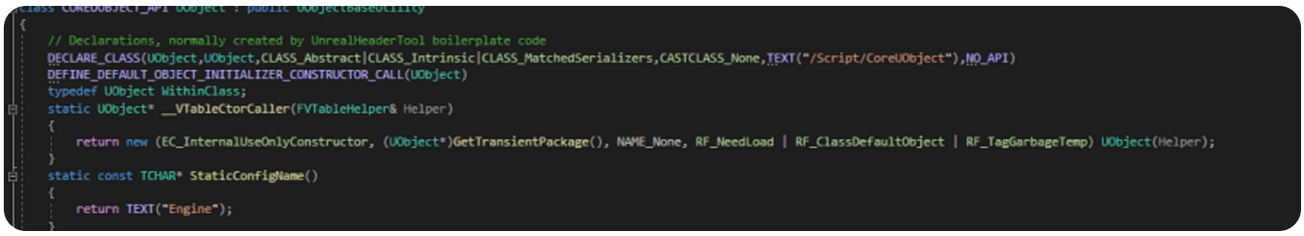
Programming with C++

Information for programmers developing with Unreal Engine.



Unreal Engine provides a robust framework for C++ programmers to help bring their vision to life.

> This section assumes that you have some experience with C++.

This section covers several powerful features that you can use to accelerate your development workflows. You can learn about:

- Creating new Gameplay classes in C++, and all changes will be reflected in the Unreal Editor after compiling with either Visual Studio or XCode. Creating classes in Unreal Engine is similar to creating standard C++ classes, functions, and variables. These are defined using standard C++ syntax.
- Using the Unreal Reflection System to encapsulate your classes with Metadata Property Specifier macros that provide Editor functionality. Each class defines a template for a new Object or Actor.
- Containers in Unreal Engine provide information on Class and Data Structure collections.
- Using the Gameplay Architecture to build your projects in Unreal Engine. The Gameplay Framework provides a hierarchy of Objects and Actors. These classes contain boilerplate variables and functions you can use when creating and designing interactive experiences.
- Creating Delegates to call member functions on C++ objects in a generic, type-safe way. You can dynamically bind a delegate to a member function of an arbitrary object, calling the function on the object at a future time, even if the caller does not know the object`s type.

Section Directory

Unreal Engine Reflection System

Information for programmers developing Objects to be used with Unreal Engine.

Coding Standard

Write maintainable code by adhering to established standards and best practices.

Containers in Unreal Engine

Information on Class and Data Structure collections in Unreal Engine.

Gameplay Architecture

Reference for creating and implementing gameplay classes.

Delegates

Data types that reference and execute member functions on C++ Objects