
SINKINSTUDIO · UNREAL ENGINE 5.5 - 5.8

Advance Character Creator

ACC · Real-time Character Customization for UE5

A fully data-asset-driven character customization system for Unreal Engine 5. Drop it into any project and give players real-time control over gender, body morphs, outfits, hair, accessories, makeup colors and animation previews, all without writing a single line of Blueprint or C++.

Developer	SinkinStudio
Engine	Unreal Engine 5.5 - 5.8
Type	Code Plugin (Blueprint + C++)
Version	v1.0.0
Platforms	Windows, Mac, Linux, Android, iOS

Official Documentation · v1.0.0

Contents

1 About ACC

2 Requirements

3 How to Implement

3.1 Enable the ACC plugin

3.2 Create your character class

3.3 Assign the data assets

3.4 Add rotating component & leader pose

3.5 Place in the level with a Camera Manager

4 Load a Saved Character

4.1 Match components, mesh & interface

4.2 Load on Event BeginPlay

5 The Data Assets Explained

6 Project Settings

7 Create Your Own Data Assets

8 Inside the Character Creator

9 Support

1 About ACC

Advance Character Creator (ACC) is a plug-and-play character customization suite for Unreal Engine 5. Everything a player can change (gender, body shape, outfits, hair, accessories, skin/eye/hair/eyebrow color and animation emotes) is described in data assets and driven by a small set of reusable components. You wire it up once, and the whole creator works.

What you get

Gender Component	Male / female base switching, driven by a Gender data asset
Morph Component	Body morph presets (Normal / Medium / Fat and your own)
Appearance Component	Outfits, hair and accessories from an Appearance data asset
Makeup Component	Skin, eye, hair and eyebrow color via material slots
Montage Component	Emotes and animation previews from a Montage data asset
Save Subsystem	One-click save & load of a full character setup
NOTE · No code required Designers assign data assets in the Details panel and the creator runs. None of the ACC components hold a hard reference to your character class, and they work entirely through an interface, so you can drop them onto any pawn.	

2 Requirements

- A **C++ project**. ACC compiles native modules, so a Blueprint-only project will not build it. Convert via Tools -> New C++ Class (add any empty class) if needed.
- Visual Studio 2022 with the "Game development with C++" (Unreal) workload on Windows.
- A skeletal mesh character that can expose body, outfit, hair and accessories meshes (the included sample character shows the exact setup).

IMPORTANT · Restart after enabling

After you enable the plugin the editor must be restarted once so the ACC modules and components register correctly.

3 How to Implement

Five short steps take you from an empty project to a working character creator. If you just want to try it, open the included sample level; it is already set up this way.

3.1 Enable the ACC plugin

1. Copy the AdvanceCharacterCreator folder into your project's Plugins directory (or install it from Fab).
2. Open **Edit -> Plugins**, search for **ACC**, tick it to enable, and **restart the editor**.

3.2 Create your character class

ACC ships a default character pawn. Create a Blueprint from it (or use your own class, see the note below). Right-click in the Content Browser -> Blueprint Class, expand All Classes, and search for AccCharacterPawn.

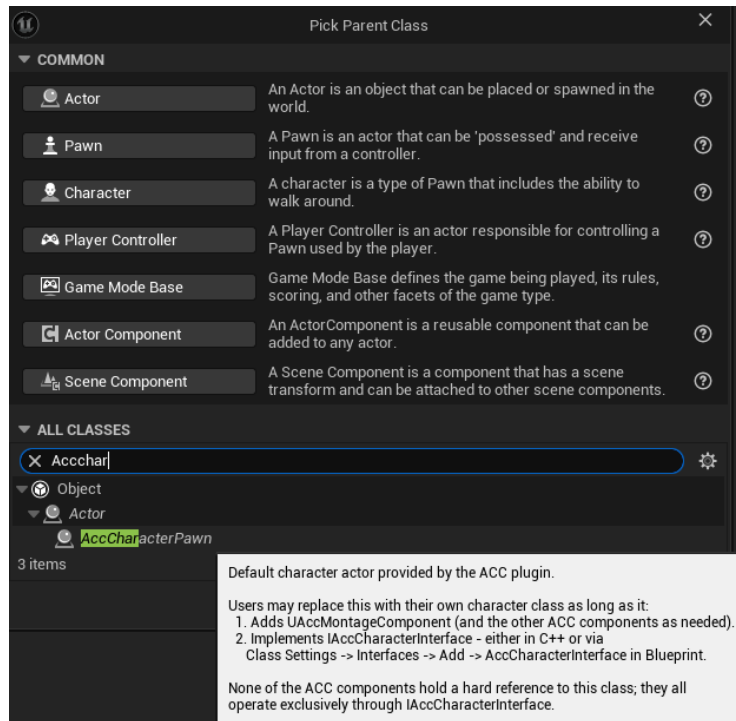


Fig. Pick Parent Class -> AccCharacterPawn. This is the default character actor provided by ACC.

TIP · Use your own character class

You may replace the default with your own class, as long as it:

1. Adds the ACC components you need (Montage, Gender, Morph, Appearance, Makeup).
2. Implements the IAccCharacterInterface, either in C++ or in Blueprint via Class Settings -> Interfaces -> Add -> AccCharacterInterface.

None of the ACC components hold a hard reference to the character class; they all operate through the interface.

3.3 Assign the data assets

Open your character Blueprint. It carries five ACC components: Gender, Morph, Appearance, Makeup and Montage. Each one exposes a single data-asset slot in its Details panel. Assign the matching data asset to each.

 Gender Component (GenderComponent)	Edit in C++
 Morph Component (MorphComponent)	Edit in C++
 Appearance Component (AppearanceComponent)	Edit in C++
 Makeup Component (MakeupComponent)	Edit in C++
 Montage Component (MontageComponent)	Edit in C++
 AccRotatingActor	

Fig. The five ACC components on the character, plus the AccRotatingActor.

Gender component

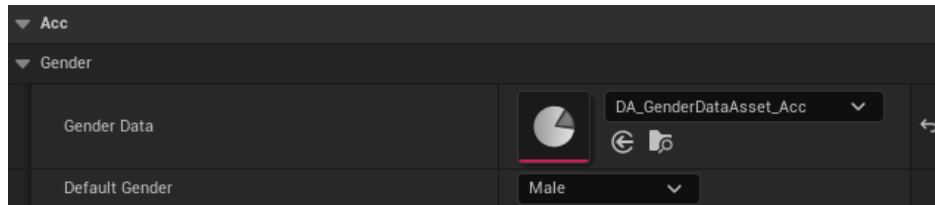


Fig. Assign DA_GenderDataAsset_Acc to the Gender component and pick a Default Gender.

Morph component

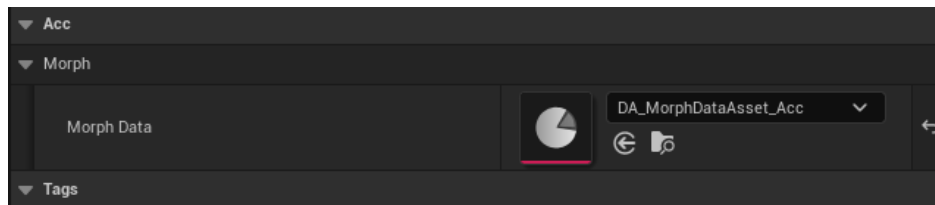


Fig. Assign DA_MorphDataAsset_Acc to the Morph component.

Appearance component

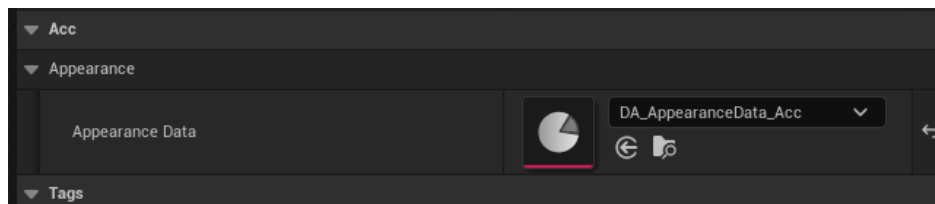


Fig. Assign DA_AppearanceData_Acc to the Appearance component.

Makeup component

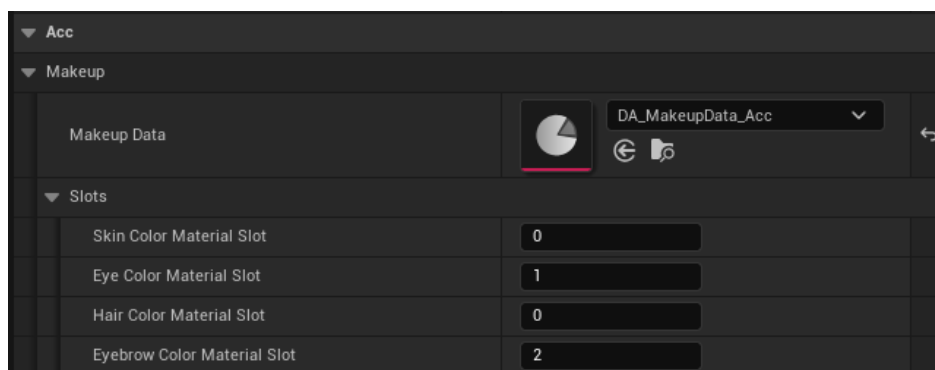


Fig. Assign DA_MakeupData_Acc, then map the Skin / Eye / Hair / Eyebrow color material slots.

Montage component

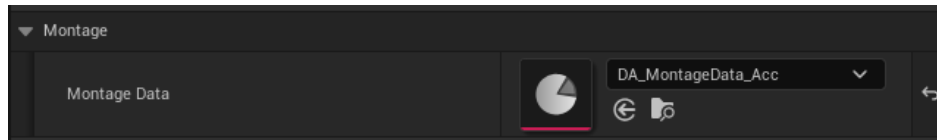


Fig. Assign DA_MontageData_Acc to the Montage component.

NOTE · Material slots matter

The Makeup component recolors the character by driving material slots. Make sure the Skin, Eye, Hair and Eyebrow slot indices match your character material.

3.4 Add the rotating component & leader pose

To let players spin the character on its podium, add the rotating-actor tag. Select the character actor and, under Tags, add an element with the value AccRotatingActor.

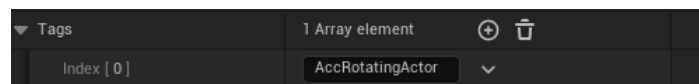


Fig. Add a Tag with the value AccRotatingActor to enable rotation.

So the outfit, hair and accessory meshes follow the body animation, drive them from the body in the Construction Script with Set Leader Pose Component, setting the character (body) mesh as the leader for each additional mesh.

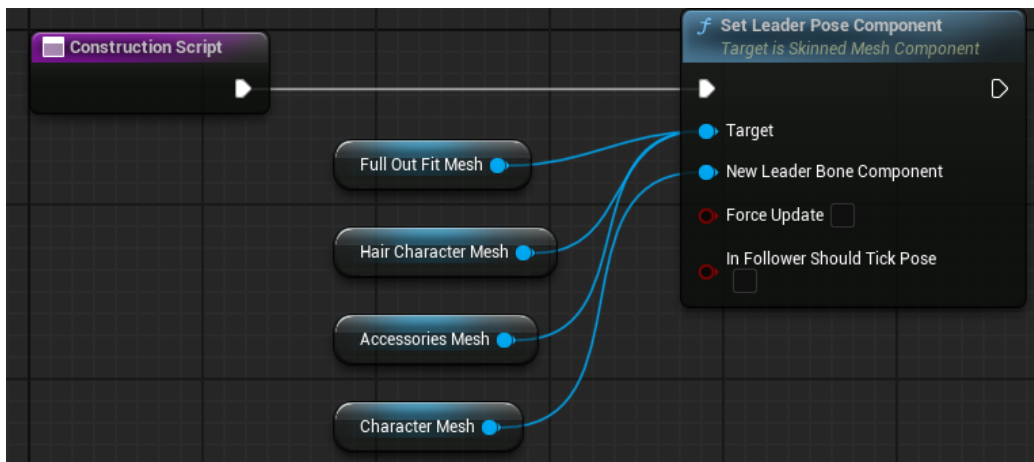


Fig. Construction Script: Set Leader Pose Component links the outfit, hair and accessory meshes to the character mesh.

3.5 Place in the level with a Camera Manager

3. Drag your character Blueprint into the level (the creator podium map, or your own).
4. Add a Camera Manager to frame the character.
5. Add the rotating component setup from step 3.4 if you have not already.

That is it. Once the pawn is in the level with its data assets assigned, your character is used automatically and the creator UI is ready to drive it.

4 Load a Saved Character

To spawn a previously saved character in your gameplay level, the target actor must expose the same components, meshes and interface the creator used, then you simply call the load function on Begin Play.

4.1 Match components, mesh & interface

Build your in-game character with the exact same component and mesh layout: a Mesh (CharacterMesh0) with Hair, Accessories and Outfit child meshes, plus the AccMorph, AccGender, AccMakeup and AccAppearance components.

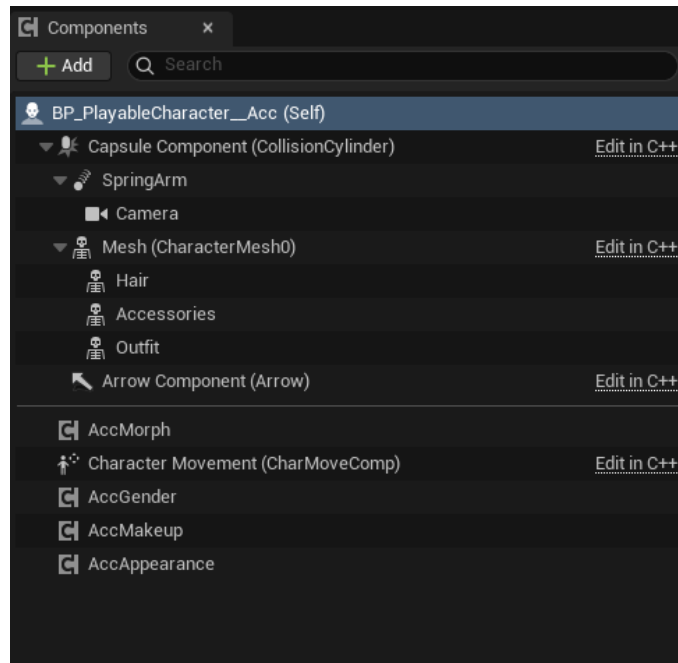


Fig. To load a character you need exactly this component and mesh layout.

Implement the same interface so ACC can read and drive the meshes. Add the Acc Character interface and provide the mesh getters it expects.

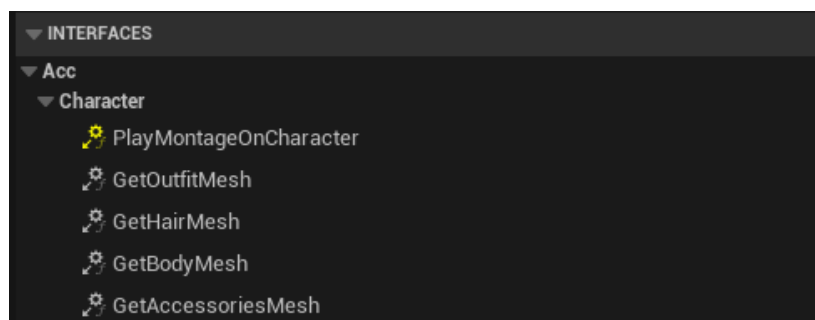


Fig. Implement the same interface: PlayMontageOnCharacter, GetOutfitMesh, GetHairMesh, GetBodyMesh, GetAccessoriesMesh.

IMPORTANT · Pass the right data

Each interface function must return the correct mesh (body, outfit, hair, accessories) for your character. If a getter returns the wrong component, that part of the saved look will not apply.

4.2 Load on Event BeginPlay

Finally, on Event BeginPlay, get the Acc Save Subsystem and call Load Saved Character Onto, passing Self as the target. When you play, the last saved character is rebuilt on your actor.



Fig. Event BeginPlay -> Load Saved Character Onto (Target: Acc Save Subsystem, Character: Self).

5 The Data Assets Explained

Everything the creator offers is described in data assets. Edit them in the Details panel to add, remove or re-skin options, with no code and no recompiling.

Appearance data asset

The Appearance data asset is organized per gender (Male / Female) and per appearance type (Outfit, Hair, Accessories). Each type holds an array of entries, and every entry has a Display Name, a Skeletal Mesh and an Appearance (thumbnail) Texture. Add an array element to add a new outfit or hairstyle.

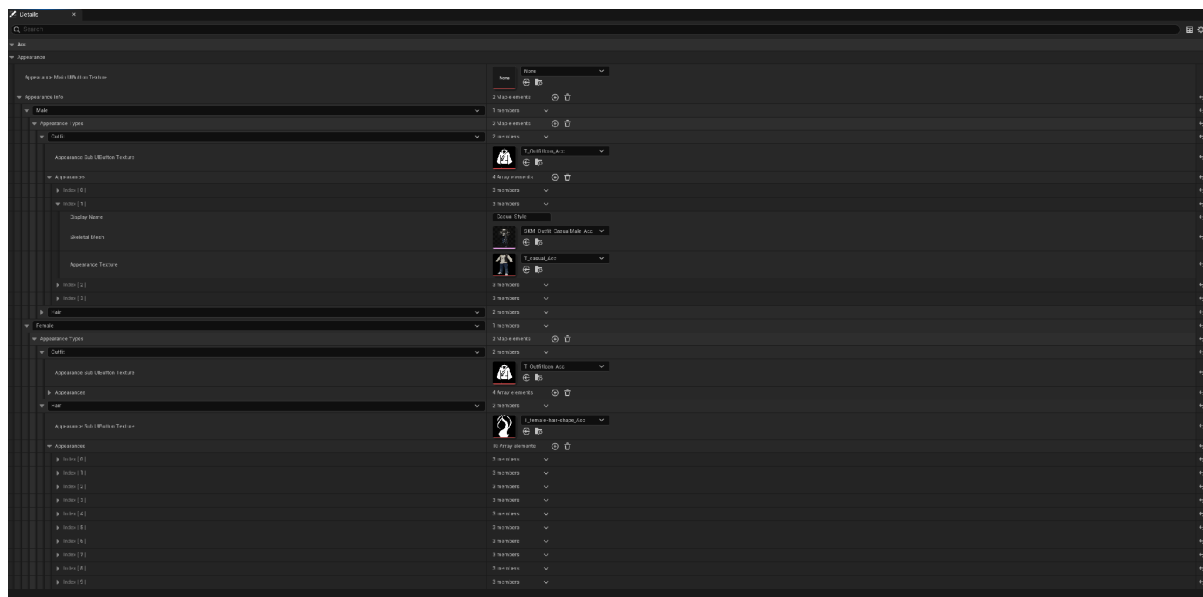


Fig. The Appearance data asset: gender -> appearance type -> array of entries (Display Name, Skeletal Mesh, Texture).

The five data-asset types

DA_AppearanceData	Outfits, hair and accessories per gender, each with mesh + icon.
DA_GenderDataAsset	Male / female base bodies and the default gender.
DA_MakeupData	Color palettes for skin, eyes, hair and eyebrows.
DA_MorphData	Body morph presets (Normal / Medium / Fat and custom).
DA_MontageData	Emote / dance montages shown in the creator.

6 Project Settings

ACC adds its own section under Project Settings -> Plugins -> Advanced Character Creator. From here you can tune the creator without touching code.

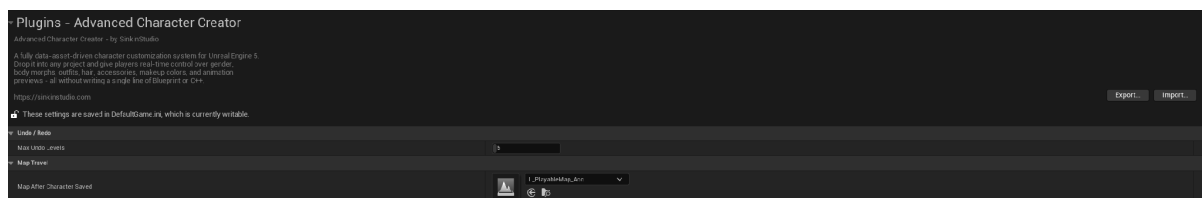


Fig. Project Settings: Undo/Redo depth, post-save map travel, and preset Import/Export.

- **Max Undo Levels:** how many undo/redo steps the creator keeps.
- **Map After Character Saved:** the level to travel to once the player confirms and saves (e.g. L_PlayableMap_Acc).
- **Export / Import:** save and reload your settings as a preset.

NOTE · Where settings live

These settings are stored in DefaultGame.ini. As long as that file is writable, changes persist with your project.

7 Create Your Own Data Assets

To extend the creator with your own content, create a new data asset of the matching ACC type and assign it to the relevant component.

6. Right-click in the Content Browser -> Miscellaneous -> Data Asset.
7. Choose the ACC data-asset class you want (Appearance, Gender, Makeup, Morph or Montage).
8. Fill in the entries, then assign the new asset to the matching component on your character.

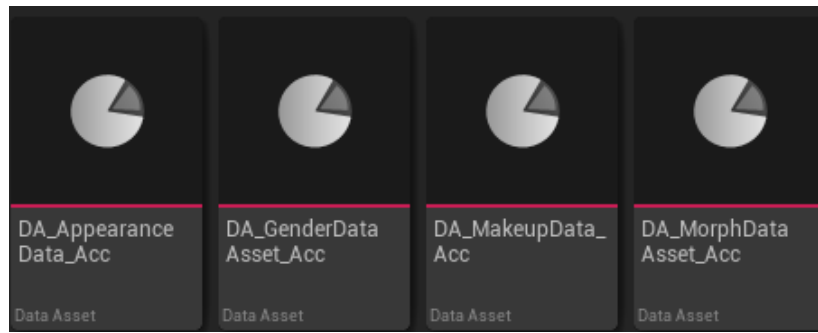


Fig. The four core ACC data assets. Duplicate or create new ones to add your own content.

8 Inside the Character Creator

Here is what your players can do once ACC is running, a quick tour of the creator itself.

Real-time customization

- **Gender:** switch between male and female base bodies instantly.
- **Morph:** choose a body build (Normal, Medium, Fat) or your own morph presets.
- **Appearance:** swap outfits, hairstyles and accessories, each with its own thumbnail.
- **Makeup:** recolor skin, eyes, hair and eyebrows from color palettes.
- **Emotes:** preview dance and action montages right in the creator.

Built-in workflow

- **Undo / Redo:** step back and forward through every change.
- **Reset:** return the character to its starting look.
- **Rotate & Zoom:** inspect the character from any angle on its podium.
- **Confirm & Save:** one click stores the full character and (optionally) travels to your gameplay map.

TIP · Fully data-driven

Every option above comes from a data asset. Add an outfit, a hairstyle, a skin tone or an emote by editing the data asset, and the UI updates automatically.

9 Support

Questions, bug reports or feature requests are welcome. We aim to reply within a couple of business days.

Developer	SinkinStudio
Discord	https://discord.gg/zszefyW7Z
Engine	Unreal Engine 5.5 - 5.8
Version	v1.0.0

Advance Character Creator · SinkinStudio · v1.0.0