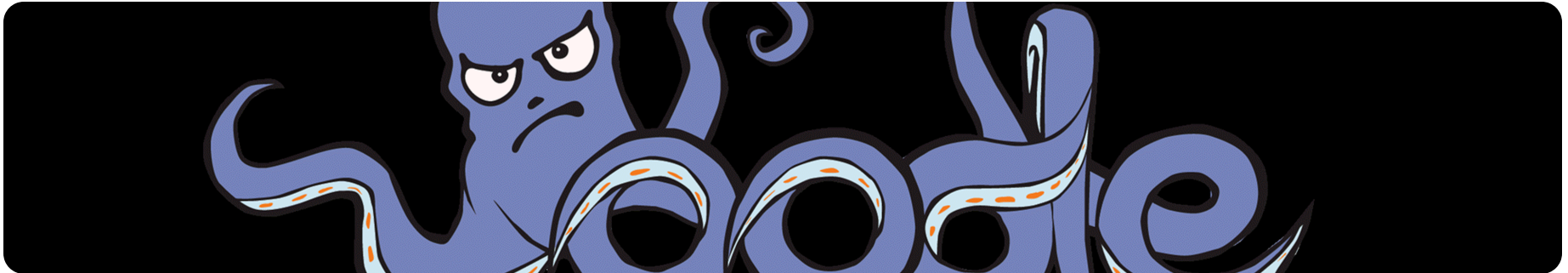


- Developer
 - / Documentation
 - / Unreal Engine ▾
 - / Unreal Engine 5.4 Documentation
 - / Testing and Optimizing Your Content
 - / Using Oodle
 - / Oodle Network

Oodle Network

An overview of using Oodle Network to improve streaming performance for your project.



Oodle Network exposes a compression system for network streams. Integrating this into your project is a bit more hands-on compared with other Oodle solutions, requiring packet captures, familiarity with `BaseEngine.ini`, and offline 'training' of the compressor. It uses the offline training step to dramatically speed up the compression time.

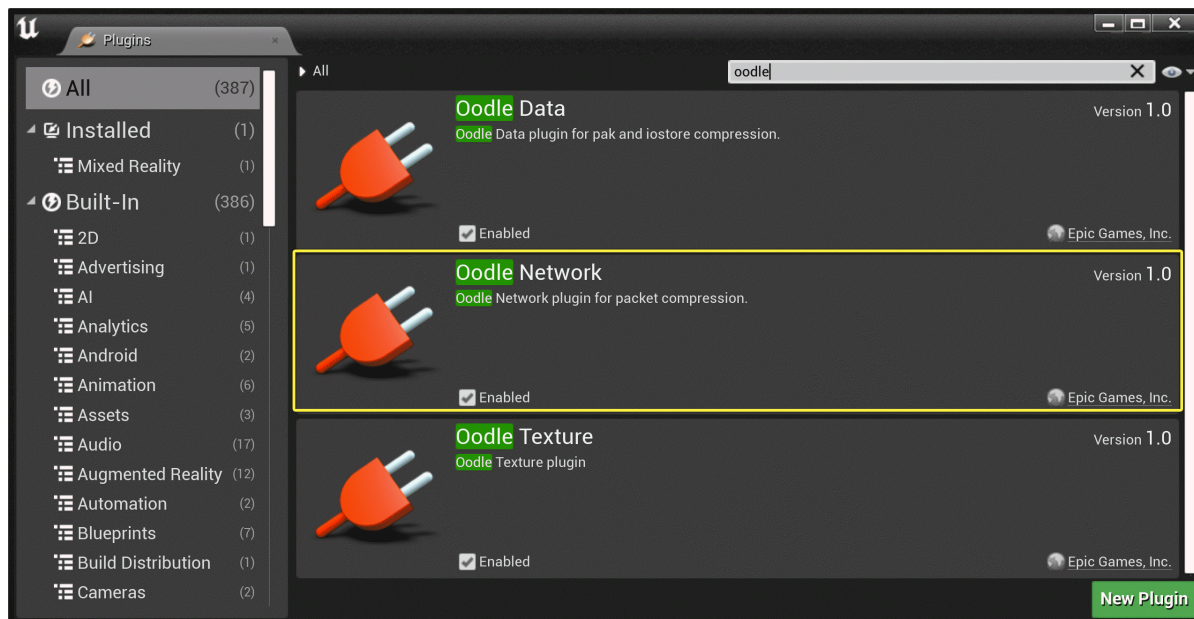
Setting Up Oodle Network

The following is a summary of the steps necessary to use Oodle Network:

1. Verify the Oodle Network plugin is enabled.
2. Capture packet data of your online gameplay, using the most realistic data possible.
3. Train the compressor, creating dictionary files for use at runtime.
4. Configure Oodle Network to use those files.
5. Ensure those files are packaged with your game.

More information about these steps is available in the sections below.

Enabling Oodle Network



The plugin itself should be enabled by default, and you should see that noted in the startup log:

```
LogPluginManager: Mounting plugin OodleNetwork
```

However, you need to tell Unreal Engine to use this plugin when working with network streams. To do this, you need to edit your `BaseEngine.ini` file, in four places:

BaseEngine.ini

```
1
2 [GameNetDriver PacketHandlerProfileConfig]
3 +Components=OodleNetworkHandlerComponent
4 [PendingNetDriver PacketHandlerProfileConfig]
5 +Components=OodleNetworkHandlerComponent
6 [DemoNetDriver PacketHandlerProfileConfig]
7 +Components=OodleNetworkHandlerComponent
8 [PacketHandlerComponents]
9 Components=OodleNetworkHandlerComponent
10
```

 Copy full snippet

Those variables will cause Unreal Engine to use Oodle Network for network streams. The first three entries specify the component for the relevant Net Drivers. If no components are specified for a given Net Driver, then Unreal Engine checks the `PacketHandlerComponents` section.



When debugging, you may need to connect an actual client to do any code-level debugging, as packet handlers are only created on active connections.

Additionally, Oodle Network will look for the following `BaseEngine.ini` file block for configuration:

BaseEngine.ini

```

1
2 [OodleNetworkHandlerComponent]
3 bEnableOodle=true
4 PacketLogFile=PacketDump
5 ServerDictionary=
6 ClientDictionary=
7 ServerEnableMode=AlwaysEnabled
8 ClientEnableMode=AlwaysEnabled
9

```

 Copy full snippet

Setting	Description
bEnableOodle	When set to false, Oodle will not compress any data.
PacketLogFile	This defines the prefix for the filename to write out when capturing packet data .
ServerDictionary	The path to the server-side dictionary file created from the training step. This file needs to be included when packaging your game.
ClientDictionary	The path to the client-side dictionary file created from the training step. This file needs to be included when packaging your game.
ServerEnableMode	<p>Provides a way to conditionally disable compression for certain memory constrained platforms.</p> <p>Options:</p> <ul style="list-style-type: none"> <code>AlwaysEnabled</code>: Self-explanatory; Oodle Network is always enabled when you set this option.

Setting	Description
	<ul style="list-style-type: none"> <code>WhenCompressedPacketReceived</code>: When you set this option, the server-side dictionary is not initialized unless a compressed packet is received.
ClientEnableMode	<p>Provides a way to conditionally disable compression for certain memory constrained platforms.</p> <p>Options:</p> <ul style="list-style-type: none"> <code>AlwaysEnabled</code>: Self-explanatory; Oodle Network is always enabled when you set this option. <code>WhenCompressedPacketReceived</code>: When you set this option, the client-side dictionary is not initialized unless a compressed packet is received.

Capture Packet Data

Packet capturing is done by playing your game with Oodle Network enabled in a special mode. This playthrough should be representative of typical play, as the quality of these packet captures affects the quality of the compression you get.

Whichever executable is acting as the server should be run with `-OodleCapturing`:

```
UE4Editor.exe ShooterGame -game **-OodleCapturing** -windowed -log
```

Clients should be connected without `-OodleCapturing`:

```
UE4Editor.exe ShooterGame -game -windowed -log
```

Packet captures are written to the `Saved/Oodle` directory for the project and configuration. You need to collect many packet captures (50 MB or more) to create a properly representative data set for training. Training takes place using all the packet captures in a directory.

Training

The Oodle Network trainer is a commandlet that reads all of the packet captures and creates dictionaries out of them. These dictionaries are used for compressing the network streams between server and clients.

This is the command to run the trainer commandlet:

```
1
2 UE4Editor.exe ShooterGame -run=OodleNetworkTrainerCommandlet AutoGenerateDictionaries -CompressionTest
3
```

 Copy full snippet

The default dictionary size is 4MB and can be overridden with the `-DictionarySize=(size in bytes)` command line option.

The trainer reads the packet captures from the same directory that Oodle Network writes them to, and writes the resulting dictionaries to the `Content/Oodle` directory:


```
1
2 Content/Oodle/ShooterGameInput.udic
3 Content/Oodle/ShooterGameOutput.udic
4
```

 Copy full snippet

A log will report the compression rates achieved with the generated dictionaries:

```
1
```

```
2 OodleNetworkHandlerComponentLog: Compression test results:
3 OodleNetworkHandlerComponentLog: - CompressionTestPackets: 380, Size: 10428 (~0MB)
4 OodleNetworkHandlerComponentLog: - Uncompressed: 10428 (~0MB), Compressed: 6699 (~0MB)
5 OodleNetworkHandlerComponentLog: - Total Savings: 35.759495
6
```

 Copy full snippet

At this point, training is complete.



If your game changes patch to patch in appreciable ways, you need to retrain Oodle Network on new packet captures, and release new dictionaries with the patch.

Configuring

The server-side and client-side dictionaries should be specified in the `BaseEngine.ini` file:

```
1
2 ServerDictionary=Content/Oodle/ShooterGameOutput.udic
3 ClientDictionary=Content/Oodle/ShooterGameInput.udic
4
```


 Copy full snippet

Once configured, you should be able to use the `stat OodleNetwork` command and view the real-time compression rates in games where stats are enabled.

Packaging

As long as Oodle Network is enabled for your target platform, the dictionary files should be automatically included. You can confirm this by listing the contents of your pak file and searching for the relevant entry:

```
1
2 UnrealPak.exe
   Path\To\ShooterGame\Saved\StagedBuilds\WindowsNoEditor\ShooterGame\Content\Paks\ShooterGame-
   WindowsNoEditor.pak -list
3
4 ...
5 LogPakFile: Display: "ShooterGame/Content/Maps/ShooterEntry_BuiltData.uasset" offset: 326025216,
   size: 735 bytes, sha1: BF611DF661BA7260E086145D328AC09E9E1149FB, compression: None.
6 LogPakFile: Display: "ShooterGame/Content/Maps/ShooterEntry_BuiltData.uexp" offset: 326026004, size:
   286 bytes, sha1: 44DB46D2C5EC43395C2659E97D4B94DCE0064E6A, compression: Oodle.
7 LogPakFile: Display: "ShooterGame/Content/Oodle/ShooterGameInput.udic" offset: 326027264, size: 46416
   bytes, sha1: 95EE01894C1E72CD12105652186515573E42C3FE, compression: Oodle.
8 LogPakFile: Display: "ShooterGame/Content/Oodle/ShooterGameOutput.udic" offset: 326074368, size:
   24339 bytes, sha1: EE0602B1263815475FE75BA5D953D7A7963CAF71, compression: Oodle.
9 LogPakFile: Display: "ShooterGame/Content/ShaderArchive-Global-PCD3D_SM5.ushaderbytecode" offset:
   326100992, size: 5427343 bytes, sha1: 129EED3E7BB14E54ACEB72F4A5D813CD1F8E5FA6, compression: None.
10 ...
11
```

 Copy full snippet

If you get an error retrieving an encryption key, try packaging with `-skipencryption` for this test.