

Logging

An overview of Logging for Network Games.



Logging for Networked Games

Client and **Server** logs are important for identifying and debugging networking issues. While many networking logs fall into the **LogNet** category, we recommend checking log categories that are more closely related to specific systems to provide greater insight into any issues.

These categories are not enabled by default and have varying amounts of verbosity that you may need to adjust to gain information about the behavior you're experiencing. The list below provides some recommended categories:

Category	Description
LogNetTraffic	Logs all network traffic when this log variable is set to VeryVerbose.
LogNetPackageMap	Logs information related to how NetGUIDs are Sent, Received, and Acknowledged.

Category	Description
LogNetVersion	Logs information related to Property Replication and RepNotify functions that are used by FRepLayout and FObjectReplicator.
LogNetFastTArray	^
LogNetDormancy	^
LogRep	^
LogRepTraffic	^
LogRepProperties	Logs information related to the sending and receiving of replicated properties.
PacketHandlerLog	Logs information on the packet handler and its components. These components have their own sub-categories. For example, LogDTLSHandler, OodleNetworkHandlerComponentLog, and LogHandshake.
LogDemo	Logs information on recording and playing back replays. Each replay streamer has a related log category: LogLocalFileReplay, LogSaveGameReplay, LogNullReplay, and LogMemoryReplay.

You can enable these log categories and adjust their verbosity by the following methods:

Command-Line Argument

Pass the `LogCmds` command line argument:

```
-LogCmds="<LOG_CATEGORY> <LOG_VERBOSITY>"
```

 Copy full snippet

For example, to set `LogNetTraffic` to `VeryVerbose`:

```
-LogCmds="LogNetTraffic VeryVerbose"
```

 Copy full snippet

Console Command


Use the `Log` console command:

```
Log <LOG_CATEGORY> <LOG_VERBOSITY>
```

 Copy full snippet

For example, to set `LogNetTraffic` to `Verbose`:

```
Log LogNetTraffic Verbose
```

 Copy full snippet

Engine Configuration

Set them in your project's `DefaultEngine.ini`:

```
1 [Core.Log]
2 <LOG_CATEGORY1>=<LOG_VERBOSITY1>
3 <LOG_CATEGORY2>=<LOG_VERBOSITY2>
4 ...
```

 Copy full snippet

For example, to set multiple categories to different verbositys:

```
1 [Core.Log]
2 LogNetPackageMap=Log
3 LogNetTraffic=Verbose
4 LogRep=VeryVerbose
```

Helpful Errors

When reading logs, the following list can be useful for determining what kind of error occurred.

UEngine::BroadcastNetworkFailure	Printed when a net driver encounters some major error. The log line will include the failure type, the error string, and the description of the net driver that encountered the error. You can see a list of the possible network failures with brief descriptions in the ENetworkFailure enum in EngineBaseTypes.h.
UNetConnection::Close	A description of the connection being closed.
UActorChannel::Close	A LogNetTraffic category that will include the channel index, the Actor for that channel, and the reason it was closed. Checking the logs around these lines can help provide some indication as to why a connection or Actor channel was closed.

The command-line argument `-LogTrace=<PARTIAL_LOG_LINE>` performs stack traces from partial log message strings. For example: `-LogTrace=UNetConnection::Close` will produce a stack trace whenever `UNetConnection::Close` is printed to the logs. The command-line argument `-DumpRPCs` provides the capability to dump RPCs and their parameters, which can be useful for tracking what RPCs are being sent and their parameters.



LogTrace and DumpRPCs both require **NetcodeUnitTest** to be enabled.