

Testing, Debugging, and Optimization

An Overview for Network Debugging in Unreal.



When creating a multiplayer game or networked project in **Unreal Engine (UE)**, there are several unique challenges in the process of debugging, profiling, and testing your project. These challenges include:

- Debugging multiple instances of your project
- Accounting for the general unreliability and instability that comes with networked communication
- Examining the different functionality that exists on **Clients** as opposed to **Servers**.



UE multiplayer is based around the Client-Server model. This means that there will be a single server that will be Authoritative over the [GameState](#), while connected clients will maintain a close approximation. See [Client-Server Model](#) for additional documentation.

Unreal Engine features specialized tools and workflows for debugging networked applications. The guides below will show you how to use these tools, as well as tips and best practices for troubleshooting common networking issues.

Index



Logging

An overview of Logging for Network Games.



Console Commands

Specify networking settings and obtain valuable debug information at runtime.



Testing Multiplayer

Set up the Unreal Editor for testing multiplayer games.



Debugging Guide

Testing and Debugging Networked Games in Unreal.



Network Emulation

Emulate network packet lag and loss in Unreal Engine.



Network Profiler

Analyze network traffic and performance information captured at runtime.



Performance and Bandwidth Tips

Some tips for optimizing the performance and bandwidth usage of Actor replication.