



Table of Contents

- 1. Introduction**
- 2. Quick Start / Installation Guide**
- 3. Core Features**
- 4. Customization Options**
 - Woodcutting Settings
 - Skill Level Settings
 - Inventory Settings
 - Wood, Loot Items, and Axes
 - Trees
- 5. Integration / Advanced Usage**
 - Inventory System
 - Equipment System
 - Save / Load System
- 6. Quick Reference Guide**
- 7. Support & Contact**



Introduction

The Woodcutting System by Unreal Toolkit is a fully modular and highly customizable woodcutting system designed for smooth integration into any Unreal Engine project. It comes with a basic inventory and equipment system that can be easily swapped for your own custom systems, ensuring smooth implementation into existing gameplay mechanics.

Woodcutting mechanics are fully customizable, supporting the option for both per-hit harvesting and experience gain as well as on-empty resource collection. The system also includes optional skill levels, which can be toggled on or off to match your game's progression system.

The system provides immediate functionality and prototyping capabilities with six trees, each with a full tree mesh, trunk mesh and tree top mesh, and corresponding harvested wood and twig item, as well as three axes. These included assets allow you to start using the system right away, while the data tables are designed for easy expansion, making it simple to add custom trees, wood, tools, and loot.

With intuitive settings and plug-and-play functionality, the Woodcutting System provides a complete foundation for chopping down trees while remaining flexible for customization and expansion. Integration is simple and efficient.

Woodcutting System

Unreal Engine 5



Unreal Toolkit



Quick Start / Installation Guide

Follow these steps to quickly set up and test the Woodcutting System in your project:

1 Migrate the Woodcutting System into Your Project

- In the **Content Browser**, locate the **Woodcutting** folder included in the asset pack.
- Right-click the folder and select **Migrate**.
- In the pop-up window, ensure all necessary assets are selected (**Uncheck the Demo Folder**), then click **OK**.
- Choose your project's **Content** folder as the migration destination.

2 Add Woodcutting Functionality to Your Player Character

- Open your **Player Character Blueprint**.
- In the Components panel, click **Add Component** and search for **AC_Woodcutting**.
- Select **AC_Woodcutting** to add it to your character.

3 Place a choppable tree in the World

- In the **Content Browser**, locate **BP_Tree**.
- Drag **BP_Tree** into the level.
- In the **Details Panel**, set the **TreeID** to determine the type of tree.

4 Place an axe in the world

- Locate **BP_WoodcuttingItem** in the **Content Browser**.
- Drag **BP_WoodcuttingItem** into the level.
- In the **Details Panel**, set the **ItemID** to an axe (e.g., "Standard Axe").

5 Test the Woodcutting System

- Press **Play** to start the game.
- Equip the **Axe** from the inventory.
- Approach the **Tree** and **strike it with the axe by facing it and clicking the mouse**.
- Observe the woodcutting mechanics in action, including **loot drops, dynamic tree falling, and experience gain**.

Congratulations! The Woodcutting System is now active in your project. From here, you can customize settings, expand the data tables, and refine the mechanics to fit your game.



Core Features

◆ Fully Modular & Data-Driven System

- Uses data tables for defining tree types, axes, loot drops, skill levels, and woodcutting mechanics.
- Easily expandable—add new trees, tools, and loot with minimal effort.

◆ Customizable Woodcutting Mechanics

- Supports harvesting per hit and/or on empty to fit different gameplay styles.
- Depletion settings allow trees to:
 - Dynamically fall upon depletion.
 - Leave a tree trunk behind.
 - Fully disappear.
 - Or not change at all.

◆ Skill & Progression System (Optional)

- Experience gain per hit or when the tree is depleted.
- Skill levels for Woodcutting, which can be enabled or disabled in settings.
- Axes can have woodcutting level requirements and bonus yield to encourage progression.
- Trees can have woodcutting level requirements to be cut down.

◆ Dynamic Loot System

- Customizable additional loot drops with rarity-based drop chances.
- Chance-based spawning allows for rare drops

◆ Pre-Configured & Customizable Assets

- 6 trees, each with a corresponding harvested wood item.
- 3 axes, each with adjustable woodcutting efficiency and skill requirements.
- 6 additional loot items, each tree has an associated twig item as a trash drop.
- Easy to expand with custom assets

◆ Visual & Audio Effects for Immersion

- Niagara FX for woodcutting impact, loot spawn, and level up effects.
- Customizable sound effects for axe swings, axe hits, wood spawning, and tree depletion.
- Optional screen shake for an added sense of impact.

◆ Easy Integration with Existing Systems

- Comes with a basic inventory system and equipment system that can be easily swapped out for your own, or integrated with your own systems.



Customization Options

The Woodcutting System offers extensive customization options, allowing developers to tailor the mechanics, visuals, and overall experience to fit their game. Below is a breakdown of the key settings available in AC_Woodcutting, DT_Trees, DT_WoodcuttingItem, and Inventory Settings.

Woodcutting Settings (Woodcutting Component Settings)

The AC_Woodcutting component, added to the player, provides settings for woodcutting behavior, animations, and experience gain.

- Starter Inventory - Select ItemID's that will be added to the player inventory at runtime.
- StarterEquippedAxe - Optionally select an axe that the player will automatically equip at runtime.
- Harvesting Style – Choose between two harvesting methods:
 - Spawn in World – Wood and additional loot spawn in the world.
 - Auto Add to Inventory – Harvested wood is added directly to the player's inventory.
- Notification Settings - Enable/disable notifications all together or edit the options per notification, toggling specific notifications on or off, and setting custom notification text
- Visual FX - Optional screen shake for successful woodcutting hits

WOODCUTTING SETTINGS	
▼ Woodcutting Settings	
StarterInventory	0 Array element ⊕ 🗑
▼ StarterEquippedAxe	
Data Table	DT_WoodcuttingItems ▼ ↶ 🗑
Row Name	StandardAxe ▼
HarvestingStyle	SpawnInWorld ▼
▼ WoodcuttingNotificationSetti...	
UseWoodcuttingNotificati...	☒
▶ NotificationSettings	
UseLevelUpNotification	☒
LevelUpNotification	Woodcutting skill increased to 🚩
UseTreeEmptyNotification	☒
TreeEmptyNotification	is empty. 🚩
UseRequiredLevelNotifica...	☒
RequiredLevelNotification	Requires woodcutting level 🚩
▼ VisualFx	
EnableScreenShakeOnHit	☒
ScreenShake	CS_Woodcutting- ▼ ↶ 🗑 ⊕ ✕



Skill Level Settings (AC_Woodcutting)

- Toggle whether or not to use the skill level system
- Toggle the use of the progress bar
 - Set progress bar settings to determine location, bar color, text settings, icon, etc.
 - Set the starting level for the woodcutting skill, as well as the maximum level
 - Set the XP required to level up
 - The experience system works with a multiplier to increase the amount of XP needed to level up with each level gained. Customize the multiplier to adjust how much the amount of XP needed for the next level changes.
 - Toggle the use of a level up sound effect, and select custom sound effects from a drop down menu
 - Toggle the use of the level up visual effect, and select custom visual effects from a drop down menu of niagara effects

▼ SkillLevelSettings		
UseSkillLevels	☒	
DisplayProgressBar	☒	
► ProgressBarSettings		
StartingLevel	1	
MaxLevel	100	
StartingXpRequire...	100.0	
LevelUpRequiredX...	1.1	
UseLevelUpSound	☒	
► LevelUpSound		
UseLevelUpEffect	☒	
LevelUpEffect		N_M ▼  

The Progress Bar widget

WBP_WoodcuttingSkillBar updates itself and can be placed anywhere. You can place it in any custom widget and it will update itself.



Inventory Settings (AC_Woodcutting)

The woodcutting system uses a default inventory system that can be swapped out for your own custom inventory system if desired. The default inventory system is fully customizable.

- Toggle whether or not to use the default inventory system
- Toggle the use of the Inventory Button widget and select an icon for the inventory button.
- Notifications:
 - Customize Pickup Notification Text (e.g., "+1 [ItemName]").
 - Adjust Notification Duration.
- Full customization settings for inventory widget aesthetics
 - Pickup item from world widget
 - Inventory widget
 - right-click menu
- Set inventory notification settings, determine notification text, duration, and aesthetic
- Pickup & Drop Settings
 - Enable/disable Pickup Animation when collecting items, and set custom animations from a drop down menu.
 - Enable/disable Inventory Sound Effects for adding/removing items, and set custom sound effects from a drop down menu.

▼ INVENTORY SETTINGS		
Use Default Inventory	☒	
▼ Inventory Settings		
DisplayInventoryButton	☒	
InventoryButtonImage		T_InventoryButton ▼  
► InventoryDisplaySettings		
► PickupWidgetSettings		
► RightClickMenuSettings		
► TooltipSettings		
► NotificationSettings		
► AnimationSettings		
► SoundSettings		



Equipment Settings (AC_Woodcutting)

- Toggle whether or not to use the default equipment system
- Set the main hand socket data for your custom character mesh - this is where the axe will visibly appear on your character.
 - Set the bone name
 - Set the local location and rotation for the socket to fine tune the positioning of the equipped axe in your character's hand.
- Set the axe swing animation
 - Set the delay from the start of the swing animation to the start of the axe hit trace, allowing you to fine tune the swing mechanics
- Toggle the use of an axe swing sound and set the swing sound
- Toggle whether or not the equipped axe widget should hide itself if no axe is equipped.
 - Fully customize the aesthetics of the widget

▼ EquipSettings	
UseDefaultEquipmentMa...	☒
▼ MainHandSocket	
BoneName	hand_r
▶ SocketLocation	-7.903017 7.118949 -19.443243
▶ SocketRotation	-12.323198 5.496126 ° -0.791372 °
SwingAnimation	 AM_AxeSwing ▼
StartTraceDelay	0.6
UseSwingSound	☒
▼ SwingSound	
Sound	 A_Swing ▼
Volume Multiplier	1.0
▶ EquippedAxeWidgetSettin...	

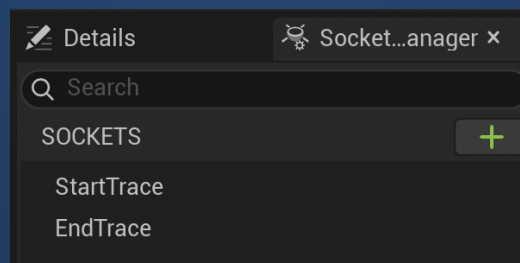
The equipped axe widget **WBP_EquippedAxe** updates itself and can be placed anywhere. You can place it in any custom widget and it will update itself and remain functional.



Preparing Custom Axe Meshes

The Woodcutting system uses sockets added to the axe meshes to determine where to detect if the axe hits a tree during the swing animation.

- Open your axe mesh
- In the details panel, click the Socket Manager tab and click + to add a socket
 - The name of the sockets **must match exactly** to the socket names shown
 - **StartTrace**
 - **EndTrace**
- Position the sockets where you want them on your mesh. The top and bottom of the blade of the axe, as shown in the example.





Wood, Loot Items, and Axes

The DT_WoodcuttingItem data table controls wood and item data, as well as axes and their stats. To add items, **open DT_WoodcuttingItems** and **click +ADD** to add a new row. Fill in the item data to add your item to the system.

Base Item Data (DT_WoodcuttingItem)

- ItemID - A reference to the data table row.
- Name - The item's name
- Description - A description of the item. Will appear in item tooltips and other UI.
- Icon - The icon/image for the item
- Static Mesh - The item's mesh
- Weight - The physical weight of the item
- Value - The item's monetary value
- Quantity - The quantity that 1 instance represents
- CanStack? - Toggles if the item can stack
- MaximumStackSize - Sets the maximum stack size
- CanPickupFromWorld - Enables/disables the ability to pick up the item from the world
- CanDestroy - Enables/disables the ability to destroy the item from the inventory
- CanRemoveFromInventory - Enables/disables the ability to remove the item from the inventory by dropping or destroying it.
- PhysicsEnabled? - Toggles physics for the item when it's in the world.

Axe Data (DT_WoodcuttingItem)

- RequiredSkillLevel - Sets the minimum skill level needed to equip the axe
- IncreaseMinWoodYield - Increases the minimum amount of wood yielded from the tree by the set amount
- IncreaseMaxWoodYield - Increases the maximum amount of wood yielded from the tree by the set amount

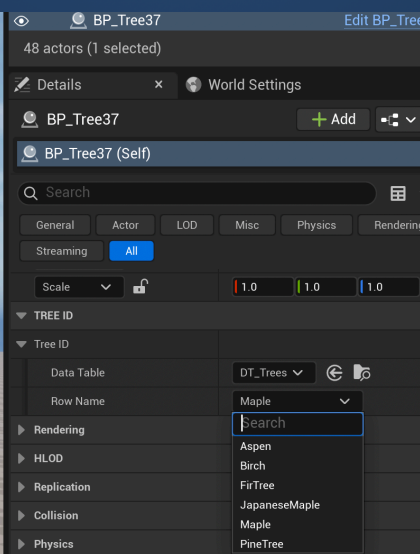
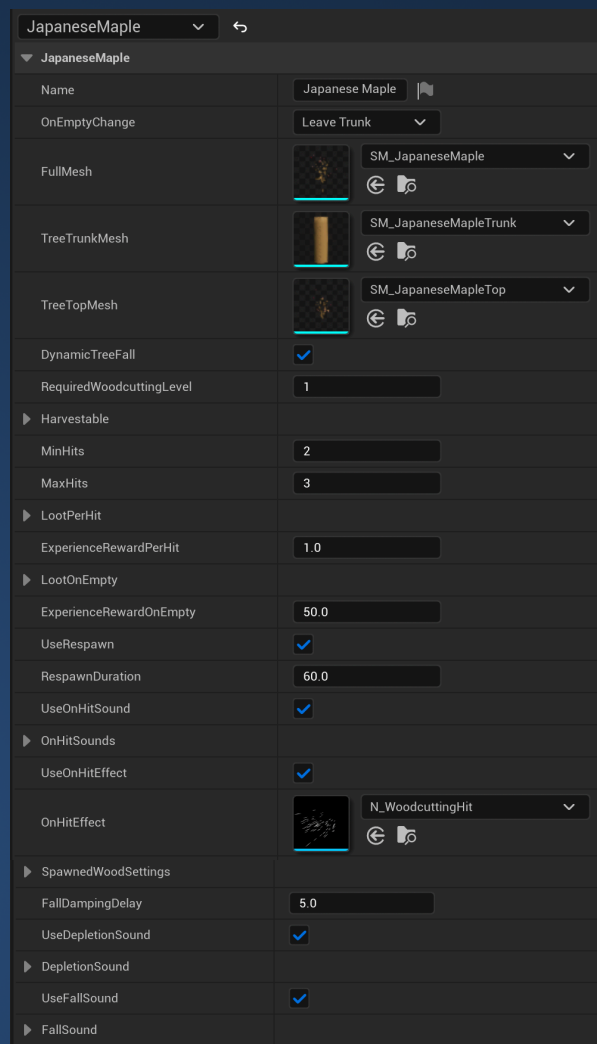
▼ StandardAxe	
▼ BaseItemData	
ItemID	
Data Table	DT_WoodcuttingItems ▼ ↺ ↻
Row Name	StandardAxe ▼
Name	Standard Axe 🚩
Description	A standard axe for chopping down trees. 🚩
Category	Tool ▼
Icon	 T_StandardAxe ▼ ↺ ↻
StaticMesh	 SM_Axe ▼ ↺ ↻
Weight	2.5
Value	45
Quantity	1
CanStack?	☐
MaximumStackSize	1
CanPickupFromWorld	☒
CanDestroy?	☒
CanRemoveFromInventory	☒
PhysicsEnabled?	☒
▼ AxeData	
IsAxe	☒
RequiredSkillLevel	0
IncreaseMinWoodYield	0
IncreaseMaxWoodYield	0



Trees(DT_Trees)

The DT_Trees data table controls the properties of different trees.

- Set the name of the tree
- Set the depletion behavior with OnEmptyChange
 - Leave Trunk - leaves a tree trunk mesh in the trees place
 - Destroy Mesh - Removes the tree upon depletion
 - No Change - Keeps the tree visually intact.
- Dynamic Fall - Toggle whether the tree should dynamically fall once it's depleted.
- Set the harvestable item with the harvestable Item ID.
- Skill Level Requirement – Set a required Woodcutting Level for the tree.
- Min/Max Yield Per Hit – Define the amount of wood collected per axe hit.
- Additional Loot System:
 - Enable/disable Chance-Based Additional Loot.
 - Define a list of extra items that may drop, each with:
 - Drop Chance (%).
 - Min/Max Quantity.
- Set a respawn duration, after the determined time the tree will respawn
- Toggle the use of hit sounds, and select the hit sounds for the tree
- Toggle the use of hit effects, and select a niagara effect to play at the impact point of the hit
- Customize the spawned wood settings
 - Set a range for how far loot can spawn from the hit point
 - Set scaling style for the spawned wood
 - Set sound and effects for wood spawning in the world



Add Trees To The World

- Drag BP_Tree into the world
- With it selected, find Tree ID in the details panel
- Select the row name of the tree you're adding

The tree will automatically update based on the data in DT_Tree



Integration / Advanced Usage

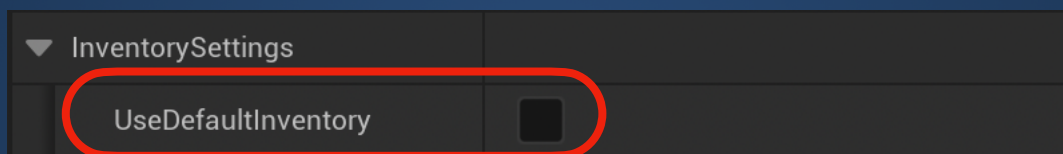
The Woodcutting System is designed to be modular and easy to integrate with other inventory and/or equipment systems. If you are using a custom inventory system, you can override the built-in functions to ensure seamless integration.

Integrating a Custom Inventory System

By default, the woodcutting system includes AC_DefaultInventory, a basic inventory setup designed for easy testing and quick implementation. However, if you are using your own inventory system, you will need to override a few key functions to replace the default item management logic with your own.

Overriding The Inventory

In AC_Woodcutting there are the following inventory related functions that you can easily replace with your own Inventory functions. Open each of the functions below in AC_Woodcutting, and replace the logic with your inventory functions. If you are integrating with your own custom inventory, ensure to uncheck “UseDefaultInventory” found in the Inventory Settings in AC_Woodcutting. (From your player character blueprint, select AC_Woodcutting, and find the settings in the details panel.)

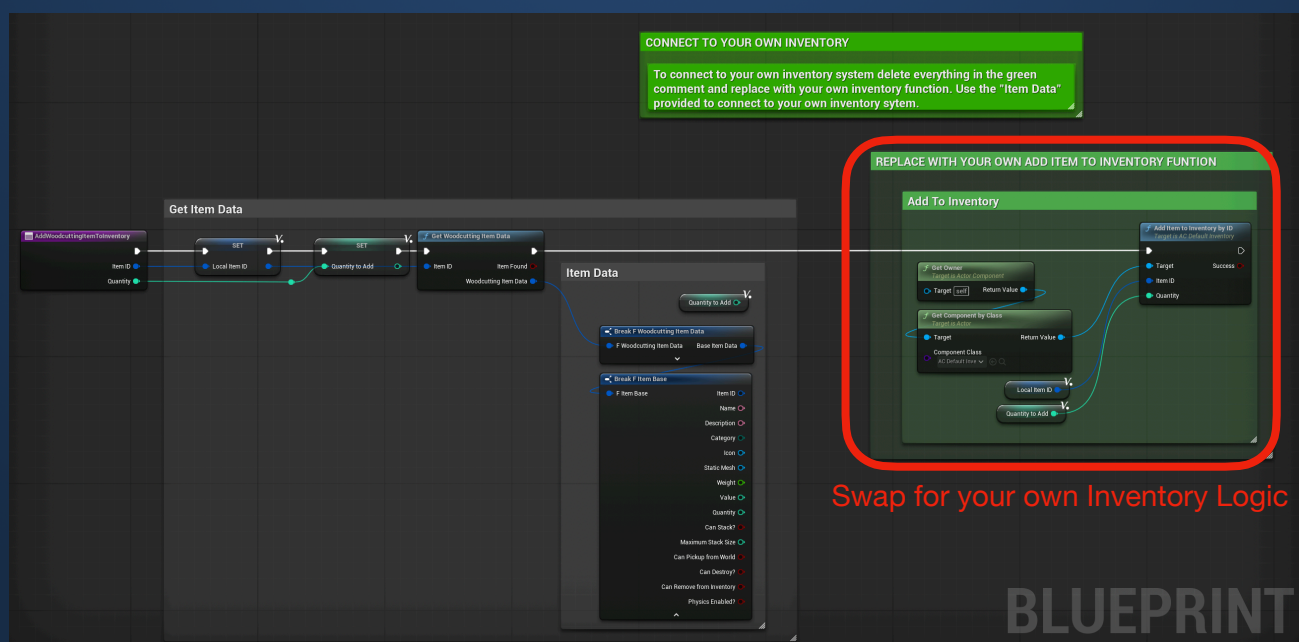


The Inventory functions are set up to expose the item data for easy integration. Replace the green commented section with your own inventory function. Use the exposed Item Data to hook into your inventory function.

Do this for the two inventory functions:

AddWoodcuttingItemToInventory

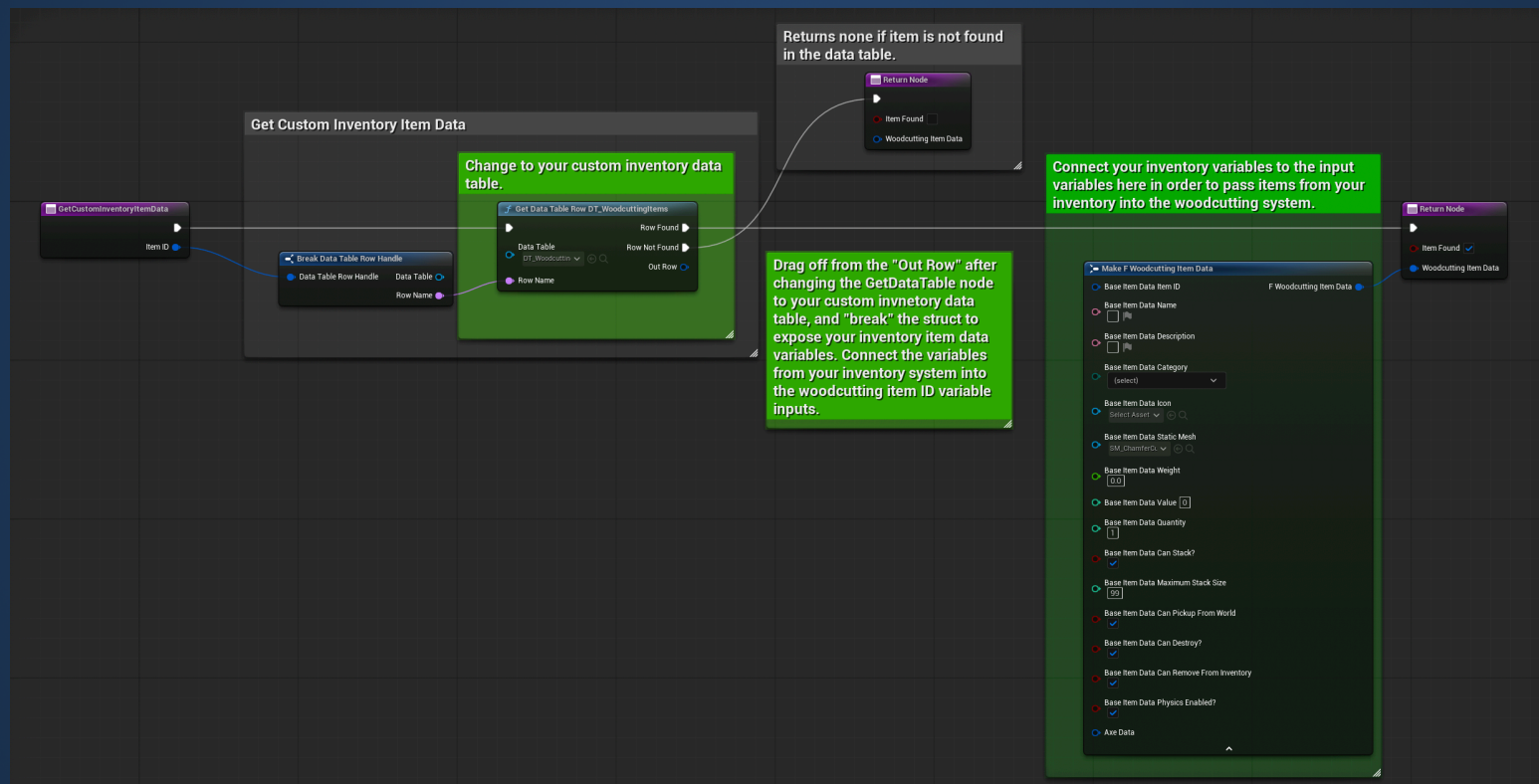
RemoveWoodcuttingItemFromInventory





To make the woodcutting system able to read item data from your custom inventory, you must edit the **GetCustomInventoryItemData** function found in **BPFL_CustomInventoryIntegration**.

- Open **BPFL_CustomInventoryIntegration**
- Select the function **GetCustomInventoryItemData**
- Change the Get data table row node to your custom inventory data table.
- Drag off from the out row and “break” to expose your custom item data.
- Connect your custom item data variables to the corresponding variables in the “MakeWoodcuttingItemData” section of the function.



This allows you to set your own custom items as loot items. After doing this you will still be able to use DT_Woodcutting Items to add new wood/items/axes to the system.

After editing those **3 inventory functions** the woodcutting system will work with your inventory by passing item data from DT_WoodcuttingItem into your custom inventory system, and item data from your custom inventory system into the woodcutting system.

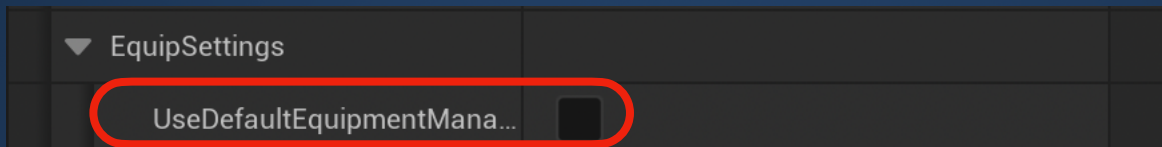


Integrating a Custom Equipment System

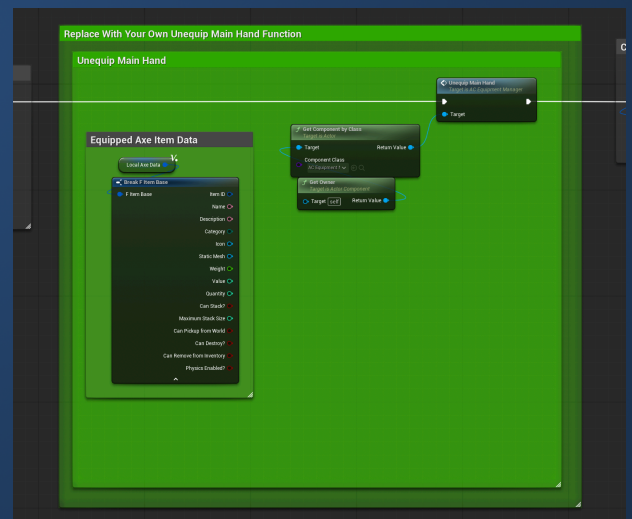
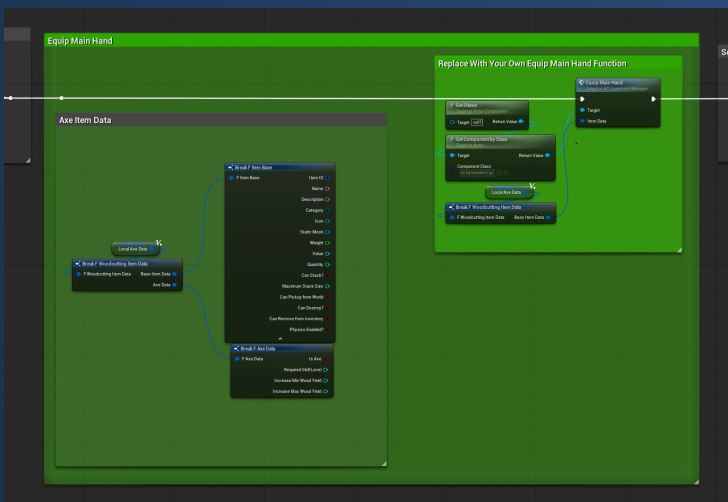
By default, the woodcutting system includes AC_DefaultEquipmentManager, a basic equipment system designed for easy testing and quick implementation. However, if you are using your own equipment system, you will need to override a few key functions.

Overriding The Equipment System

In AC_Woodcutting there are the following equipment related functions that you can easily replace with your own equip functions. Open each of the functions below in AC_Woodcutting, and replace the logic with your equip functions. If you are integrating with your own custom equipment system, ensure to uncheck “UseDefaultEquipmentManager” found in the Equipment Settings in AC_Woodcutting. (From your player character blueprint, select AC_Woodcutting, and find the settings in the details panel.)



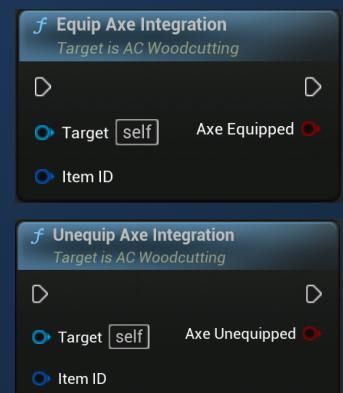
The EquipAxe and UnequipAxe functions are set up for you to edit. Open AC_Woodcutting, and edit only the green commented areas of the EquipAxe and UnequipAxe functions. The woodcutting system handles the logic for marking “AxeEquipped”, you’re overriding the part of the function that visibly attaches the item mesh to your player, and marking something like “MainHandEquipped” in your custom equipment system. This will allow the woodcutting system to communicate with your custom equipment system whenever the woodcutting system calls its equip/unequip functions.



In your own custom equipment system, add the “EquipAxe_Integration” function into your equip function. This function will check if the item is an axe, and if so it will mark “AxeEquipped” in the woodcutting system. You can add this at the end of your custom equip function.

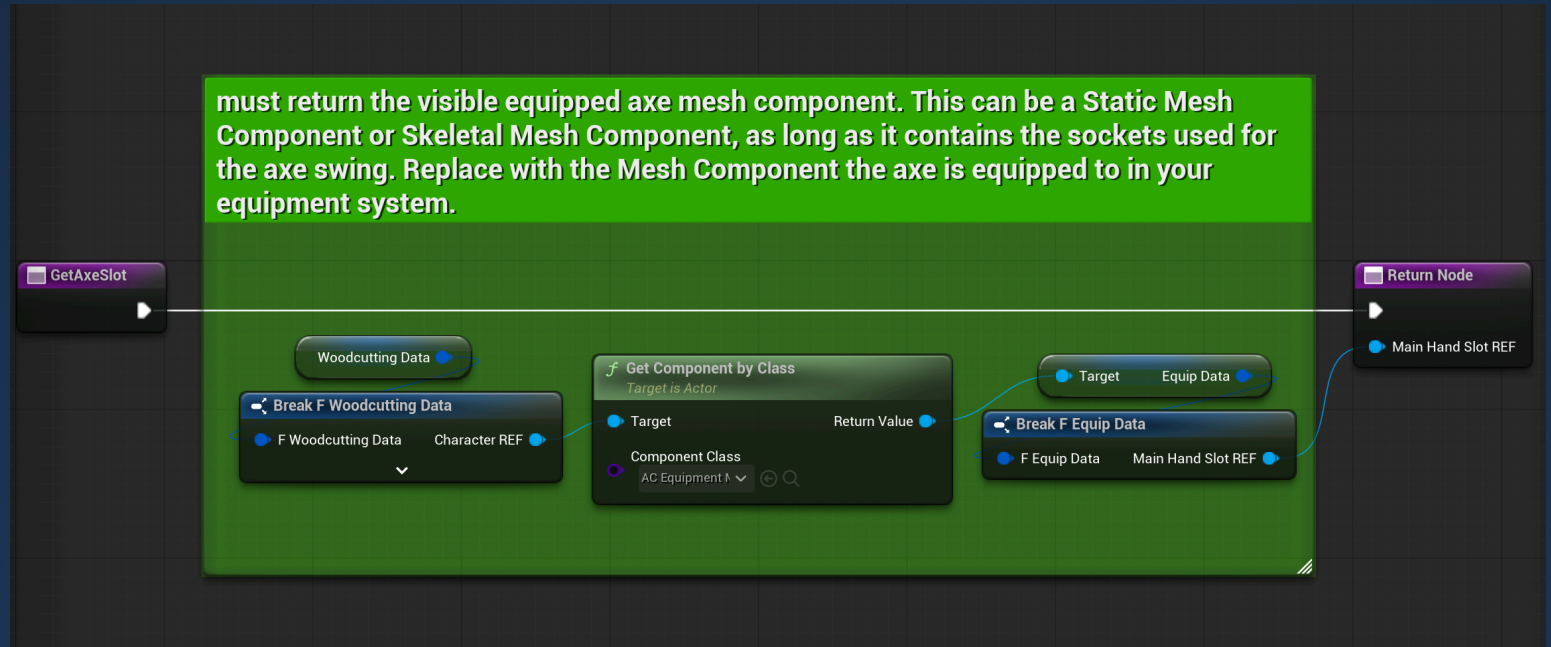
In your unequip function, add the “UnequipAxe_Integration”. This function will check if the item being unequipped is an axe, and if so it will uncheck “AxeEquipped” in the woodcutting system.

For both of these functions, feed in the ItemID (Data table row) of the item being equipped or unequipped.





The Woodcutting system sometimes needs a reference to the axe slot, or main hand slot of the player character. To do so it uses an overridable function “GetAxeSlot”, that you can edit to return a reference to the slot being used from your custom equipment system. This reference can be either a static mesh component or a skeletal mesh component.



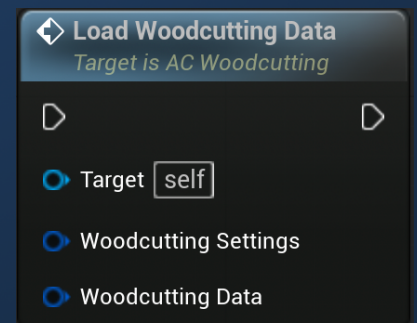
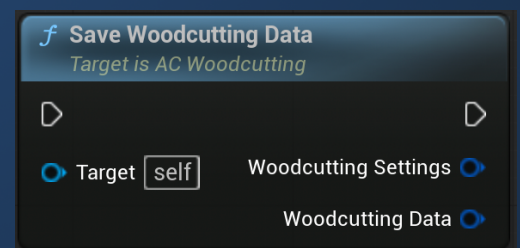
Editing these equipment functions and adding the integration functions to your custom equipment system allows the Woodcutting System to integrate seamlessly with your custom equipment system. This enables your equipment system to manage equipped and unequipped states independently while still allowing the Woodcutting System to communicate with it when needed.

Integrating Save/Load Functions With Your Save System

The woodcutting system uses two variables to store all data. WoodcuttingSettings, and WoodcuttingData. AC_Woodcutting includes a SaveWoodcuttingData and LoadWoodcuttingData function to make it easy to set up with your save system.

In your Save Game Blueprint, in your save function, call the SaveWoodcuttingData function, and promote the two outputs (WoodcuttingSettings, WoodcuttingData) to variables within your save game blueprint. Make sure that in the save game function, you are setting these variables from the output of the SaveWoodcuttingData function. This will store the data in your save game file.

In your Load Game function, after accessing the save game file, pull off of it and drag the WoodcuttingData and WoodcuttingSettings variables into the LoadWoodcuttingData function. This will ensure that the woodcutting data and settings are loaded from your saved data.





Quick Reference Guide

Actor Components

AC_Woodcutting	The component that adds woodcutting functionality to the player character. Add it to your player character blueprint.
AC_Default Inventory	The component that adds inventory functionality to the player character. AC_Woodcutting will add this automatically if you have "UseDefaultInventory" set to true.

Blueprints

BP_Tree	The blueprint that represents trees in the world. Drag into the world and set the TreeID to set the tree.
BP_WoodcuttingItem	The blueprint that represents items in the world. Drag into the world and set the ItemID to set the item.

Data Table

DT_Trees	The data table that allows you to add custom trees and/or edit existing ones.
DT_WoodcuttingItems	The data table that allows you to add custom items (Axes, wood, etc.) and/or edit existing ones.

Structs

F_TreeData	The struct that contains tree data, used for DT_Trees
F_WoodcuttingItemData	The struct that contains Item Data. Used for DT_WoodcuttingItems

Function Libraries

BPFL_CustomInventoryIntegration	A function library containing a function that allows you to make the woodcutting system understand how to read item data from your custom inventory system.
--	---



Support & Contact

For any questions, issues, or feedback, please contact us at:

support@unreal-toolkit.com

Explore all of our assets on: Unreal-Toolkit.com