be placed into the upper inventory area. Mushrooms/Netherstalk to be planted should be inserted into the middle inventory area. Mushrooms and Netherstalk both have special requirements that must be met for their planting and growth – low light for all and soul-sand necessary for Netherstalk. Meeting these requirements is the responsibility of the player/user. Can use bounds, enchanted tools, and quality tools upgrades.

#### Reed/Cactus/Cocoa Farm

The Reed Farm plants and harvests the 'odd' crops from vanilla Minecraft: reeds, cactus, and cocoa beans. The blocks to be tilled / planted / harvested may be adjusted in the 'Bounds' adjustment GUI. Harvested resources will be placed into the upper inventory area. Items to be

planted should be inserted into the middle inventory area. Any special requirements for the crops growth must still be met – jungle logs are needed for cocoa beans, sand is needed for cactus, and reed requires water nearby with either grass/dirt/sand to grow on. Can use bounds, enchanted tools, and quality tools upgrades.

### Animal Farm

The Animal Farm allows for the automated feeding and harvesting of vanilla livestock. The maximum number of allowed adult animals per type is configurable via the ‘Adv’ button in the main GUI. Harvested items are placed into the top inventory area. Feed items for the animals may be placed into the lower inventory area. Animals will only be fed if there are two adult and breed-ready animals available. Can use bounds, enchanted tools, and quality tools upgrades.

### Fish Farm

The Fish Farm provides an automated means for catching fish (all vanilla types) as well as ink. More water in and below the bounds of the fish farm will result in faster production. Can enable/disable the catching of fish and/or ink through the ‘Adv’ GUI. Can use bounds, enchanted tools, and quality tools upgrades.

### Quarry

The Quarry uses input energy to harvest blocks. By default the Quarry can harvest any block harvestable with a stone pickaxe. Upgrades are available to harvest harder blocks. The upper inventory area is where harvested resources will be placed. The Quarry will remove water source blocks, but will not harvest below lava. Can use bounds, enchanted tools, and quality tools upgrades.

### Warehouse

The Warehouse is an upgradeable mass-storage site. It requires a Warehouse Control block to set up the site, as well as storage and interface blocks for actual operations. The Warehouse can use bounds, enchanted tools, and quality tools upgrades. See the Warehouse section for more information on setup and use.

### Auto Crafting Station

The Auto Crafting Station is a worksite devoted to the automation of crafting. Lay out your recipe in the layout matrix. Provide materials in the raw materials area. For every work unit of power input, one item will be crafted and put into the output area. See the Auto Crafting section for more information on setup and use.

### Hand-Cranked Generator

The Hand-Cranked Generator provides a means of turning NPC and player based ‘work’ into a form of energy useable by worksites and convertible into Torque power or Redstone Flux. Its main purpose is to allow an interface between NPC workers and BuildCraft – can use workers to power any RF machines with the use of a Hand-Cranked Generator. It also allows a late-game means for players to consolidate and optimize their energy use and production for worksites. See the Torque Power System section for more details on setup and use.

## Auto Crafting



Auto Crafting GUI

The Auto Crafting Station is a Worksite dedicated to automating the crafting of items. It uses a research book (like the Engineering Station) to determine what researched recipes may be crafted. Like other Worksites, the Auto Crafting Station requires energy input to function.

The Auto Crafting Station has three inventory areas – raw goods storage, recipe layout, and crafting output storage. Block-side access for each of these inventories can be configured from the ‘Inventory’ button.

To use the Auto Crafting Station begin by laying out your intended recipe in the recipe grid. Items in the layout grid will not be used up while crafting, only items from the raw-goods inventory area will be used when crafting. After your recipe is laid out as intended, the item to be produced should show in the output area. Place any resources needed for crafting the recipe into the raw goods area.

To process the crafting of items in the Auto Crafting Station either supply a power input from workers (work type: Crafting), torque conduits, ‘using’ a Hammer on it, or by pressing the ‘Craft’ button in the inventory GUI. For every press of the button or 50 energy units input a single output item will be crafted. Any items crafted will be placed into the crafting output inventory area. If the output area cannot hold the item to be crafted, no action will be taken (and no energy used).

## Worksite Upgrades

Worksite Upgrades are items that may be applied to a worksite to upgrade or alter its functionality. To apply an upgrade to a worksite right click the item while targeting the block – if it can be used it will be and the item will be removed from your inventory. If the applied upgrade overwrites an existing one, the existing upgrade item will be ejected into the world. Not all upgrades are applicable to all worksites.

The following upgrade types are available:



### Bounds Upgrade Medium

Increases the work bounds maximum width/length to 9 blocks for most bounded worksites. Is overridden by the Large Bounds Upgrade. Cannot be applied to Quarries (use the Quarry Bounds Upgrade items).

### Bounds Upgrade Large

Increases the work bounds maximum width/length to 16 blocks for most bounded worksites. Overwrites the Medium Bounds Upgrade. Cannot be applied to Quarries (use the Quarry Bounds Upgrade items).

### Bounds Upgrade Quarry Medium

Increases the work bounds maximum for a Quarry to 32 blocks. Is overwritten by the Large Quarry Bounds Upgrade. Can only be applied to Quarries (use the normal bounds upgrades for other Worksites).

### Bounds Upgrade Quarry Large

Increases the work bounds maximum for a Quarry to 32 blocks. Is overwrites the Medium Quarry Bounds Upgrade. Can only be applied to Quarries. Can only be applied to Quarries (use the normal bounds upgrades for other Worksites).

### Enchanted Tools 1

Adds the equivalent of 'Looting I' or 'Fortune I' for generating extra loot when harvesting block or animal drops. Also increases Worksite efficiency by a small amount. Is overwritten by Enchanted Tools 2.

### Enchanted Tools 2

Adds the equivalent of 'Looting II' or 'Fortune II' for generating extra loot when harvesting block or animal drops. Also increases Worksite efficiency by a moderate amount. Overwrites Enchanted Tools 1.

### Quality Tools 1

Increases Worksite efficiency by a small amount. Also provides the Quarry with the ability to harvest at an Iron pickaxe level. Is overwritten by Quality Tools 2 and Quality Tools 3 Upgrades.

### Quality Tools 2

Increases Worksite efficiency by a moderate amount. Also provides the Quarry with the ability to harvest at a Diamond pickaxe level. Overwrites Quality Tools 1 Upgrade. Is overwritten by Quality Tools 3 Upgrade.

### Quality Tools 3

Increases Worksite efficiency by a large amount. Also provides the Quarry with the ability to harvest any breakable block. Overwrites Quality Tools 1 and Quality Tools 2 Upgrades.

### Worksite Chunkloader

Force-loads all chunks that are intersected by either the worksite block or any block in its bounds. Ensures that the worksite will continue to process even with no players nearby.



## Quarry Chunkloader

Force-loads all chunks that are intersected by either the quarry block or any block within its work bounds. Ensures that the quarry will continue to process even with no players nearby. Highly recommended for medium and large quarries.

## Warehouse



Warehouse Setup

The Warehouse is meant as an upgradeable self-sorting and auto-organizing mass item storage area. It has work bounds like other Worksites, but is also a Multi-block structure in that it requires other blocks for its functionality. Like other Worksites, the Warehouse requires energy input to function. This energy may come from torque conduits, workers, or a player using a Hammer. The Warehouse comprises four different types of blocks – the Warehouse Control Block, Warehouse Storage blocks, Warehouse Interface blocks, and Warehouse Stock Viewer blocks. Additionally there is a Warehouse Crafting Station to allow a Player to use items directly from the warehouse for crafting.

## Warehouse Control Block



Warehouse Control GUI

The main block of the Warehouse is the Warehouse Control Block. This block manages the bounds of the warehouse and is the target for workers and energy input. It does not by itself offer any storage capability – storage capacity is added by placing Warehouse Storage blocks within the managed bounds. What it does offer is a way to view all stored items, filter view by text input, and manually input and withdraw from the managed storage blocks.

By default the Warehouse manages an area of 5x5x4 (width, length, height) blocks. This area can be expanded through the use of the Medium and Large Bounds Upgrades, which will increase the bounds to 9 and 16 blocks respectively. The bounds to be managed by be adjusted by pressing the ‘Bounds’ button in the main GUI.

The Warehouse Control Block does not offer any externally accessible inventories, all automated item input/extraction must be done through Warehouse Interface blocks. The Warehouse Control Block provides coordination amongst the many types of warehouse block and facilitates moving items from one block to another within the warehouse (from storage to interface, and from interface to storage).

The Warehouse Control Block requires constant energy input in order to move items between blocks. This energy may come from torque conduits, workers, or a player using a Hammer. Energy is only consumed when items are moved, and is the same regardless of stack size or contents.

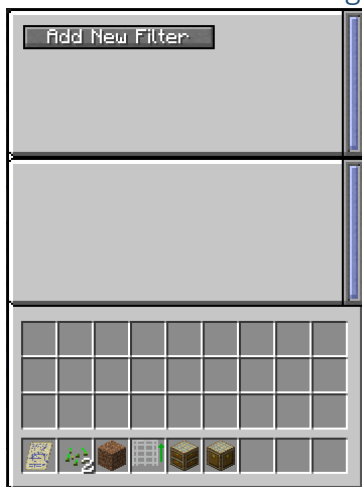
The warehouse controller block may itself be interacted with to display the warehouse GUI. This GUI displays all items stored in the storage blocks controlled by this warehouse controller. Multiple sort modes are available – by name, by quantity, or by user-input text string.

A search input box is available. To use it, set the sort type to ‘Input’. Items with names matching the input will be filtered to the top. Partial matches below full matches. There is a toggle to sort by ascending or descending – default is descending (a-z, 0-9).

Click on any item in the warehouse GUI to withdraw that item. Shift-click to move it directly into your inventory. Items requested in this manner are withdrawn directly from storage blocks and workers are not needed to process the request.

Drop any item into the warehouse item-view panel to manually deposit that item into the warehouse. Items deposited in this manner are moved directly into storage blocks and do not need workers to process the request.

### Warehouse Storage Blocks

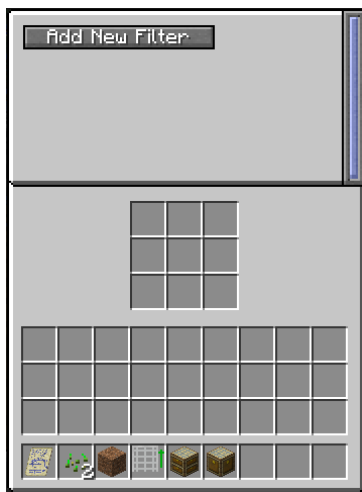


Warehouse Storage GUI

Warehouse Storage blocks provide a means to expand the capacity of the Warehouse. Every additional Storage block placed within the managed bounds will increase the total capacity of the warehouse by a set amount; higher grades of Storage block add more capacity than lower grades. Items are actually stored in these Storage blocks. If you break a Storage block all items it contains will be dropped in the world and no longer available in the Warehouse. Players may manually add and remove items from the Storage blocks, but their main use is for the Warehouse Control Block to manage their stored items

Warehouse Storage blocks may be interacted with to view or manipulate contents or to set content filters. Once a filter is set, only items matching that filter will be placed into that storage block; any existing items will be left in place. An empty filter list denotes that any item may be stored there. An empty filter matches no items (and thus a storage block with a single empty filter will not store any items). These blocks do not provide any externally accessible inventories, all automated interaction must go through the Warehouse Interface blocks.

### Warehouse Interface Blocks

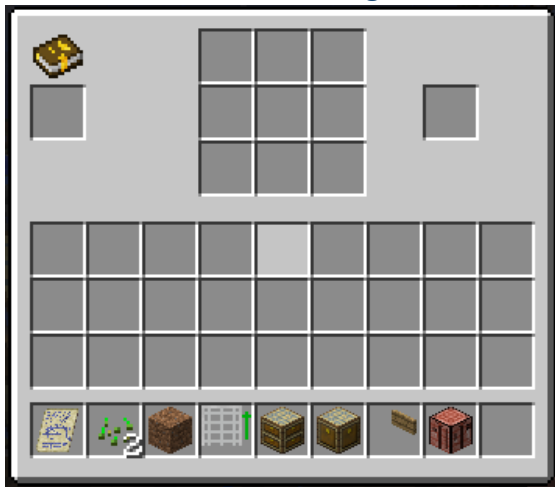


Warehouse Interface GUI

Warehouse Interface blocks provide the means to access the items in the Warehouse through external and automated processes. These blocks provide an externally accessible inventory that may be directly manipulated by NPC Couriers, vanilla hoppers, BuildCraft pipes, or any other mod-added means of manipulating block-based inventories. The Interface block serves as both the input and output for the Warehouse, and every Interface block may be used as input, output, or both.

The Warehouse Interface block can be interacted with to open up a GUI to control its filter list. The filter list determines what items will be automatically placed into the interface, and which items should be automatically removed. Any items in the filter list will be automatically placed into the Interface block at regular intervals by the Control Block, up to the quantity specified in the filter (as long as the Control block is powered). Any items not in the filter list will be automatically removed from the Interface block at regular intervals by the Control Block (again, as long as it is powered).

### Warehouse Crafting Station



Warehouse Crafting GUI

The Warehouse Crafting Station must be placed on top of a Warehouse Control block (not within the bounds), and works with materials from a single Warehouse. Like other crafting stations, the recipe may be laid out in the recipe grid. Results may be pulled from the result slot. Any resources needed from crafting will be pulled from the Warehouse's Storage blocks.

### Warehouse Stock Viewer



Warehouse Stock Viewer GUI

Allows for creating a 'sign' that will display the quantity of specific items stored in a warehouse. It must be placed inside or on the side of block inside the warehouse bounds. It has a configurable list of what items it will display; these are setup similar to the warehouse interface filters.

## Mailbox



Mailbox GUI

Mailboxes serve as a late to end-game item-teleportation system. Item travel time is determined by distance (20 ticks/block, configurable/60 seconds cross dimension, configurable). Mailboxes can be flagged for either the 'public' network, or a 'private' network for each player. Mailboxes do not require any power input or interaction other than adding items to send and remove items received.

Public mailboxes can be interacted with by anyone regardless of team or the owner of the mailbox. Public mailboxes can send to, and receive from, any other public mailbox. Everyone can see the full list of public mailbox names, and add/remove names from this list.

Private mailboxes can only be interacted with by their owner. Private mailboxes can only send to and receive from other private mailboxes of the same owner. Only the owner of a private network can add or remove mailbox names or assign a box a particular private name.

## Chunk Loaders



Deluxe Chunkloader GUI

Chunkloaders force Minecraft to keep specific world chunks loaded and ticking. This removes the requirement of a player being nearby for the world to be active, but only for those chunks under the effect of the Chunkloader. This allows for worksites to continue functioning, NPCs to continue doing their thing, and overall for the game to run like normal for effected chunks.

The 'Simple Chunk Loader' loads a 3x3 chunk area centered on the chunk it is placed in. The 'Deluxe Chunk Loader' has a GUI where you may configure which chunks around it are to be loaded (the top of the GUI denotes north, the chunkloader is in the center). It may load any chunks in a 5x5 area, up to 25 total chunks.

## Torque Power System

The Torque power system is meant as a stand-alone power system compatible with the RF and BC power systems, to allow Ancient Warfare to have full automation capabilities without relying on other mods, and to give players a higher-tier of automation related tech to work towards in the Ancient Warfare research system. The Torque power system also provides a means for power-generation for the Automation Module in cases where the NPC module is not present or otherwise disabled.

The Torque power system includes blocks for power generation, transport, and storage. All blocks are directional – they will only accept or emit power from specific block sides.

All non-machine blocks also have power loss; 1% of stored energy per second base, adjusted by block tier.

Most block types have multiple tiers available; higher tiers have larger transfer and storage amounts, and lower power loss, but also require more research to enable crafting and higher resource use for each block crafted.

Power is converted at a 1:1 ratio between BuildCraft-MJ. MJ may be output directly from any torque tile into BC consuming machines. MJ may be input into torque tiles directly from BC kinesis pipes. There is no way to transfer Torque into BC pipes (only into BC consuming machines).

Power is converted at a 10:1 ratio between Redstone Flux and Torque ( $10 \text{ RF} = 1 \text{ Torque}$ ). RF may be input or output from any torque tile just as if it were a native connection with another torque tile. Full input/output and connectivity are supported. Tested to work with Thermal Expansion storage tiles and MFR energy conduits. Should work with any other RF enabled mod.

The Torque power system comprises many different blocks, these are:

### Torque Shafts

Torque Shafts serve as the main method of transport for Torque energy. A torque shaft has a single input and a single output. A torque shaft may only transfer energy in a straight line (will not bend around corners). For corners use either a junction or a distributor. A torque shaft has lower energy loss than either a distributor or a junction, is cheaper to craft, and should generally be used whenever possible when designing a torque distribution system.

### Torque Junctions

Torque Junctions serve as the main method of transferring Torque power from the point of generation to the point of use (or storage). Each Junction block comprises a set of torque-shafts, multiplexing gear set, and high-efficiency clutch mechanisms to facilitate the transfer of energy between blocks.

Torque Junctions come in three progressively better tiers, Light, Medium, and Heavy. Lower tiers have lower energy transfer and storage ability and higher energy loss. Higher tiers have higher transfer and storage ability and lower power loss.

Torque Junctions have up to five inputs, and a single output (denoted by the pink/flanged side of the block). By default the output is oriented to face the side of the block clicked on when

placing the Junctions. You may use a Hammer or other block-rotation tool to re-orient an already placed Junctions. Junctions will automatically connect to adjacent junctions or other connectable Torque tiles.

### Torque Distributors

Torque Distributors are the multi-output version of conduits. They have a single input and up to five outputs. Output sides are colored pink. All other statistics and uses are the same as Conduits. You may use a Hammer or other block-rotation tool to re-orient an already placed Distributor.

### Torque Flywheels

Flywheels serve as a temporary energy-storage mechanism. As they do incur internal power-loss, long term storage is not recommended, but they do work extremely well as a power-buffer in between your generation and consumer areas. Lower tiers have lower energy transfer and storage ability and higher loss. Higher tiers have higher transfer and storage ability and lower power loss. Flywheels have a single input, and a single (determined by placing direction, the direction faced towards is the output).

Each flywheel is a multiblock structure that must consist of one flywheel control block and one or more flywheel storage blocks. Storage blocks may be placed singly (1x1), or formed into a 3x3 multiblock structure. Once the storage block area is complete, you may place a single flywheel control on the top-center block of the storage area.

Energy is stored in the center/middle storage block on the lowest row – breaking or destroying this block will result in the loss of any energy stored. Any other block in the structure may be temporarily removed without loss of energy.

### Hand Cranked Generator

The Hand Cranked Generator serves as a means to generate Torque power from player and NPC input. NPCs will see it as a Worksite (Type: Crafting), a valid target for work by any worker NPC. Players may use it to generate energy by right-click interacting with a Hammer equipped.

### Sterling Generator

The Sterling Generator is the equivalent of the BuildCraft Sterling / Stone engine. It burns flammable fuels to generate one energy unit per tick as long as it is fed fuel. Anything that may be burned in a furnace should be valid fuel – energy output is a constant 1/tick regardless of fuel type. Burn duration is determined by vanilla burn time / the mod registering the fuel source.

### Waterwheel Generator

The Waterwheel Generator uses the flow of water to output a constant, but small, amount of Torque energy. Waterwheels must be placed facing towards a moving water source. Some example valid setups for a waterwheel generator are below:





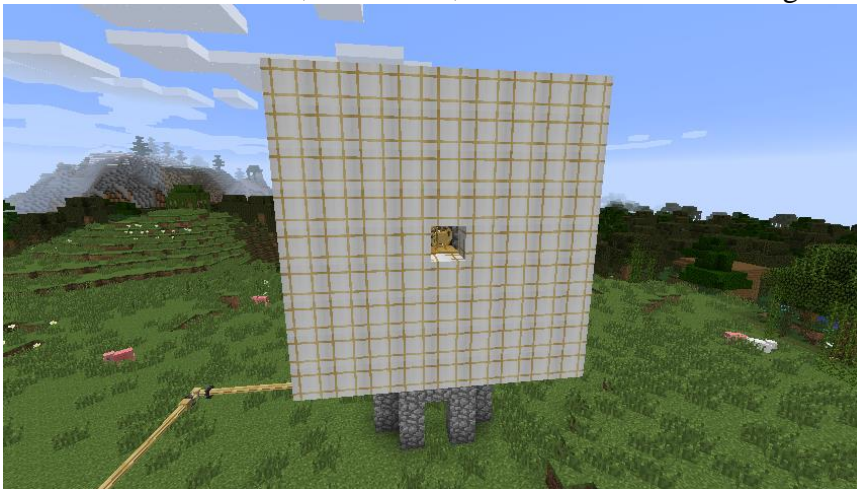
### Windmill Generator

The Windmill Generator is a multi-block structure that consists of a blocks defining a 'blade', and a single 'controller' to manage the energy output.

To create a windmill construct a vertical windmill blade in any odd size from 5x5 to 17x17, all sizes being one block thick (can be built in either the X or Z dimensions, but not both). Once the entire blade structure has been built the individual blocks will join together and the 'blade' will be rendered and begin spinning.

Once the blade has been constructed, place a Windmill Control block on either side of the center shaft in blade. The Windmill Control will output an amount of torque depending upon the size of the blade it is hooked up to. Connecting multiple controls to a single blade will have no benefit – only a single controller will output power.

A 9x9 windmill, unfinished, with center block missing.



The same 9x9 windmill, after the center block is put in place.



## Automation Module Items

### Worksite Upgrades

Please see the section on Worksite Upgrades for more information on these items.

## Automation Module Blocks

### Worksites (Farms, Quarry)

See the Worksites section for a description of each of these blocks functions.

### Auto Crafting Station

See the Auto Crafting section for a description of this blocks functions.

### Mailbox

See the Mailbox section for a description of this blocks functions.

### Basic Chunkloader

See the Chunk Loaders section for a description of this blocks functions.

### Deluxe Chunkloader

See the Chunk Loaders section for a description of this blocks functions.

### Warehouse (Control, Storage, Interface, Stock Viewer, Crafting)

See the Warehouse section for a description of each of these blocks functions.

### Torque Tiles (Conduits, Flywheel, Distributor, Generators)

See the Torque Power System section for a description of each of these blocks functions.



### Section III – NPC Module



Player Owned Workers



Player Owned Utility



Player Owned Combat

## General

The NPC Module adds several types of player-owned and commanded NPCs. It also includes a computer-controlled faction oriented NPC system for use in map creation and world generation through the Structure Module.

Player-Owned NPCs can be recruited for combat, worker, and utility roles. Worker NPCs are intended to be used with the Automation module, and serve little function on their own. Worker NPCs are available in several configurable roles – farmer, lumberjack, miner, researcher, and craftsman. Combat NPCs are available in several configurable roles – swordsman, archer, medic, and engineer. Utility NPCs are also available in several varieties – Priest, Courier, Bard, and Trader (Bard and Trader might not be available on initial release).

Faction controlled NPCs are available in several varieties (both combat and non-combat) for up to eight different factions (five pre-configured, three user-configurable). Many aspects of the Faction system are configurable, including starting faction standing, standing gain/loss for specific inter/actions, and even how the Factions view one another (computer Faction vs. computer Faction warfare).

All NPC types maintain experience and level stats. Player-Owned NPC types track level and experience separately per available role. Higher levels result in higher damage and work output and higher maximum health. Experience is gained by doing whatever action is appropriate for the NPC; workers gain experience by working, soldiers gain experience by attacking and killing enemies, and utility NPCs gain experience by performing their specific roles.

## Player-Owned NPC Crafting (recruiting)

NPC spawning items for the player-owned NPC types may be crafted at the Engineering Station. Each spawning item requires a food bundle (also crafted at the Engineering Station) in addition to some gold for their initial pay, and some equipment to be used in their training. Each

spawning item will create one NPC of the specified type. For exact recipes, please use NEI or similar.

## Player-Owned NPC Interaction



NPC Inventory GUI

To access a Player-Owned NPC's inventory, right click/interact with them – this should open up the NPC Inventory GUI. From here you can adjust the NPC's equipment (which may change their class), change the name displayed on their nameplate, and give the NPC upkeep orders and/or other orders items. Players in creative mode have additional controls available to them – see the Creative Mode Interaction section for more information.

Workers and Combat NPCs use the item equipped in their main hand to determine their current class or role. Changing the item equipped will immediately change their role, including any necessary changes in AI. Utility NPCs only have a single role, but must still be equipped properly in order to function. All Player-Owned NPCs may be given armor, which contributes to damage reduction through vanilla MC methods.

### NPC Equipment

An NPC's equipment may be changed by adding or removing equipment from the various equipment slots. Equipment changes happen immediately, and for some NPC types may change their current role or AI depending upon the slot changed.

The equipment slots available, and their uses, are:

#### *Main Hand*

The main-hand equipment for an NPC determines his attack damage or work-effectiveness, and for many NPC types will determine their role and abilities.

#### *Offhand*

The offhand equipment slot for NPCs may be used to equip them with a shield or other decorative item. Shields will grant an armor bonus / damage reduction. Other items are purely cosmetic.

For combat NPCs a second weapon may be equipped to enable dual wielding. Dual-wielding increases damage output by 50% of main-hand weapon damage, regardless of offhand weapon equipped.



#### Helmet

Equip a helmet here to grant armor bonuses.

#### Chest

Equip chest armor here to grant armor bonuses.

#### Legs

Equip leg armor here to grant armor bonuses.

#### Boots

Equip boots here to grant armor bonuses.

#### Upkeep Orders

Equip an Upkeep Orders item here to manually specify upkeep location for an NPC. If no orders are present, the NPC will use the nearest available same-team Town Hall.

#### Orders

Some Player-Owned NPC types have a specific Orders item that may be equipped in this slot to issue persistent commands. Please see the specific NPC descriptions or the Orders and Routes Items sections for more details.

#### NPC Name Change

An NPC's name may be changed by editing the name displayed in the text box in the Inventory/interaction GUI. Changes will immediately update the displayed name.

#### NPC Repacking

NPCs may be repacked into a spawning item by pressing the 'Repack' button in the Inventory/interaction GUI. When an NPC is repacked the NPC will retain all inventory, experience, and levels.

#### Custom Texture Override

Here you may specify a custom lookup type for the NPC's skin. This custom type should exactly match the custom type specified in the skin-pack.meta file. E.g. if the skin-pack has 'custom.boss.1=boss1.png', then this field should say 'custom.boss.1'.

#### Set / Clear Home Point

Pressing the 'Set Home' button will set the NPC's home point to the block he is standing on. Pressing the 'Clear Home' button will clear the NPC's current home point.

#### Advanced Controls

Some NPCs offer additional advanced controls. If applicable, an 'Advanced Controls' button will be available. Pressing this button will open the NPC's advanced control GUI. The exact types of controls will vary by NPC.

### Player-Owned NPC Horse Interaction

All player-owned NPCs may be ordered to mount and ride a horse while performing their normal duties. This increases the speed at which they move and can give them an advantage in combat against non-mounted enemies. An NPC riding a horse will need special accommodation for his increased entity-size – larger doorways and walkways will be necessary for optimal use, and paths will need their doors removed as mounted NPCs cannot activate doors.

An NPC may be commanded to ride a horse via the command baton by selecting the NPC and issuing a move/guard/mount command (default C key) while targeting the horse to be mounted. The horse must be tamed and saddled (the saddle will disappear while the NPC is riding, but is still present).

To dismount an NPC that is riding a horse, select the NPC with the command baton and issue another move/guard/mount command while targeting the horse he is riding. The NPC should dismount and return to his normal duties.

## Player-Owned NPC Upkeep

All Player-Owned NPCs require upkeep in the form of food. If they are not fed, they will cease doing their duties; workers and utility NPCs will wander about, and Combat NPCs will only attack in self-defense. They will also cease to automatically heal as long as they are hungry. If they have an assigned upkeep point, they will wait near that point until fed.

The specific food values withdrawn at every upkeep period is configurable through the Upkeep Orders item. If no Upkeep Orders are equipped, the NPC will attempt to withdraw enough food for five minutes (real-time).

The specific food value granted by an item and which items are considered food are completely configurable through the “AncientWarfareNpcFood.cfg” configuration file. The values specified for food are the amount of ticks each item will feed the NPC for.

NPCs without an Upkeep Orders item equipped will attempt to withdraw their upkeep from the nearest available Town Hall block. A player may also manually specify the upkeep location for an NPC by using the Command Baton, but this does not allow for setting how much food will be withdrawn at every upkeep cycle.

## Player-Owned NPC Teams

All Player-Owned NPCs take team status directly from their owner. If their owner changes team, they will change team at the same time. NPCs may only be commanded by their owner or another player on the same team. Neutral NPCs may only be commanded by their owner.

Players not on a team are considered ‘Neutral’. NPCs from players on teams will not attack neutral NPCs, nor will neutral NPCs attack any other Player-Owned NPC. Neutral NPCs will still respond to Faction NPCs as appropriate for their faction standing (see the Player-Owned NPC Faction Standing section for more details).

## Player-Owned NPC Faction Standing

All Player-Owned NPCs derive their faction standing (and view of Faction owned NPCs) from their owners faction standing. If a faction is hostile towards an NPC’s owner, that NPC will also be hostile to that faction. A player faction standing can be directly influenced by his NPCs; if they kill a faction owned NPC, their owner’s faction standing will be negatively affected.

## Player-Owned Worker NPC Types

When first recruited, Worker NPCs are of a generic ‘Worker’ type and must be properly equipped before they can begin their career. The role the worker takes on is dependent upon the type of equipment given to them – you must equip the worker appropriately for his new job.

The material quality of the equipment will contribute to the workers effectiveness and work output. A workers level will also contribute to his effectiveness at his current task.



Worker levels are tracked independently per role; changing role will result in the worker using his level for the new role for efficiency calculations, and any experience gained will contribute to his level for that role only.

NPC Workers will automatically seek the nearest active worksite that has work to begin working at. If no worksites have work to be done, they will begin to wander around. Automatic selection of worksite can be overridden by placing a properly configured 'Work Order' item in the workers 'Orders' slot.

Workers will automatically return to their home point at night or whenever it is raining. The home point is set when the NPC is initially spawned, but may also be changed through the interaction GUI or with the Command Baton. To disable the 'go home at night' AI, remove/clear/un-assign the NPCs home point.

If a Worker NPC spots a hostile entity they will broadcast an alert and begin moving towards the nearest Town Hall while attempting to evade the enemy. If no Town Hall is near, they will move towards their home point while attempting evasion. If no Town Hall is present, or home point set, they will continue working like normal. Other nearby NPCs may respond to this broadcast in different manners – other nearby Workers and Utility NPCs will begin to flee as well, while nearby Combat NPCs may move to attack the enemy.

The types of Worker NPC, and necessary equipment, are:

#### Worker

The generic Worker NPC. When equipped with no item or any item not activating any other role, the Worker will revert to this role. The only thing he is good for is eating all of your food. Don't let these guys mill about, give them some proper equipment and send them to work!

#### Farmer

The Farmer works at the Crop Farm, Reed Farm, Mushroom Farm, Animal Farm, and Fish Farm worksites. Required equipment is a Hoe, of any material level, in the main-hand slot. Offhand and armor may be configured as desired.

#### Miner

The Miner works at the Quarry and Construction Site worksites. Required equipment is a pickaxe, of any material level, in the main-hand slot. Offhand and armor may be configured as desired. Pickaxe material level does not determine harvestable blocks at a Quarry – that is determined by the Quarry's upgrade level.

#### Lumberjack

The Lumberjack works at the Tree Farm worksite. Required equipment is an axe, of any material level, in the main-hand slot. Offhand and armor may be configured as desired.

#### Researcher

The Researcher works at Research Stations, to enable continual research progress without player interaction. Required equipment is a Quill, of any material level, in the main-hand slot. Offhand and armor may be configured as desired.

#### Craftsman

The Craftsman works at the Auto Crafting Station and Warehouse worksites. Required equipment is a Hammer, of any material level, in the main-hand slot. Offhand and armor may be configured as desired.

## Player-Owned Combat NPC Types

Player-Owned Combat NPCs, like Workers, are unskilled when first recruited. You must equip them properly for whatever their intended job will be. Damage output is determined by NPC level and the material level of the items equipped. Like workers, Combat NPCs track experience and level per-role.

If a Combat NPC spots a hostile entity they will broadcast an alert and begin moving towards attack distance from that enemy. Other nearby NPCs may respond to this broadcast in different manners –nearby Workers and Utility NPCs will begin to flee, while other nearby Combat NPCs may move to attack the enemy.

If a Combat NPC has a home point set he will attempt to stay within a set distance of this home point even when pursuing enemies. If he strays too far, he will abandon his current task/target and begin moving back within a set distance from his home point.

Combat NPCs may be assigned patrol routes by equipping them with a properly setup Combat Orders item in their Orders slot.

The types of Combat NPC, and necessary equipment, are:

### Combat

The basic Combat NPC is the unskilled recruit. He has a very basic combat AI and only does a single damage per hit. Any time a combat NPC has no main-hand item, or other unrecognized item, he will revert to this role.

### Soldier

Equipping a Combat NPC with a Sword (any material level) will turn him into a Soldier. The soldier has no specific special abilities. His damage output is determined by the material level of the sword equipped – diamond weapons do more damage than wood. Offhand and armor may be configured as desired.

### Archer

Equipping a Combat NPC with a vanilla bow will assign him the role of Archer. An Archer's sole special ability is ranged attacks. They have the same range for attacks as a vanilla Skeleton. Offhand and armor may be configured as desired.

### Medic

Medics will attempt to heal nearby same-team injured units, including Players. A Combat NPC may be assigned the Medic role by equipping an Axe in his main-hand. Damage output is determined by material level of the axe equipped. Offhand and armor may be configured as desired.

### Engineer

Engineers will pilot an assigned Siege Engine. A Combat NPC can be assigned the role of engineer by equipping him with a Hammer in his main-hand. Offhand and armor may be configured as desired. As the Vehicles Module is not-yet-implemented, the Engineer currently serves no purpose.

### Commander

Commanders boost the damage and armor of nearby same-teamed NPCs. The Commander role may be assigned to a Combat NPC by equipping him with a Command Baton

in his main-hand. Damage output is determined by material level of equipped Baton. Offhand and armor may be configured as desired.

## Player-Owned Utility NPC Types

Several types of utility NPC are available to aid the Player in a several types of tasks. Each utility NPC is of a set type, once recruited/spawned their role may not be changed. Many still require specific equipment to function properly, see the details below for more info.

Like Worker NPCs, Utility NPCs will return to their home point whenever it is night time or raining. This function may be disabled by clearing/removing/un-assigning the NPCs home point.

The types of utility NPCs available are:

### Courier

The Courier is the master of logistics. He is skilled at moving items from one place to another and keeping things stocked up and flowing. The Courier must be equipped with a Backpack in his main-hand for inventory space. The type of Backpack determines how much room the Courier has for storing items. Items that the Courier is moving are stored in the backpack; you may remove it at any time and view or manipulate the contents.

A Courier must also be equipped with a Routing Order in his orders slot. The Routing Order specifies what blocks the courier will be transferring items to and from, and what items he will be moving at each stop. Please see the NPC Orders section for more details regarding the Routing Order setup and functioning.

### Priest

The Priest NPC can make combat operations less risky, or at least less costly. His only function is to resurrect slain NPCs. In order to be eligible for resurrection the slain NPC must have had a Town Hall assigned, died within range of that Town Hall, and meet other (still in development) criteria.

Currently there is no cost associated with the resurrection process, but this will be changed in the near future – each resurrection will require a material and time cost. It should be costly, but still less expensive than recruiting a new NPC and training him up.

Resurrected NPCs return unequipped – their equipment was dropped when they were slain, and must be manually retrieved and/or replaced.

### Bard

The Bard is a ‘flavor’ NPC – he has no real use aside from playing music. The music to be played, play frequency, and volume may all be configured from his advanced controls GUI. These controls may be accessed by pressing the ‘Advanced Controls’ button in his inventory/interaction GUI.

To add a new song to the bards’ play-list, click on the ‘Add Tune’ button. A blank entry will be created at the bottom of the list for you to fill in. Input the registered sound name (see Appendix IV – Bard Tunes for a list of names) in the name box. Input the length of the song (in minutes, decimal format) into the length box. Any sound loaded and registered in Minecraft may be played through the bard; you may need to consult external documentation to locate the registered name and play-length of the sound.

A bard may be configured to only begin playing when a player first enters his range. Check the ‘Play on Player Entry’ checkbox at the top of the GUI to enable this feature. Once

enabled he will only play a single song when the first player comes within his range; he will not play again when a second player enters range unless the first had already left –and- the song was over. The bard will continue to play songs until there are no more players within range.

Song play-order may be randomized by checking the 'Random' checkbox at the top of the GUI. The min and max delay settings at the top of the GUI determine the random pause that will occur between playing of each song. This setting is given in game-ticks (20 ticks per second).

### Trader

The trader is a player-configurable mobile bartering station. The trader must be equipped with a Backpack in his main-hand for item storage. Any items to be traded, and proceeds from trades, are stored in this backpack. A Trader must also be equipped with a Trade Orders item in his Orders slot. Through this you may assign him a withdraw and deposit point for his traded goods, assign the specific trades he will have available, the route he should take for trading, and how long he should linger at each stop on his route.

The Trader will seek shelter at night or in the rain, he will seek shelter at the nearest waypoint specified in his trade orders item. If a Trader is not equipped with a trade orders item he will fall back to default AI regarding upkeep and night/rain.

Please see the Trade Orders item section for more details on setting up the Trade Orders for Player-Owned Traders.

### Town Hall

The Town Hall block functions as an auto-assignable upkeep point for its owners NPCs. NPCs in range will automatically select the closest Town Hall as their upkeep point if no Upkeep Orders are present. The Town Hall has a modest sized inventory for storing food for NPC upkeep.

The Town Hall additionally serves as a death tracker for any NPCs that died nearby. Players may view this list by pressing the 'Death List' button in the Town Hall GUI. The list may be cleared out manually by pressing the 'Clear List' button in the death list GUI. Priests use this death list to track potential resurrection targets, so do not clear the list if you want to resurrect any of the NPCs on it.

### Orders and Routes Items

Orders items offer a means to issue advanced persistent orders to an NPC. All NPCs have two orders slots; the top slot is the Upkeep Orders slot, and the bottom slot is the role-specific orders slot.

Each Orders item functions slightly differently, but all share a common interaction scheme: Right click to open the GUI and view/edit the orders. Press the item-interaction key 1 (default 'Z') to add a new target to the orders.

While holding an Orders item the target blocks denoted by the orders will be highlighted, and lines rendered between them to denote the route/ordering.

The types of Orders items, and their functions, are:

#### Upkeep Orders

The Upkeep Order item denotes the site from which a Player-Owned NPC should withdraw his upkeep, and offers the ability to set the upkeep period (how often the NPC should

go back to eat). Press 'Z' while holding the Order to assign the target block. Right click to open the GUI to alter the upkeep period. Altering the upkeep period directly effects how much food is withdrawn on each visit. If not enough food is available nothing will be withdrawn, and the NPC will wait at the upkeep point for it to be restocked.

### Combat Orders

Combat Orders offer a means to assign a patrol route to Combat NPCs. The patrol can comprise any number of stops, but they must each be within path-finding distance from the previous stop (if the route is too far, the path-finding will fail and the NPC will not move). New patrol stops can be added by pressing the 'Z' key while targeting the intended position. Patrol points can be removed or re-ordered by right-clicking to open the GUI.

### Routing Orders

Routing Orders are used by Courier NPCs to denote the route they should travel and which items should be transferred at each stop. To add a new route-point, press the 'Z' key with the intended block (and side!) targeted. Right-click the Routing Order to adjust the route type, inventory side, items transferred, or to change the route-point ordering.

Each routing point operates on a specific side of the inventory, enabling sided-inventory interaction for furnaces and mod-added inventories. To change the target side, open the Routing Orders GUI and click the button labeled with the current side (it will say on of: North, South, East, West, Up, or Down). The selected side will scroll through the side-list with each press of the button.

Each point on the route is allowed a single type of item-transfer. The same block/inventory may be specified multiple times for more complex interactions. To change the route-type, press the button labeled with the current route type. The types of item transfers available are:

#### *Take Matching*

The Take Matching route type instructs the courier to remove any items matching those specified in the points filter list. The courier will continue to take items matching the filters until either there are no remaining items or his inventory is full.

#### *Deposit Matching*

The Deposit Matching route type instructs the courier to deposit any items in his backpack that match the filters specified for that point. The courier will continue to deposit matching items until either his inventory is empty or the target inventory is full.

#### *Take Non-Matching*

The Take Non-Matching route type instructs the courier to remove from the target inventory any item that does not match the filter list specified for that route point. An empty filter list will see the courier remove all items from the target inventory. The courier will continue to remove any non-matching items as long as he has room in his backpack.

#### *Deposit Non-Matching*

The Deposit Non-Matching route type instructs the courier to deposit into the target any items in his backpack that do not match any of the filters specified for that route point. An empty filter list will see the courier deposit all of his items. The courier will continue to deposit non-matching items as long as there is room in the target inventory.

### *Fill Target*

The Fill Target route type will see the courier attempt to fill the target inventory with the specified quantities of the specified items. If the target inventory already contains the specified amount, no action will be taken. The courier will continue to deposit items into the target inventory until either the specified quantity is met or the courier is out of the specified items.

### *Fill Courier*

The Fill Courier route type will see the courier attempt to fill his inventory from the specified target, of the items specified, up to the specified quantity. The courier will continue to take items as long as he currently is carrying less than specified and the target still has items to take.

### *Trade Orders*

Trade Orders are equipped by Traders in the orders slot, and allow specification of the trade route, what trades will be offered, and where/what the NPC should restock/deposit. The Trade Orders GUI has one tab/section dedicated to each; 'Trades' allows setting what trades the NPC would have, 'Route' allows for manipulating the route the NPC will follow, and 'Restock' allows for specifying the position(s) and item(s) the NPC will deposit/withdraw at the conclusion of every trip. Press the corresponding button to view that section in the setup GUI.

Right-click to open the configuration GUI. Press item activation key 1 (default 'Z') to add a new point to the trade route specified in the Trade Orders. Press item activation key 2 (default 'X') to specify the deposit point, and item activation key 3 (default 'C') to specify the withdraw point. The points specified may be manipulated further from within the setup GUI.

### *Trades*

The Trades section of the GUI allows for adding/changing/removing the trades that could be available for the trader. The left grid is for the input items, the right grid for the output items. Trade recipes are both layout and number specific.

### *Route*

The Route section of the GUI allows for manipulating the Traders intended route. You may remove waypoints, change their order, change the 'wait delay' at each waypoint, or specify a point as an 'Upkeep' point.

The Delay setting is the number of game ticks that the NPC will stand at the waypoint for (waiting for people to trade). After the specified number of ticks the NPC will either proceed with upkeep (if waypoint was marked as upkeep), or proceed to his next waypoint.

At each point specified as 'Upkeep' the NPC will attempt to return to his assigned upkeep point to withdraw food. He will then proceed to his deposit and withdraw points (if specified) to perform restock operations. See the next subsection for details regarding Restocking.

### *Restock*

The Restock section of the GUI allows for specifying what items the Trader should drop off or refill at the configured deposit and withdraw points. Here you may specify what items to be deposited and withdrawn at each point, with various options for item-quantity manipulation for each item.

The deposit/withdraw types are:

- Deposit Excess – deposit any items in excess of the # specified in the GUI.
- Deposit All – deposit all items matching that specified in GUI



- Deposit Quantity – deposit the quantity specified in GUI (or partial if full not available).
- Withdraw All Of – withdraw all of the specified item found, until full.
- Fill Courier To – withdraw enough of the specified item to ensure the courier has the specified quantity in his inventory.

## Command Baton Interaction

The Command Baton gives a player the means to issue orders to owned or same-team NPCs. Each Command Baton stores its own list of NPC targets; multiple Batons may be used for ‘command groups’. The Command Baton also functions as a sword, doing damage equivalent to a same-material sword.

The list of commanded NPCs is displayed in the upper-left of the screen. The current target is displayed in the upper-right of the screen. To add or remove an NPC from the list of commanded NPCs, right click while targeting that NPC.

Orders are issued to a Batons’ commanded NPCs through the item-interaction keybinds. Orders are issued with the currently highlighted block or entity as the target. Orders are issued to the entire list of commanded NPCs for that Baton. To issue an order press the corresponding order key. The currently available keys/commands are:

Z – Clear current command

X – Attack target, or move to target position while attacking any sighted enemies

C – Follow target / Guard target / move to target position / Mount target Horse

V – Set home point, will clear point if sneak is held while issuing order.

B – Set upkeep point, will clear point if sneak is held while issuing order.

## Base NPC Stats and Attributes

### Health

NPC Health is determined by the NPCs level. For NPCs with multiple roles, health is tracked from a base level statistic and does not depend on their level for any specific role (all roles contribute experience to the base level).

### Armor

NPC Armor value is determined by the armor that is equipped on the NPC. Damage reduction calculations are done through vanilla methods – armor on an NPC should have the same effects as armor on a Player.

### Damage

NPC Damage is calculated by a combination of their equipped weapon and their level at their current role. Higher tiers of material generally do more damage, and swords do more damage than axes.

### Work Effectiveness

Work Effectiveness for an NPC is calculated from his equipped tool and his level at his current role. At the highest level and diamond tools an NPCs output is ~2x that of a fresh recruit with wooden tools.



## Level

NPC Level is determined by the amount of experience gained. After gaining a set amount of experience (value increases per level) the NPC will gain one level in his current role. NPCs achieve a maximum of ten levels (configurable in configuration file).

## Experience

Every career related action that an NPC takes will grant him experience. The amount granted depends upon the type of action. Killing an enemy gives more experience than attacking an enemy, and attacking an enemy gives more than working at a worksite or transferring items.

## NPC Leveling and Experience

All NPCs have a base (shared) level and experience value. Base level determines max hit-points. Player-Owned NPCs also have a sub-type level for every role. Sub-type level determines bonus damage or work-effectiveness.

Base and sub-type experience and levels are stored and calculated separately. Experience and level for each sub-type/role is stored separately. Changing an NPC's role (by changing equipment) will change the sub-type that he uses for experience/level lookup and for experience/level gain. NPCs without subtypes track only a single experience and level value.

Every time an NPC gains experience that experience is contributed towards both his base experience/level and his current role experience/level (if applicable).

## NPC Healing

All NPCs will 'heal' over time on their own, slowly. The time delay between heals is customizable via configuration file. Player-owned NPCs require that their upkeep be met in order to heal – they will not auto-heal when hungry.

## Creative Mode Interaction and Advanced Controls

Players in creative mode may interact with Faction owned NPCs as if they were a Player-Owned NPC owned by the interacting Player. A creative mode player may access the inventory GUI and issue commands to Faction owned NPCs. See the Player-Owned NPC Interaction section for more details.

Faction based NPCs do not use their upkeep or work orders slots, so these slots may be filled with items to act as custom-rewards for when the NPC is slain. They also do not use their primary weapon slot to determine combat type, so you may equip any item in their weapon slot for cosmetic effect. However, if using a non-weapon item, it is recommended to set an Attack Damage Override, as the NPCs **do** use the weapon to determine their base attack damage.

Players in creative mode have additional controls available on all NPCs. These controls are accessible via the 'Creative Controls' button in the NPCs main inventory screen. The additional controls are:

### Set Owner Name

This option is used to reassign a player-owned NPC to a different player. Has no effect on faction-owned NPCs.

### Enable/Disable wander

This checkbox will turn on and off the NPCs wander AI. This is a temporary setting, not saved. It should be used to aid in structure or map creation and NPC positioning.

### Attack Damage Override

If this value is set to  $\geq 0$ , this value will be used for attack damage instead of the NPC's equipped weapon. It is most useful for creating custom NPC bosses or subtypes.

### Armor Value Override

If this value is set to  $\geq 0$ , this value overrides the armor value from whatever armor the NPC is wearing. Most useful for creating custom NPC bosses or subtypes that should be armored but still show their texture.

### Max Health Override

When set to  $>0$ , this value will be used to determine the NPC's max health. It can be used in combination with the other overrides to create custom NPC bosses.

## NPC Faction System

Faction owned NPCs are NPCs not under the control of any Player, whose hostility towards any particular Player is determined by a faction standing value (also commonly known as reputation).

If a Player's standing with a faction is too low, combat NPCs belonging to that faction will be hostile towards him. If/when the Player raises his faction standing enough the NPCs will cease to be hostile. Faction standing is tracked per faction per player – every player has their own faction standing value with each faction.

Players may influence their standing with a particular faction by interacting with NPCs belonging to that faction. Attacking or killing an NPC belonging to a faction will result in a loss of faction standing. Trading with a trader belonging to that faction will result in a gain in faction standing.

There are also 'Elite' versions of several of the faction owned combat NPC types. These elite versions have an especially hostile view of players, and will continue to be hostile long after the rest of the faction has turned neutral/friendly. The elite versions are intended to be used in structure generation and map creation as royal guards or treasury guards; placed in areas that the player should not have access to until they are extremely friendly with that faction. The amount of standing offset for elite units is configurable in the configuration files.

The starting faction standing for new players joining a game can be configured through the configuration files. Faction standing / hostility of one faction towards another may also be configured through the configuration files, so you may have any particular faction at war with another.

There are currently five fully defined factions ready for use in structure and map creation, as well as three user-configurable factions for use by advanced map creators. User-configurable factions may be defined through configuration file and skin-packs. See the User-Defined Factions subsection below for more details. Names, health values, and starting equipment for every NPC type for each faction may be individually specified through the configuration files.

### Faction Owned NPC Types

There are currently several combat and non-combat NPC types available to every faction. Each type has a specific role and the accompanying AI to suit that role. Faction owned NPCs spawn with basic equipment, but may be further customized by map makers or structure designers through their Inventory GUI and creative mode controls.

The following types of faction owned NPCs are currently available:

### *Soldier*

The basic foot-soldier, a Soldier is a melee unit with no special abilities. Damage output is determined by the weapons(s) equipped.

### *Elite Soldier*

An Elite Soldier is the more hostile version of a Soldier. All starting equipment and stats are identical to the Soldier.

### *Mounted Soldier*

The Mounted Soldier is a horse-riding version of the Soldier. All starting equipment and stats are identical to the Soldier.

### *Archer*

An archer is a basic ranged attack unit. It has no other special abilities. Attack range is the same as a vanilla Skeleton.

### *Elite Archer*

The Elite Archer is the more hostile version of the Archer. All starting equipment and stats are identical to the Archer.

### *Mounted Archer*

The Mounted Archer is a horse-riding version of the Archer. All starting equipment and stats are identical to the Archer.

### *Leader*

Leaders are the chief or commander of a settlement, but may occasionally be seen in command of raiding parties as well. They have slightly better stats than a regular Soldier, but share a common AI.

### *Elite Leader*

Elite Leaders are a more hostile version of the Leader type. All starting equipment and stats are identical to the Leader.

### *Priest*

Faction owned NPC priests act as spiritual leaders, wise-men, and healers for their settlements and excursions. They have minimal combat damage, but can heal the wounds of nearby friendly units.

### *Civilian Male and Female*

The civilian units are decorative, non-hostile, faction owned NPCs. They will not initiate combat, but if attacked will signal to the nearest combat unit for help.

### *Trader*

The Trader is a faction owned NPC with a configurable trade-list. Players in creative mode may interact with him to set his list of trades. Players in survival mode will see the trade list GUI when they interact with him.

Trades are specified similar to how crafting recipes are laid out. Position –and– quantity matter. If you ask for 2x planks in a slot, the user must input 2x planks in that same slot.

Left side is the ‘input’ side. Right side is the ‘output’ side. Max Available is how many the trader may have on hand at any particular time. Restock frequency specifies how often the trader will restock a single unit of the given trade. A Restock frequency of 0 denotes INSTANT restock (will never run out). A Restock frequency < 0 denotes that

the trade is limited and will NEVER restock (as soon as the trader is out of stock, the trade is no longer displayed).

### Pre-Defined Factions

The pre-defined factions are each themed for a particular biome. This is merely a guideline for their default naming and not enforced anywhere in the code. Many aspects of the pre-defined factions are available for configuration via the configuration files, including default names, health values, starting equipment, and skins.

The pre-defined factions are:

#### *Bandits*

Bandits may be found in plains and forested areas, waiting in ambush for unsuspecting travelers, or occasionally seen hanging around one of the rare bandit settlements.

#### *Pirates*

Pirates can often be found in coastal regions, on secluded islands, or on rare occasions you might find one of their treasure-laden ships becalmed in the middle of the ocean.

#### *Vikings*

The Vikings have no fear of the cold, and have had great success in surviving in frozen plains and tundra. Rumors tell of Viking raids on coastal towns, but no survivors have been found for confirmation.

#### *Jungle Natives*

Jungle Natives make their homes in and under the trees of Jungles and other biomes with dense flora.

#### *Desert Natives*

One of the few living things capable of surviving in such a landscape, the Desert Natives don't just survive but often thrive in the harsh conditions.

### User-Defined Factions

There are three user-definable factions included for map makers to further customize the experience. They do not come with pre-defined names, skins, or equipment – all of these must be set by the map maker. Names may be specified through the NPC names configuration file. Skins may be added through the skin-pack system (see the Skin Packs section for more details). Starting equipment and health for each faction and NPC type may also be specified through the respective configuration files.

### Skin Packs

All NPCs in Ancient Warfare use a configurable skin system to determine the texture used for rendering. Custom skins may be loaded through a Skin Pack. Skin Packs may add additional skins for any existing NPC type. When an NPC type has more than one skin available, a random one will be chosen from all available for that type. Any NPC may be specified to use a custom skin by entering the skin name in the 'Custom Skin' text box in their Inventory GUI. The default skin pack that ship with the mod may be disabled through the configuration files.

Skin Packs consist of any number of Minecraft-skin formatted .png files as well as a single skin-pack.meta file all packaged as a single .zip file. Custom skin-packs must be placed into the skin-pack folder, located at: <Minecraft directory>/config/ancientwarfare/npc/skins/. Custom skin-packs may have any file-name, as long as they have a .zip extension.

The format of the skin-pack.meta file is:

- One entry per line. Each entry denotes a new skin. No spaces, no blank lines, no comments.
- Entries are specified by skin name, equals sign, image-file name. E.g. 'worker=worker1.png' specifies a new skin called 'worker' using the texture 'worker1.png' from the skin-pack.
- Skin names may be specified multiple times with different images. Each additional entry adds to the pool of skins that will be used for random selection for that name/NPC type.
- Many pre-defined names exist for the existing NPC types. Adding additional entries for these names. See Appendix I for a list of the pre-defined NPC names for use in Skin Packs

## NPC Custom Naming

The default name for all NPCs (faction, player owned) may be over-ridden through the use of a Resource Pack containing a language file with the intended names to override. For more information on resource-pack creation, please see:

[http://minecraft.gamepedia.com/Resource\\_pack](http://minecraft.gamepedia.com/Resource_pack) .

An example name-overriding resource pack may be found in the Ancient Warfare source repository, located at:

[https://github.com/shadowmage45/AncientWarfare2/tree/master/project\\_resources/respack](https://github.com/shadowmage45/AncientWarfare2/tree/master/project_resources/respack) .

## NPC Module Items

### NPC Recruiting Slips (NPC Spawners)

The NPC Recruiting Slip spawns a single NPC when used. Right click it on the block where you wish to spawn the NPC. The NPCs 'home' point will be the block targeted. An NPC may be repacked into a Recruiting Slip by pressing the 'Repack' button in his interaction GUI. A repacked NPC retains all equipment, levels, and health statistics when re-spawned.

### NPC Orders (Upkeep, Routing, Combat, Trade, Work)

See the NPC Orders section for details about these items' functions.

### Command Baton

See the Command Baton section for details regarding this items use and functions.

### Ration Pack

Used as a crafting component for NPC Recruiting Slips

### Instruments (lute, harp, drums)

Used by the Bard NPC for cosmetic effect.

### Shields

A Shield may be equipped in an NPC's offhand to grant him an armor bonus. The exact bonus depends upon the material level and type of shield equipped.



## NPC Module Blocks

Town Hall

See the Town Hall section for details about this blocks functions.

## Section IV – Structure Module



### General

The Structure Module in Ancient Warfare encompasses a template based structure generation system, tools to aid in creative mode building and map creation, and tools to enable in-game creation of new structure templates. The Structure Module additionally contains a system to allow automated construction of buildings while in survival mode, with resource use and energy requirements for construction.

The template system handles the scanning and placement of blocks and entities through an extensible plugin system. By default all vanilla and Ancient Warfare blocks and entities are supported through a set of included plugins.

Support for additional mod blocks and entities may be added through the template plugin API by other modders. Templates using additional plugins will require those same (or compatible) plugins be present for the template to be properly constructed or used for world generation. Mod added blocks without a plugin present will be supported through block-id and metadata only. Mod added entities without a plugin present will be ignored.

### Configuring World Generation

By default Ancient Warfare does not ship with world generation enabled. Due to the multitude of settings, mod configurations, and personal preferences, enabling world generation is left as an option to the user or server administrator.

Structure packs will be available for download from the same sources as the mod (with more available over time), or new structures and packs may be created manually for additional customization. See the Template Packs and Creating Templates sections for more details regarding creating new templates and packs.

The first step in configuring world generation is to select the structures and/or structure packs that will be used. Place your desired templates and template packs into the templates configuration folder, located at <Minecraft folder>/config/ancientwarfare/structures/included/. Any templates or template packs found will be loaded at when the game starts up. Any templates flagged for world generation will be included for selection by the world generation system.

Depending upon the structures and packs selected, some templates or configuration settings may need to be modified for them to all play nicely together. Template packs with vastly different recommended configuration settings may need to be modified to be brought in-line with each other. It is generally recommended to only use a single structure pack, or only use packs in combination that were designed to be used together. Any conflicts between recommended settings for multiple packs are left to the user or administrator to resolve – do not contact the mod or pack authors with these conflicts.

Additionally, some template packs were designed to be used with other mods (mostly biome/world gen mods) – please see the description and details on each pack for any mod requirements. If the required mods are not present the templates may not generate properly or at all.

A few configuration options are available in the structures configuration file. Some of these settings may need to be adjusted to conform to a structure pack's recommended settings. These options, and their functions, are:

#### [max\\_cluster\\_value](#)

This setting denotes the maximum total 'Value' worth of structures that may be generated in any given region. The size of the region is determined by the configuration setting 'validation\_cluster\_value\_search\_radius'.

Cluster Values for a structure are specified within that structures template. The value that should be used for the 'max\_cluster\_value' setting is very closely related to the structure pack(s) in use, and an incorrect value may result in no structures generating, or far too many structures generating. Generally the readme or credits file of a structure pack should list the recommended setting for 'max\_cluster\_value'.

A higher value for this setting will result in more structures being allowed in any specific region, and a generally more dense generation of structures. Lower values reduce the number of structures generated in a region, too low of a value and not even a single structure will be allowed to generate.

See Appendices II and III for more details regarding how cluster value and cluster search radius are used during world generation.

#### [validation\\_cluster\\_value\\_search\\_radius](#)

This setting determines the radius in which cluster value will be totaled for determining if any more structures may be generated in any specific chunk. All structures within the range specified by this setting will have their 'Cluster Value' totaled and checked against 'max\_cluster\_value'. Only templates with a cluster value less than the remainder will be eligible for generation.

See Appendices II and III for more details regarding how cluster value and cluster search radius are used during world generation.

### validation\_duplicate\_search\_radius

This setting determines the maximum range in chunks that will be searched for duplicate structures when validating the 'minDuplicateDistance' setting in a template. Even if the template specifies a higher value, this value will be used as a maximum.

## Template Format Overview

The template file format was designed as a computer generated, user-editable, structure data storage file. It stores information regarding the blocks and entities contained in a structure as well as any settings pertaining to world-generation placement validation for that structure.

The default file extension is '.aws', but may be changed through the configuration file on a per-user basis – if changed ALL templates (including those in packs) must have the new extension or they will not be loaded.

Many areas in the template format store information in JSON formatted condensed NBT. These sections are parsed by Minecraft directly into NBT tags, which are then decoded like normal to retrieve the stored information. Editing of these sections is not recommended, as slight variances in the formatting can lead to unpredictable results, and in many cases may crash the game when the data is attempting to be decoded.

The template file is separated into several structured areas:

### Header

Contains information shared by all template types regarding basic settings and configuration.

#### Version

The version of template format used in the template

#### Name

The name of the structure, must match the name of the file (minus extension)

#### Size

The size of the structure, specified in width, height, length

#### Offset

Offset is the offset position from back-left corner that should be considered the ground-level main entrance. This setting is used to determine placement of a building when using the instant construction tools, and for any potential village structure additions.

### Validation

The validation block contains information used for placement validation for world-generation. These settings are responsible for making sure a structure is appropriately placed in the world; that it does not hang over a cliff or cut into hillsides. Multiple validation types exist, and each validation type has different options used to configure validation of structure placement. See Appendix II – Structure Validation Types for more information regarding validation types and their settings.

### Layers

The layers block contains a comma and line separated grid of block-lookup keys for every block position in the structure. A value of zero denotes an empty air block. Any other

value specifies a single block rule from the Blocks section. It is possible to edit the layers manually, but not recommended.

### Blocks

The Blocks section contains a list of block data sets pertaining to the blocks in the template. When constructing a template the block to be placed is looked up by rule number from the layers grids. Each rule in this section should specify a single unique type of block – many duplicate blocks are removed during the automated scanning process. Each rule also specifies the plugin that is responsible for loading the rule from disk – if this plugin is not present, the block will be skipped during template construction.

The rules include any information about metadata, tile entities, and inventories for the referenced block. This information will be used when constructing the template to recreate the block and tile entities. If the block specified by the rule is not present in the game (removed or not-present mod) the block will be skipped.

It is possible to manually edit the block rules for various effects, but is unsupported and generally not recommended.

### Entities

The Entities section of the template contains a list of all entities that were included in the structure. Each entry denotes a separate entity, and includes information about their position in the structure, inventory/equipment, current rotation, and often full NBT data from the entity. Each entry also contains information regarding the plugin that is responsible for interpreting the entity data. If this plugin is not present, the entity will not be generated with the rest of the structure. Due to the complexity of the JSON-NBT data format, it is highly not recommended to do any manual editing of the entities entries.

## Template Packs

Template packs are a collection of structures, usually with a specific theme, packaged together for easier install and use. Most of the template packs available will be for use in world-generation, but some survival mode and creative mode templates/packs may also be available.

Each template pack is simply a .zip file containing a collection of templates, their preview images, and any relevant documentation for that collection. All contents in the pack should be placed in the root directory of the .zip file – internal directories are not supported.

A preview image for each template in the pack may also be optionally included. These images must be 512 x 288 sized .png files, and named the same as the template it represents. If the preview image is present it will be displayed in GUIs that allow selection of structures (Drafting Station, Creative Mode instant construction item).

New template packs may be installed by placing the entire .zip file into the <Minecraft folder>config/ancientwarfare/structures/included/ directory. Any templates or packs found here will be loaded when the game starts up – external changes to these files will require the game be restarted before it will see the changes.

### Default Structure Pack

Ancient Warfare ships with an internal non-modifiable default structure pack. This is a minimal pack that includes a small assortment of templates for both world-gen (for vanilla biomes) and for construction in survival mode via the drafting station.



By default this structure pack is loaded on game start. To disable this behavior you may edit the corresponding line in the AncientWarfareStructures.cfg file (change to load\_default\_structure\_pack=false).

## Creating Templates

The creation of new structure templates is facilitated by an in-game structure scanning tool, the aptly named Structure Scanner. This tool is located in the Ancient Warfare Structures creative tab and is not available in survival mode. It allows full specification of all options and settings for the template to be created – no manual editing of templates should be necessary.

The Structure Scanner is used by defining a bounding box surrounding the structure to be scanned, defining what position should be considered the front entrance/above ground level and what orientation should be considered ‘forward’, and finally by opening the GUI to configure the rest of the template options.

The bounding box for the structure is defined by first targeting one corner of the structure and pressing item activation key 1 (default ‘Z’). You may hold shift to offset by side hit and select the air block in front of/on top of the block targeted. A cube will be rendered denoting the block to be selected. This cube will turn into a dynamic bounding box as soon as the first point is selected, allowing a real time preview of the area that will be scanned.

The second coordinate of the bounding box may be defined in the same manner; target the block at the opposite corner of the structure and press the activation key. As you are moving to select the target the rendered bounding box will be updated to show what will be included in the template.

After the first and second coordinates for the bounding box are specified it is necessary to specify what position will be considered the ‘front door’ / ground-level entrance. This position should be the first air block above ground level at the front edge of the structure; it does not need to be an actual door, but should be the logical front entrance to the structure area. You should face towards the front of the structure when selecting this position, as this selection also determines the build-orientation for the structure. You may hold shift to enable selecting an air block on top of/in front of the block targeted.

As soon as the bounding box and door position have selected you may right-click to open the template settings GUI. From this GUI you may configure all options that are available in the template. See Appendix II – Structure Validation for details regarding each setting.

Once you are satisfied with your settings, enter the desired structure name in the name input box and press the ‘Export’ button. All supported blocks and entities within the denoted bounding box will be scanned and included in the template.

If you opted to include the template immediately in the settings GUI it will be placed in the /included template directory, and immediately available for selection in the instant structure creation tool, world generation (if enabled), and survival mode building (if enabled). If you did not select include immediately, the template will be exported to the /export template directory and will not be added to world gen or survival mode lists.

## Adding New World Generation

Adding individual templates or template packs to the <Minecraft path>/config/ancientwarfare/structures/included/ template directory will result in those templates being loaded into the game. Any loaded templates that are flagged for world generation will be included for selection during world generation.

When adding new templates and packs care must be taken regarding the recommended cluster value and cluster range configuration settings for those packs or structures. Large differences between the current settings and the recommended settings will result in the structure pack or structures not working as intended. Changing your configuration settings to those recommended by the structure pack will usually resolve any issues.

When adding multiple structure packs, large differences between the recommended settings for the structure packs may also cause issues for world generation. Manually editing the templates in the packs to conform to a single range of values is an option, however it is recommended to only use multiple structure packs if they have all been designed to be used together.

Creating new structure packs is a bit more of a complex undertaking. A complete understanding of how the various values contribute to template selection and validation are necessary to create a well behaved structure pack. See Appendices II and III for more details regarding the validation settings and structure selection.

### Additional Mod Support / API

Support for blocks and entities from additional mods may be added to the template system through a programming API / plugin system. Plugins may be written as a stand-alone mod or packaged with the mod they are adding support for. Each plugin may add support for one or more blocks, from one or more mods. Stand-alone plugins should be added to the mods folder like all mods, and will be loaded with the game. Packaged plugins should be loaded at the same time as their packaged mod.

For more details regarding writing plugins see the GitHub repository or contact the author of Ancient Warfare.

### Survival Mode Structure Creation

Structures flagged as enabled for survival mode will be available for construction in the Drafting Station. The Drafting Station itself may be crafted in the Engineering Station, and has research requirements that must be met before it can be crafted.

There are two phases to creating a structure through the Drafting Station. The first phase is resource gathering and compilation. Once a structure is selected in the Drafting Station a list of necessary resources will be displayed. As resources are added to the resource input slots they will be used towards fulfilling the requirements for the template. As soon as all resource requirements have been met a Structure Builder item will be placed in the output slot.

The second phase of structure creation is the actual construction of the structure. Remove the Structure Builder from the output slot of the Drafting Station. As long as you are holding a Structure Builder item a bounding box will be drawn denoting the bounds which the structure would occupy if built at the currently targeted location. Once a suitable location has been located press item activation key 1 (default 'Z') to place the Structure Builder control block. The control block may be picked back up into item form as long as construction has not yet started; once construction has started, breaking the control block will not drop an item – all unplaced resources will be lost.

After the control block has been placed it will need to be powered in order to proceed with construction. It will need to be powered with either NPC workers (work type: Mining) or the Torque power system. If neither the NPC module nor Automation module are installed the control block will not require power and will proceed with construction automatically.

## Gates

Several types of large doors are added, collectively known as Gates. Vertical, Horizontal, and Drawbridge type gates are available. Vertical and Horizontal types are available in two styles, wood and iron. Gates may currently be of any size in width and height, but may only be a single block thick.

### Construction

To construct a gate you must designate the two corners of the space it will occupy when in the vertical/closed state. Each position is specified by pressing item activation key 1 (default 'Z') with the target block highlighted. A block/bounding box should be rendered denoting the space the gate would occupy if constructed.

Once both positions have been selected you may right-click to spawn the gate. For drawbridge types your facing direction when spawning it is important, you should be standing on the 'inside' facing the direction you want the gate to open towards. If the chosen spawning area is clear and everything was setup properly the gate should be created and ready to be used.

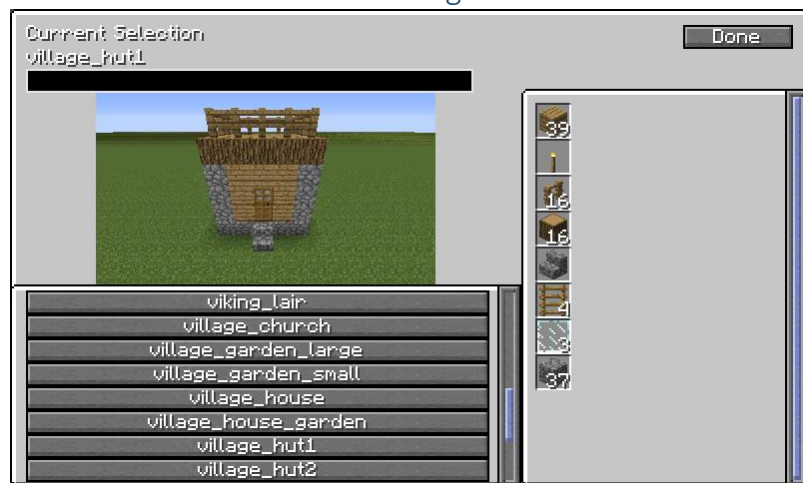
### Interaction

You may right-click on a gate to open or close it. When a vertical gate is in the 'open' position, you must click on the very top of the space it occupies. Alternatively you may input a redstone signal to open or close the gate. Applying a constant redstone signal input will toggle the gate and lock it in its new state; in this manner it is possible to lock a gate from the inside.

## Creative Mode Tools

Various tools are available for aid in structure and map creation while in survival mode. These tools are designed to reduce the tedium of creating large structures and help in the creation of compelling map setups.

### Instant Structure Building Tool



Screenshot of Instant Builder GUI

The Instant Structure Builder does precisely as its name infers. It allows for the instant construction of any loaded structure at a selected position. Right click to open the GUI and select the structure. Press item activation key 1 (default 'Z') to build the selected structure.

## Construction Tool

The construction tool has various modes for filling areas with blocks. You may toggle between these modes with the 'Z' key. The block type to be filled may be selected by pressing 'X' while targeting the intended block. You may press the 'C' key to set or change the first bounding box corner, and 'V' to set or change the second. Finally, 'B' may be pressed to clear the currently selected first and second positions.

Each mode has a different function when activated and some depend upon the position being targeted when activating the item. The modes of operation are:

### *Solid Fill*

Fills the entire selected area with the currently selected block

### *Outline Fill*

Fills the outermost block layer of the currently selected area with the selected block

### *Lake Fill*

Replaces the targeted block and all neighboring and lower blocks with the currently selected block

### *Layer Fill*

Works like lake fill, but only fills the current layer and does not propagate downwards

## Lake Construction Tool

This tool has the single function of 'Lake Fill' from the Construction tool. Fill block is pre-selected as water. Simply right-click it while an air block is highlighted to fill all connected air blocks with water. If in survival mode it may only be used once (e.g. may be used as a reward in chests/structures).

## Vanilla Mob Spawner Placement Tool

This tool allows for placing a vanilla mob spawner in the world, with configurable mob type and other options. Right click to open the GUI and configure the spawner, press 'Z' to place the block once configured. Please validate that the entity can safely be spawned via a vanilla spawner before using this in a cared-for world. Improper use can corrupt a world beyond easy recovery.

## Advanced Spawner Placement Tool

The Advanced Spawner is an item/block that spawns monsters, and allows for much greater control of the spawning parameters. You may set multiple mob types to spawn in a single group, add multiple randomized groups, change spawn group sizes, limit the total number of mobs spawned, add items to be rewarded when the player defeats the spawner, change block hardness, and many other options.

## Structure Module Items

### Structure Scanner

The Structure Scanner allows for creation of new structure templates. See the Template Creation section for more details on its use.

### Structure Builder

The Structure Builder allows for instant construction of structures from templates while in creative mode. See the Creative Mode Tools – Instant Structure Building Tool section for more details on its use.

### Survival Mode Structure Builder

These set of items (one for every survival-mode enabled structure) spawn a creative mode worksite block and bounding box for the designated structure when used. See the Survival Mode Structure Creation section for more details on how to get and use these items.

### Construction Tool

See the Creative Mode Tools – Construction Tool section for more details about this items use.

### Lake Construction Tool

See the Creative Mode Tools – Lake Construction Tool section for details regarding this item.

### Gate Spawning Items

These items spawn the various types and styles of gates. See the Gates section for more information on their use.

### Spawner Placer

See the Creative Mode Tools – Spawner Placer section for details regarding this item.

### Advanced Spawner Placer

See the Creative Mode Tools – Advanced Spawner Placer section for details regarding this item.

## Structure Module Blocks

### Drafting Station

The Drafting Station allows for creation of structure-spawning items while in survival mode. See the Survival Mode Structure Creation section for more details.

### Advanced Spawner

The Advanced Spawner block is a configurable mob spawner. See the Creative Mode Tools – Advanced Spawner section for more details on its use.

### Structure Builder

The Structure Builder block is responsible for managing the construction of a template for structures created through the drafting station. See the Survival Mode Structure Creation section for more details.

### Gate Proxy Block

The Gate Proxy block is an indestructible invisible block with full bounding box and collision. It is used by gates to enable collisions and to stop AI from seeing and path-finding through the entity-based portion of the Gate.



## Section V – Vehicle Module

Instruction Manual WIP

Module not yet implemented.

## Section VI - Credits

### Programming

Shadowmage

### Modeling

Shadowmage

### Textures

Shadowmage

### Sound

Dulciphi (donations of music)

### Languages

Shadowmage45

### Testing

Pelirow

Dulciphi

MohawkyMagoo

Varkas

Me2003

Nozgo

### Structures

MohawkyMagoo

Shadowmage45

Pelirow

Dulciphi

### NPC Skins

Shadowmage

### Financial Supporters (donators)

Dulciphi

Matthew / Brittany Holden

## Section VII – Appendices

### Appendix I – Pre-Defined Skin Pack Names

The following names are pre-defined in the NPC skin-pack system. Specifying one of these names for the skin name in the skin-pack.meta file will add the specified texture to the random selection pool for the given NPC type.

Pre-Defined names:

- courier – used by player owned Courier
- trader – used by player owned Trader
- priest – used by player owned Priest
- bard – used by player owned Bard
- worker – used by un-assigned player owned Worker NPCs
- worker.researcher – used by the Researcher Worker role.
- worker.craftsman – used by the Craftsman Worker role
- worker.farmer – used by the Farmer Worker role
- worker.miner – used by the Miner Worker role
- worker.lumberjack – used by the Lumberjack Worker role
- combat – used by un-assigned player owned Combat NPCs
- combat.soldier – used by the Soldier Combat role
- combat.archer – used by the Archer Combat role
- combat.medic – used by the Medic Combat Role
- combat.engineer – used by the Engineer Combat Role
- combat.commander – used by the Commander Combat Role
- bandit.archer – used by Bandit Archers
- bandit.archer.elite – used by Elite Bandit Archers
- bandit.soldier – used by Bandit Soldiers
- bandit.soldier.elite – used by Elite Bandit Soldiers
- bandit.leader – used by Bandit Leaders
- bandit.leader.elite – used by Elite Bandit Leaders
- bandit.trader – used by Bandit Traders
- bandit.priest – used by Bandit Priests
- bandit.cavalry – used by Bandit Mounted Soldiers
- bandit.mounted\_archer – used by Bandit Mounted Archers
- bandit.civilian.male – used by Bandit Civilian Males
- bandit.civilian.female – used by Bandit Civilian Females
- pirate.archer - used by Pirate Archers
- pirate.archer.elite - used by Pirate Elite Archers
- pirate.soldier - used by Pirate Soldiers
- pirate.soldier.elite - used by Pirate Elite Soldiers
- pirate.leader - used by Pirate Leaders
- pirate.leader.elite - used by Pirate Elite Leaders
- pirate.trader - used by Pirate Traders
- pirate.priest - used by Pirate Priests

- pirate.cavalry - used by Pirate Mounted Soldiers
- pirate.mounted\_archer - used by Pirate Mounted Archers
- pirate.civilian.male - used by Pirate Civilian Males
- pirate.civilian.female - used by Pirate Civilian Females
- viking.archer - used by Viking Archers
- viking.archer.elite - used by Viking Elite Archers
- viking.soldier - used by Viking Soldiers
- viking.soldier.elite - used by Viking Elite Soldiers
- viking.leader - used by Viking Leaders
- viking.leader.elite - used by Viking Elite Leaders
- viking.trader - used by Viking Traders
- viking.priest - used by Viking Priests
- viking.cavalry - used by Viking Mounted Soldiers
- viking.mounted\_archer - used by Viking Mounted Archers
- viking.civilian.male - used by Viking Civilian Males
- viking.civilian.female - used by Viking Civilian Females
- desert.archer - used by Desert Native Archers
- desert.archer.elite - used by Desert Native Elite Archers
- desert.soldier - used by Desert Native Soldiers
- desert.soldier.elite - used by Desert Native Elite Soldiers
- desert.leader - used by Desert Native Leaders
- desert.leader.elite - used by Desert Native Elite Leaders
- desert.trader - used by Desert Native Traders
- desert.priest - used by Desert Native Priests
- desert.cavalry - used by Desert Native Mounted Soldiers
- desert.mounted\_archer - used by Desert Native Mounted Archers
- desert.civilian.male - used by Desert Native Civilian Males
- desert.civilian.female - used by Desert Native Civilian Females
- native.archer - used by Jungle Native Archers
- native.archer.elite - used by Jungle Native Elite Archers
- native.soldier - used by Jungle Native Soldiers
- native.soldier.elite - used by Jungle Native Elite Soldiers
- native.leader - used by Jungle Native Leaders
- native.leader.elite - used by Jungle Native Elite Leaders
- native.trader - used by Jungle Native Traders
- native.priest - used by Jungle Native Priests
- native.cavalry - used by Jungle Native Mounted Soldiers
- native.mounted\_archer - used by Jungle Native Mounted Archers
- native.civilian.male - used by Jungle Native Civilian Males
- native.civilian.female - used by Jungle Native Civilian Females
- custom\_1.archer - used by Custom 1 Archers
- custom\_1.archer.elite - used by Custom 1 Elite Archers
- custom\_1.soldier - used by Custom 1 Soldiers
- custom\_1.soldier.elite - used by Custom 1 Elite Soldiers

- custom\_1.leader - used by Custom 1 Leaders
- custom\_1.leader.elite - used by Custom 1 Elite Leaders
- custom\_1.trader - used by Custom 1 Traders
- custom\_1.priest - used by Custom 1 Priests
- custom\_1.cavalry - used by Custom 1 Mounted Soldiers
- custom\_1.mounted\_archer - used by Custom 1 Mounted Archers
- custom\_1.civilian.male - used by Custom 1 Civilian Males
- custom\_1.civilian.female - used by Custom 1 Civilian Females
- custom\_2.archer - used by Custom 2 Archers
- custom\_2.archer.elite - used by Custom 2 Elite Archers
- custom\_2.soldier - used by Custom 2 Soldiers
- custom\_2.soldier.elite - used by Custom 2 Elite Soldiers
- custom\_2.leader - used by Custom 2 Leaders
- custom\_2.leader.elite - used by Custom 2 Elite Leaders
- custom\_2.trader - used by Custom 2 Traders
- custom\_2.priest - used by Custom 2 Priests
- custom\_2.cavalry - used by Custom 2 Mounted Soldiers
- custom\_2.mounted\_archer - used by Custom 2 Mounted Archers
- custom\_2.civilian.male - used by Custom 2 Civilian Males
- custom\_2.civilian.female - used by Custom 2 Civilian Females
- custom\_2.archer - used by Custom 2 Archers
- custom\_3.archer.elite - used by Custom 3 Elite Archers
- custom\_3.soldier - used by Custom 3 Soldiers
- custom\_3.soldier.elite - used by Custom 3 Elite Soldiers
- custom\_3.leader - used by Custom 3 Leaders
- custom\_3.leader.elite - used by Custom 3 Elite Leaders
- custom\_3.trader - used by Custom 3 Traders
- custom\_3.priest - used by Custom 3 Priests
- custom\_3.cavalry - used by Custom 3 Mounted Soldiers
- custom\_3.mounted\_archer - used by Custom 3 Mounted Archers
- custom\_3.civilian.male - used by Custom 3 Civilian Males
- custom\_3.civilian.female - used by Custom 3 Civilian Females

## Appendix II – Structure Validation Types

WIP

Will be adding more details regarding each validation type, and each validation setting

Multiple validation types are available for structure templates for world generation. Each has a specific set of options for validation. The types of structure validation, their intended uses, and type-specific options are:

### Ground

Validates structures for spawning at ground level. Checks the surrounding area for level-ness and invalid target blocks. Includes options for post-generation smoothing (border leveling/clearing).



## Harbor

Validates a structure for spawning at ground level while straddling the border between land and water. The front edge of the structure will be validated to be over land, and the rear edge validated to be over water; sides are validated to be either water or land. All land blocks are validated for leveling/clearing and proper target block types.

## Island

Validates a structure for spawning as an island in the ocean. Validates ocean depth. Once constructed the validator will generate land to go from the ocean floor to the bottom of the template.

## Sky

Validates a structure for spawning in the air. Validation options include min and max absolute height as well as above-ground min and max heights. Designed for generation of 'flying' structures (sky temple, cloud realms...whatever).

## Underground

Validates a structure for spawning entirely underground, similar to the MC dungeons. Validates that the entire structure is at least a certain distance under the ground. Has options to determine min and max generation depths.

## Underwater

Validates a structure for spawning entirely under the water. Has options for min and max ocean depth and minimum overhead water level (to ensure fully submerged).

## Water

Validates a structure for spawning on the surface of water. Has options for min and max water depth (to validate the water is deep enough for any submerged portion of the structure).

Several options are shared amongst all validation types. These options are:

### worldGenEnabled

True/false value for if the structure should attempt to generate during world-gen and chunk gen.

### unique

True/false value for if the structure should only generate a single time, ever.

### preserveBlocks

True/false value for if existing blocks should be left in-place when the template contains an air-block for that position.

### selectionWeight

This value determines how often a particular template would be selected out of all potentially valid templates for any given position. Higher values will result in the template being chosen more often. Specific percentage/ratio will vary per-setup AND per-position being generated (as each position may have a different number of potentially valid structures).

For best use this value needs to be adjusted by the difficulty of the validation for a structure. The stricter the validation is, the higher the 'weight' should be for a structure. Generally the validation will become stricter/more difficult for large structures (i.e. large structures should be given additional weight value to compensate for their size).

#### clusterValue

This value is used to determine if a particular structure is capable of spawning in an area by examining all existing local structures for their clusterValue. If the found value + the new template value is greater than the maxClusterValue in the structure module config file, the template will NOT be included for further selection.

Using these mechanisms allows for setting some structures to have a higher value than others, which can aid in placing lower-value structures around higher-value ones.

#### minDuplicateDistance

If set to higher than 0, this value will be used to determine if a structure may have any duplicates (of any orientation) located within this distance. If duplicates are found within the specified distance, the structure will not be included for further selection/validation.

#### dimensionWhiteList

True/false value for if the dimension list should be treated as a whitelist (true), or blacklist (false).

If set to true, the template will only generate in the dimensions specified under dimensionList.

If set to false the template will only generate in the dimensions NOT specified under dimensionList.

#### dimensionList

The list of dimensions to be used for whitelist/blacklisting. Use of white/black list is determined by the dimensionWhiteList setting.

#### biomeWhiteList

True/false value for if the biomeList should be treated as a whitelist (true), or blacklist (false).

If set to true, the template will only generate in the biomes specified under biomeList.

If set to false the template will only generate in the biomes NOT specified under biomeList.

#### biomeList

The list of biomes to be used for whitelist/blacklisting. Use of white/black list is biomes by the biomeWhiteList setting.

#### maxLeveling

This setting determines how uneven the ground may be extending above the chosen generation point ground-level. If the found unevenness is greater than this value, the template will not be validated, and nothing will be generated.

If borderSize >0, this setting also determines the maximum height that border leveling/smoothing operations will occur.

### maxUnderfill

This setting determines how uneven the ground may be extending below the chosen generation point ground-level. If the found unevenness is greater than this value the template will not be validate, and nothing will be generated.

If borderSize >0, this setting also determines the maximum depth that border fill/smoothing operations will occur.

### borderSize

This setting determines how far out from the template bounds to extend the maxLeveling and maxUnderfill checks during validation.

If found to be valid this setting also determines how far out border leveling and fill operations will occur.

### blockSwap

True/false value for if the structure should use the vanilla block swapping (e.g. desert villages use sandstone). This setting –also- enables the MinecraftForge block-swap event, which any mod may use to modify the placed blocks for a template. This setting only effects world-gen structures, and has no effect if the structure is built in survival mode (i.e. through the drafting station).

## Appendix III – World Generation Structure Selection

### WIP

This section will contain detailed information regarding how structures are selected during world generation; the aim is to better inform structure and structure pack designers so they may create more balanced packs.

## Appendix IV – Bard Tune Info

Ancient Warfare ships with a small selection of sound files specifically for use with the Bard NPC type. The name and play-length of these files are (length specified in minutes, decimal format, can input this value for the ‘length’ in the Bard GUI):

- ancientwarfare:bard.tune1 – 1.23
- ancientwarfare:bard.tune2 – 0.91
- ancientwarfare:bard.tune3 – 0.53
- ancientwarfare:bard.tune4 – 0.45
- ancientwarfare:bard.tune5 – 0.32
- ancientwarfare:bard.tune6 – 0.48