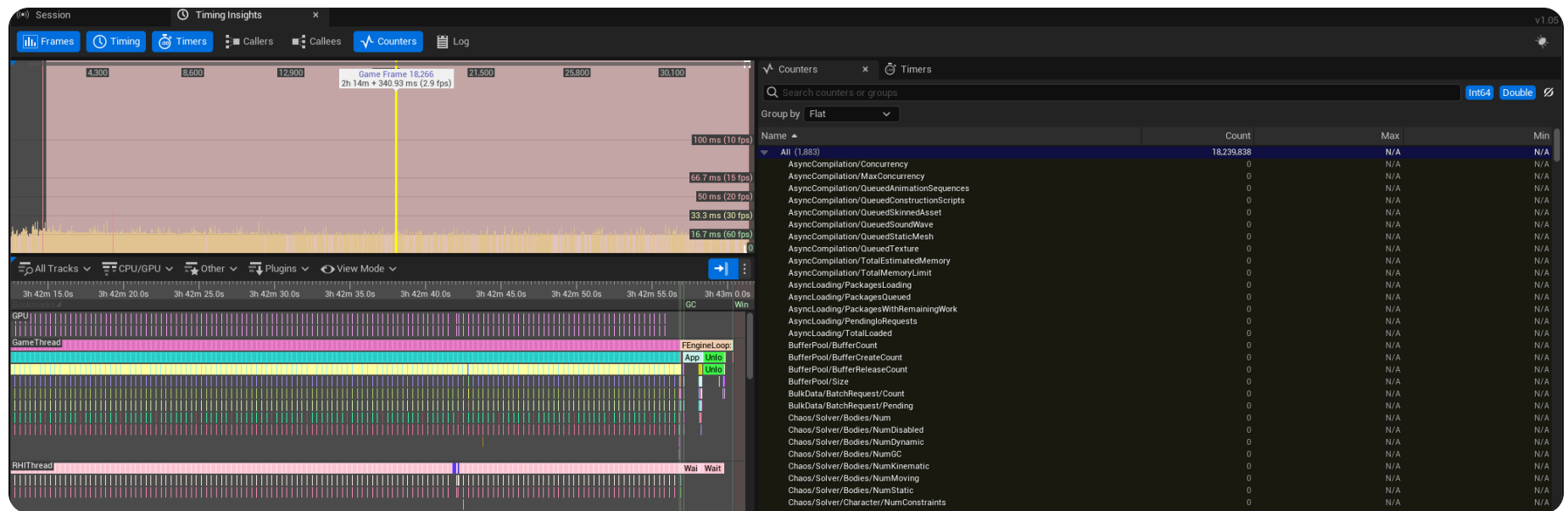


- Developer
- / Documentation
- / Unreal Engine ▾
- / Unreal Engine 5.4 Documentation
- / Testing and Optimizing Your Content
- / Unreal Insights

Unreal Insights

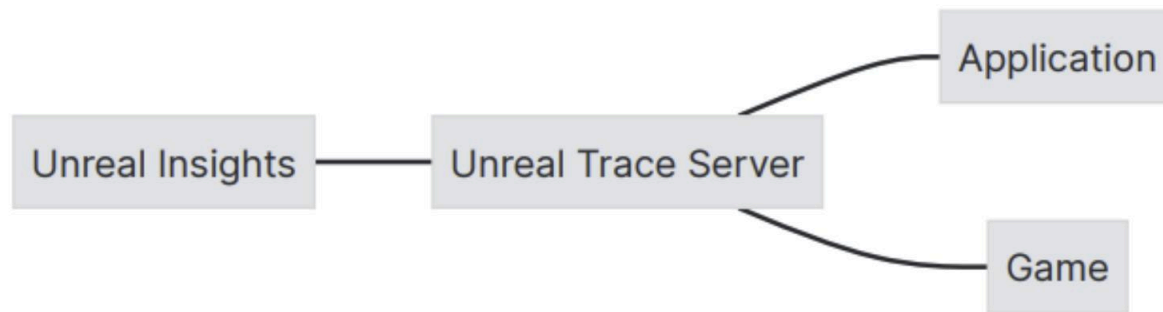
Profile your project's performance with Unreal Insights.



Unreal Insights is a telemetry capture and analysis suite that can capture events from your project at high data rates. Unreal Insights helps you identify areas of data that might require optimization.

The major components of Unreal Insights are:

- **Trace events**, which contain `EventName` and `FieldName` parameters to define an event and specify a field that the event should include.
- The **Unreal Trace Server**, which records and saves traces from the application.
- **Unreal Insights**, which analyzes and visualizes the data.



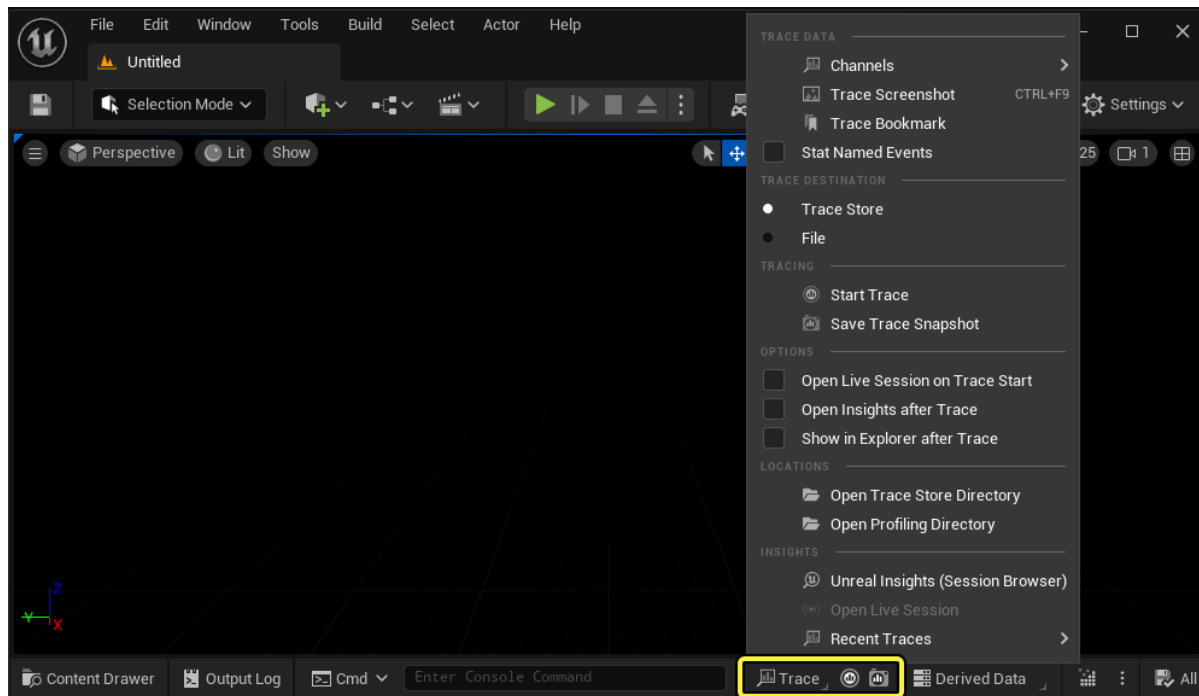
Visualization of the major components of the Unreal Insights framework.

Trace sessions are self-describing, and compatible with different engine release versions. They are stored in `.utrace` files. Any companion data that is generated is stored in `.ucache` files located within the same directory as your trace files.

Setting Up Unreal Insights

Launch From Editor

To start Unreal Insights from the **Unreal Editor**, navigate to the **Trace/Insights Status Bar Widget** located in the bottom toolbar of the Editor.



When you run a Trace to profile your project data, you can choose from multiple workflow options that vary depending on your Unreal Engine Build and Operating System. For more information about these workflow options, see the following pages:

- [Trace](#)
- [Trace Quick Start Guide](#)

Launch a prebuild of Unreal Insights

If you installed a binary version of Unreal Engine you should have a compiled version of Unreal Insights located in the following directory:

```
Engine\Binaries[Platform]\UnrealInsights[.exe]
```

 Copy full snippet


Build From Source

If you don't have a binary version of the engine installed, or you want to compile Unreal Insights from source, you can use the following options:

- **Using an Integrated Development Environment (IDE).** Find the UnrealInsights target located in the Programs folder.
- **From the command prompt.** In your engine installation folder build Unreal Insights using the Unreal Build Tool:


On Windows:

```
Engine/Build/BatchFiles/RunUBT.bat UnrealInsights Win64 Development
```

 Copy full snippet

On Linux or Mac:

```
./Engine/Build/BatchFiles/RunUBT.sh UnrealInsights [Linux|Mac] Development
```

 Copy full snippet

Trace

Trace is a structured logging framework for tracing instrumentation events from a running process. The **Unreal Trace Server** runs in the background as a single server instance and can be shared between multiple projects or branches. It is

an optimized program that has minimal impact on performance and does not include a User Interface.

The **Trace Server** is launched automatically by a separate server process executable, `UnrealTraceServer.exe`, which is located in the `Engine/Binaries/Win64` directory folder.

The **Trace Server** has two components:

- The **Trace Recorder** listens on port 1981 for incoming trace connections and records the live trace stream.
- The **Trace Store** stores the recorded traces as files in a folder. It watches this folder for changes and exposes the list of available traces to Unreal Insights UI.

The **Trace Server** stores configuration and log files in:

- Windows: `%LOCALAPPDATA%/UnrealEngine/Common/UnrealTrace`
- MacOS: `~/UnrealEngine/UnrealTrace`
- Linux: `~/UnrealEngine/UnrealTrace`

The default `Store` folder is created here.

For additional documentation see the following pages:

- [Trace](#)
- [Trace Quick Start Guide](#)
- [Trace Developer Guide](#)

Shut Down Trace Server

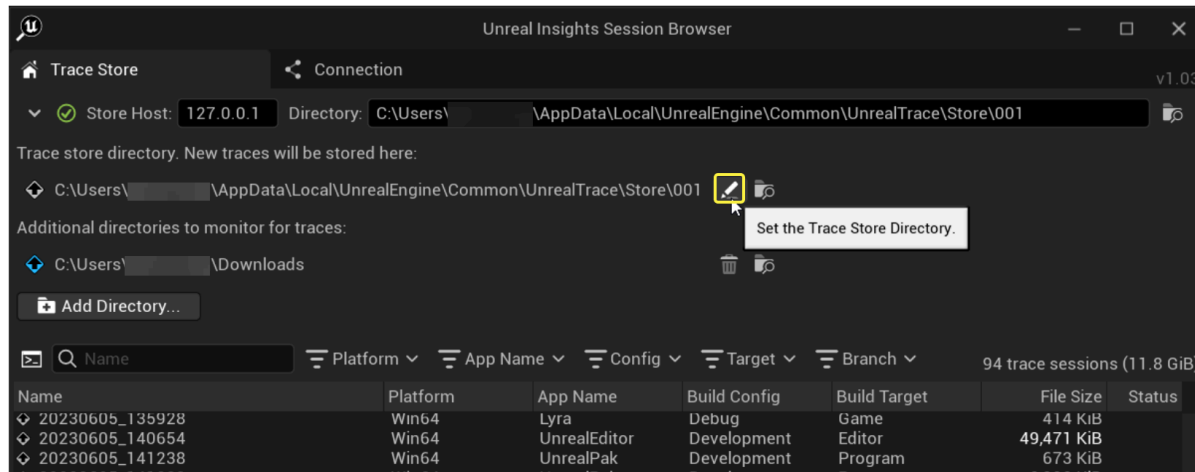
The server can be shut down using the "kill" command:

```
> UnrealTraceServer kill
```

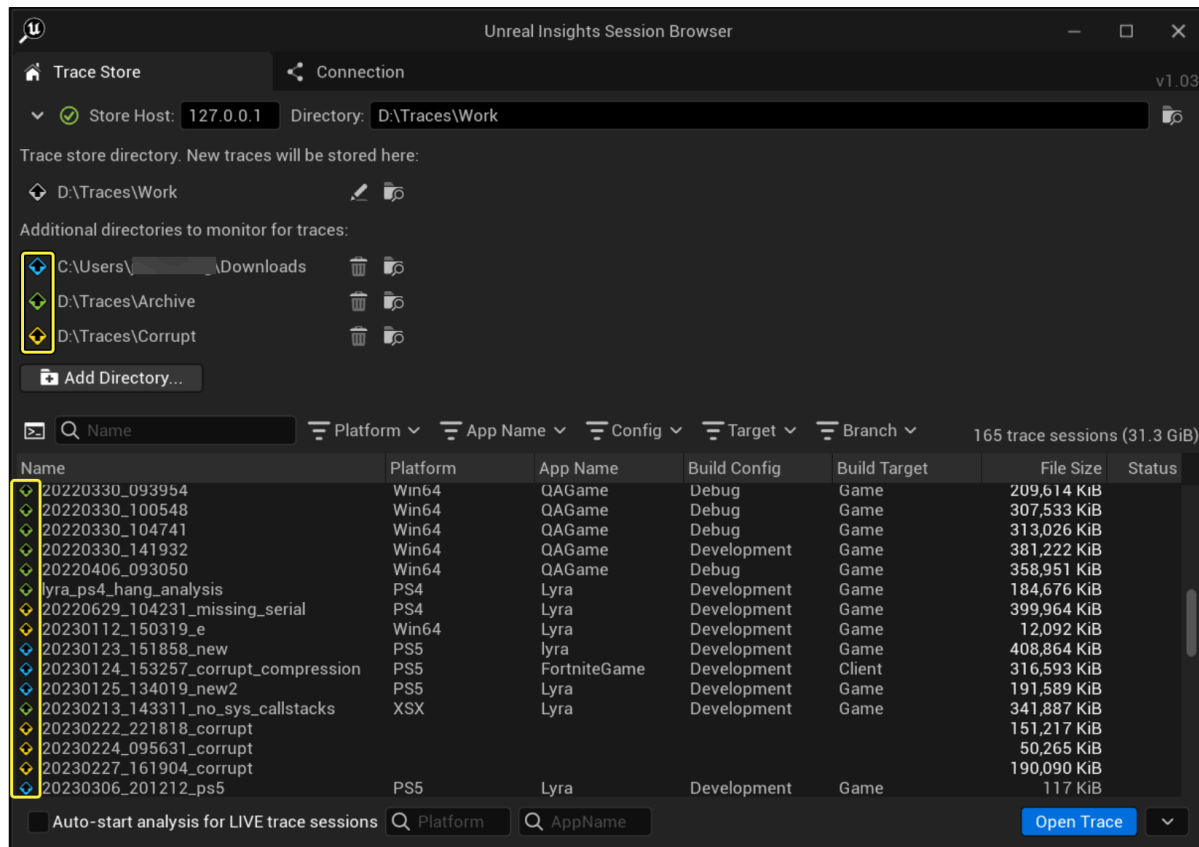
Configuring the Unreal Trace Server

You can configure the Unreal Trace Server to add additional directories to scan for trace files like the download folder or the profiling directory of a specific project. In Unreal Insights, you can control these settings to:

- Set the trace store directory. This is the location where new traces are saved.



- Set additional trace directories and additional sources for trace files, for example, your user Download folder.



When configured with additional watch folders, multiple traces, and their corresponding trace file origin will be displayed with the associated color

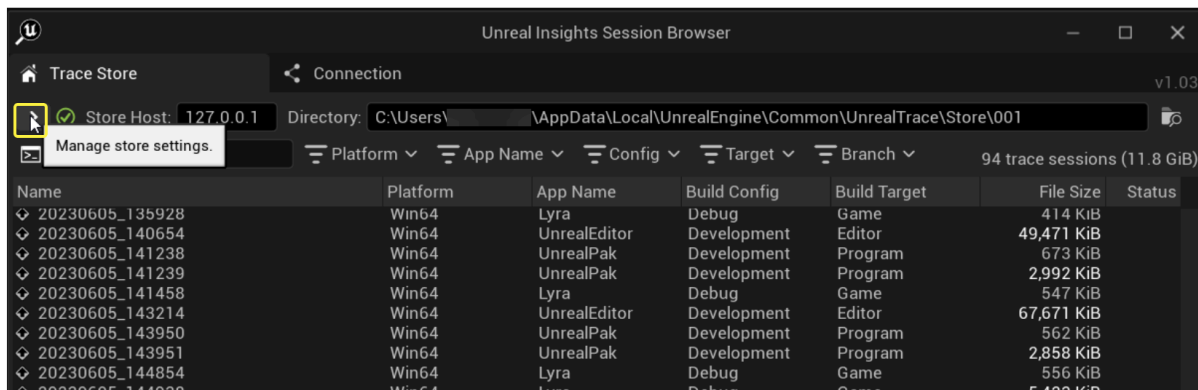
- The Unreal Trace Server can store settings persistently while your computer is shut down or restarted.



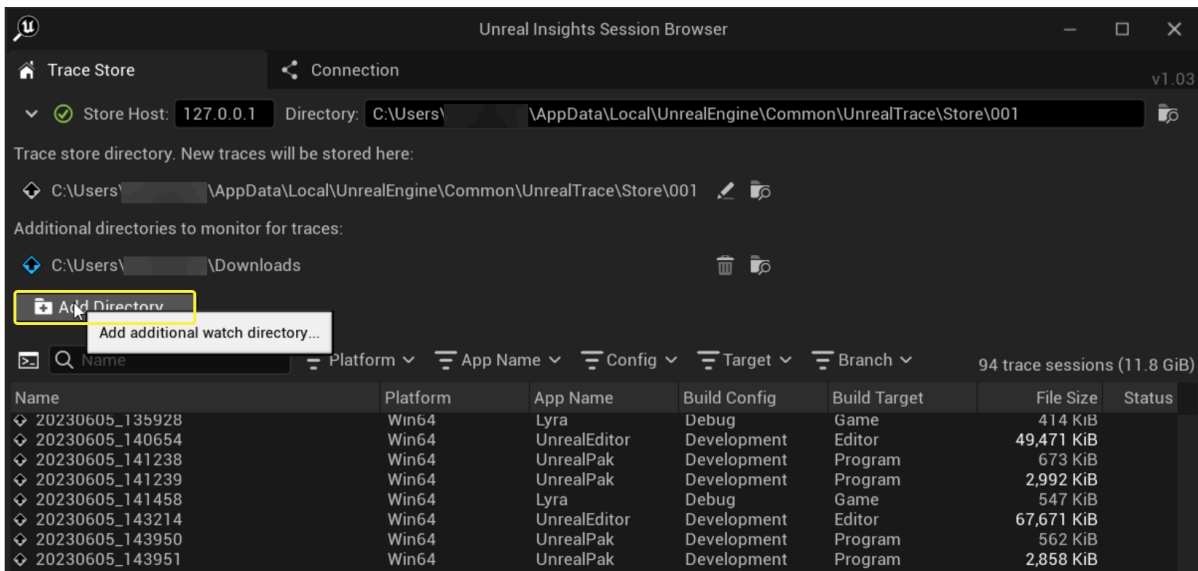
As of UE 5.3, the Unreal Trace Server is enabled for all desktop platforms. This deprecates the store which was hosted in Unreal Insights for Linux and Mac.

Follow the steps below to Configure the Unreal Trace Server.

1. Open Unreal Insights. This will start the Unreal Trace Server, if not already running, on Windows, Mac, or Linux
2. Click on the **Manage store settings** dropdown button then modify the default store directory by clicking the "Set Trace Store directory" button. When starting a new trace the file will be stored in this directory.



3. The old trace store directory is automatically added to watch folders.
4. You can add one or more watch folders by clicking the **Add directory** button. If the new folder contains trace files they will appear in the session list with an icon colored by a unique color.

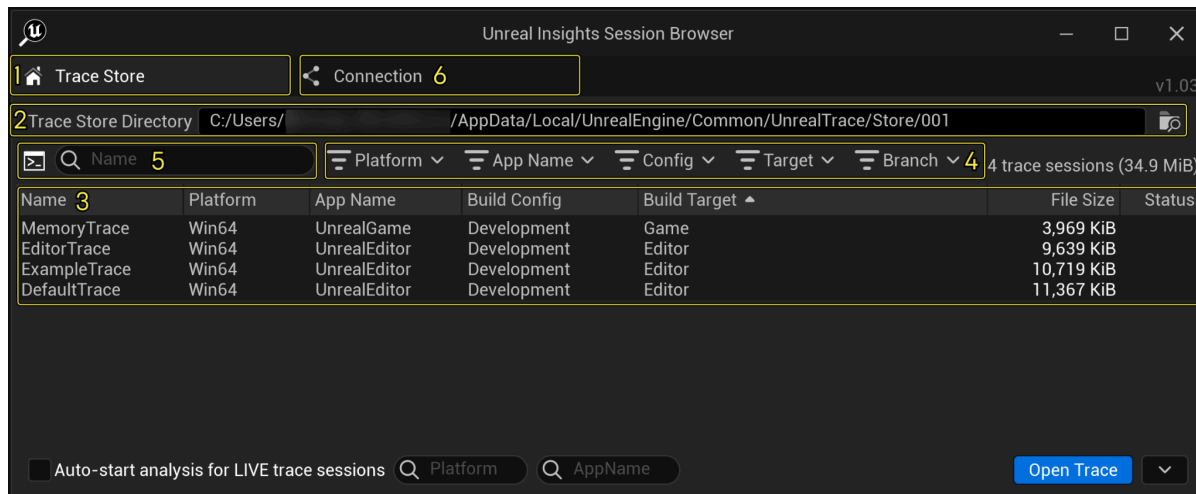


Unreal Insights Session Browser

The Unreal Insights [Session Browser](#) is an interface to observe trace data. To launch the browser, navigate to the bottom toolbar, and click **Trace > Insights > Unreal Insights (Session Browser.)**

Trace Store

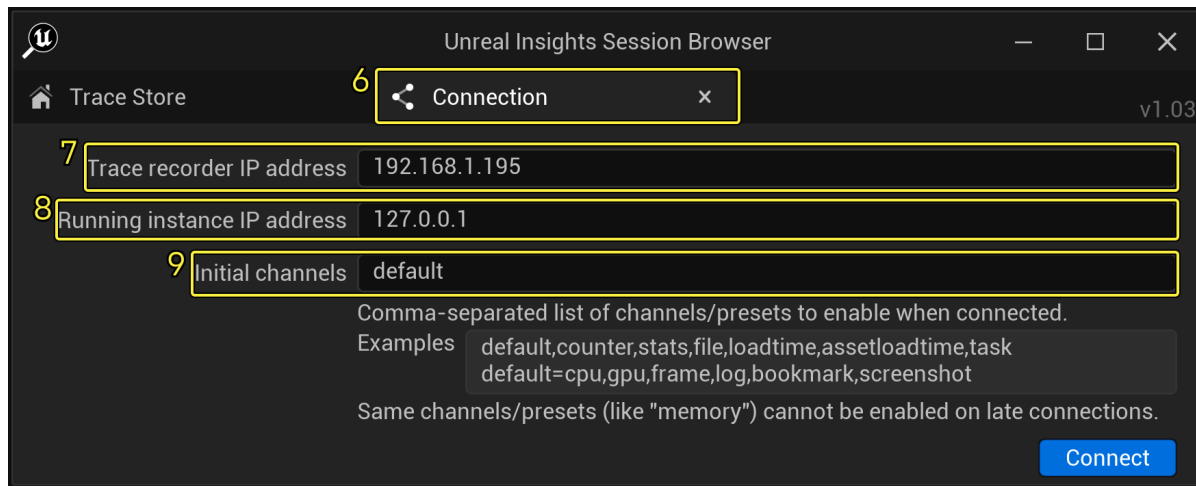
The **Trace Store** is an interface for you to observe and manage all of your stored Trace Sessions. The recorded traces are stored as files in a folder and Unreal Insights watches this folder for any data changes and then exposes the list of available traces to Unreal Insights UI.



For more information, see the [Session Browser](#) page.

Connection Tab

The connection tab provides an interface to connect to a running game or editor with a trace server. It features multiple options to change your connection settings.






For more information, see the [Session Browser](#) page.

Load a Trace for Analysis

There are multiple options to choose to load a Trace for Analysis. You can:

- Double-click any trace session in the Unreal Insights Browser.
- Select a trace session and click **Open Trace**.
- Search for `.utrace` files in other locations by using the **Open Trace** drop-down arrow.
- Start an analysis of a respective trace file immediately, drag and drop a .utrace file from Explorer over the Unreal Insights window.




MISC

-  Open File...
-  Import table...
-  Diff tables...

TOP MOST RECENTLY CREATED TRACES

20230324_075129_00
20230324_075129
20230322_143935
20230322_124140
20230321_122205
20230321_104114
20230321_104049
20230321_103517
20230321_101709
20230321_070433

DEBUG OPTIONS

- ☐  Enable Automation Testing
- ☐  Enable Debug Tools
-  Starship Test Suite

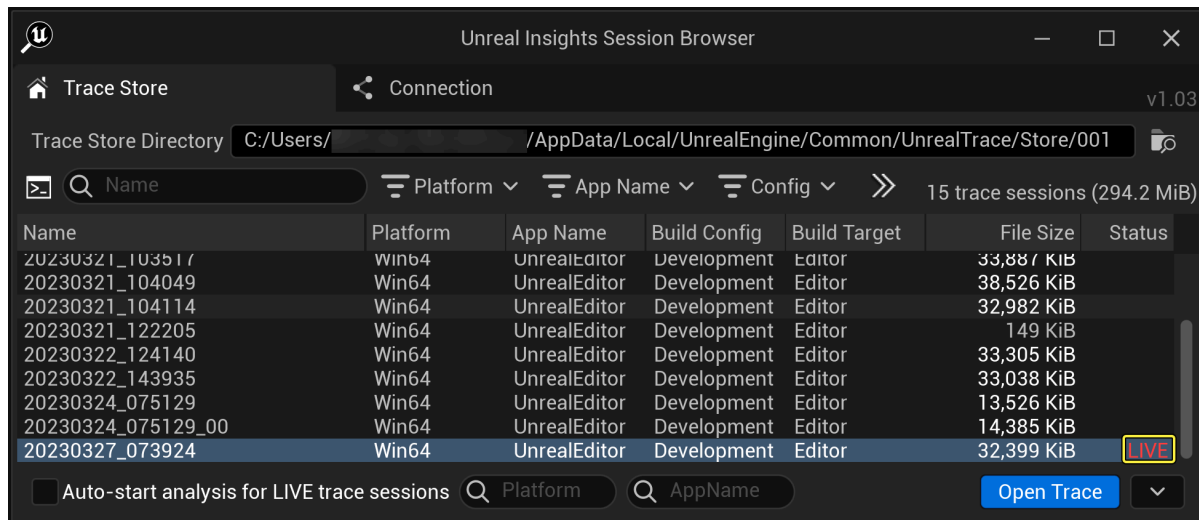
Open Trace



For more information, see the [Session Browser](#) page.

Live Connect

If a live Trace session connects to the tool, it also appears in the list. Live sessions display the word **LIVE** in the status column and update in real-time while you analyze them. Otherwise, they are identical to pre-recorded sessions.



The tool can connect to multiple sessions at the same time, and it automatically records data for all of them as the data streams in. To analyze these sessions in real-time, load them from the list the same way you load pre-recorded sessions.

For more information, see the [Session Browser](#) page.

Insights Menu

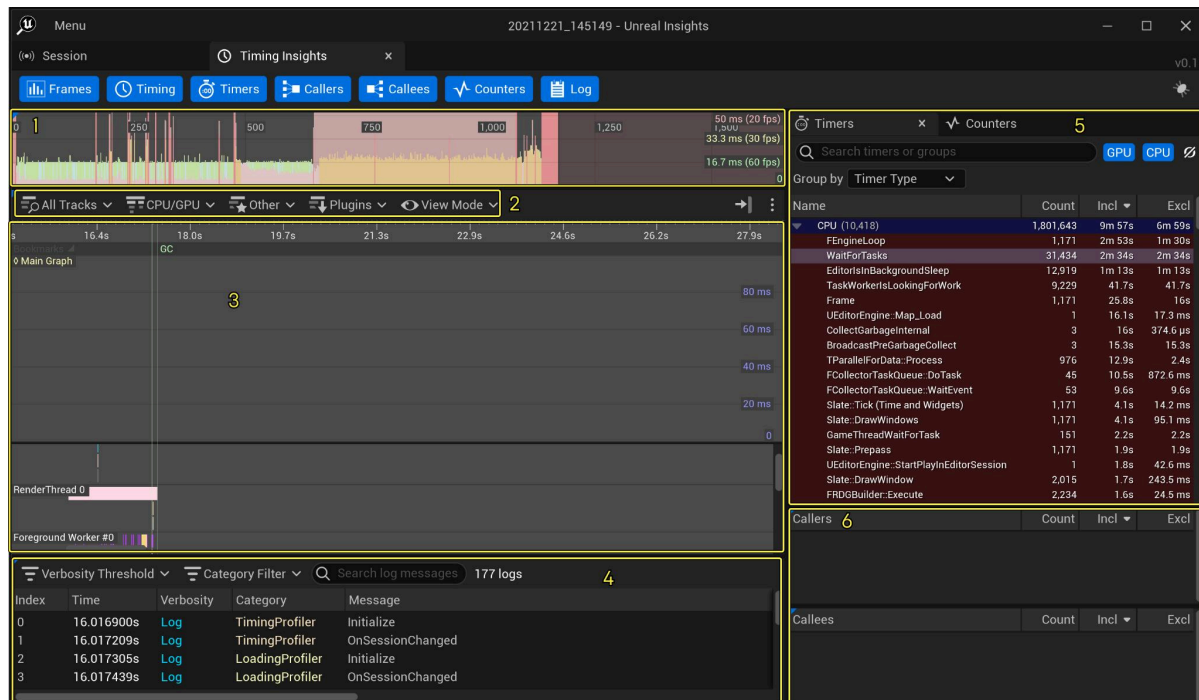
When viewing a session in Unreal Insights, you can select Menu in the upper-left corner of the window to access the menu.

In the menu, you can access several functions, including the following:

- **Import Table** - Import a `.csv` or `.tsv` file into an Insights table.
- **Session Browser** - Opens the Unreal Insights Session Browser window.
- **Open Trace File** - Opens a specified trace file for analysis.
- **Auto Open Live Trace** - If enabled, analysis starts automatically for each new live trace session, replacing the current analysis session.

Timing Insights Window

The **Timing Insights** window collects performance data. It displays the data for **CPU** and **GPU** tracks. These tracks feature multiple sub-menus to help you sort and visualize the various processing tasks and the amount of time your project spends on executing them.

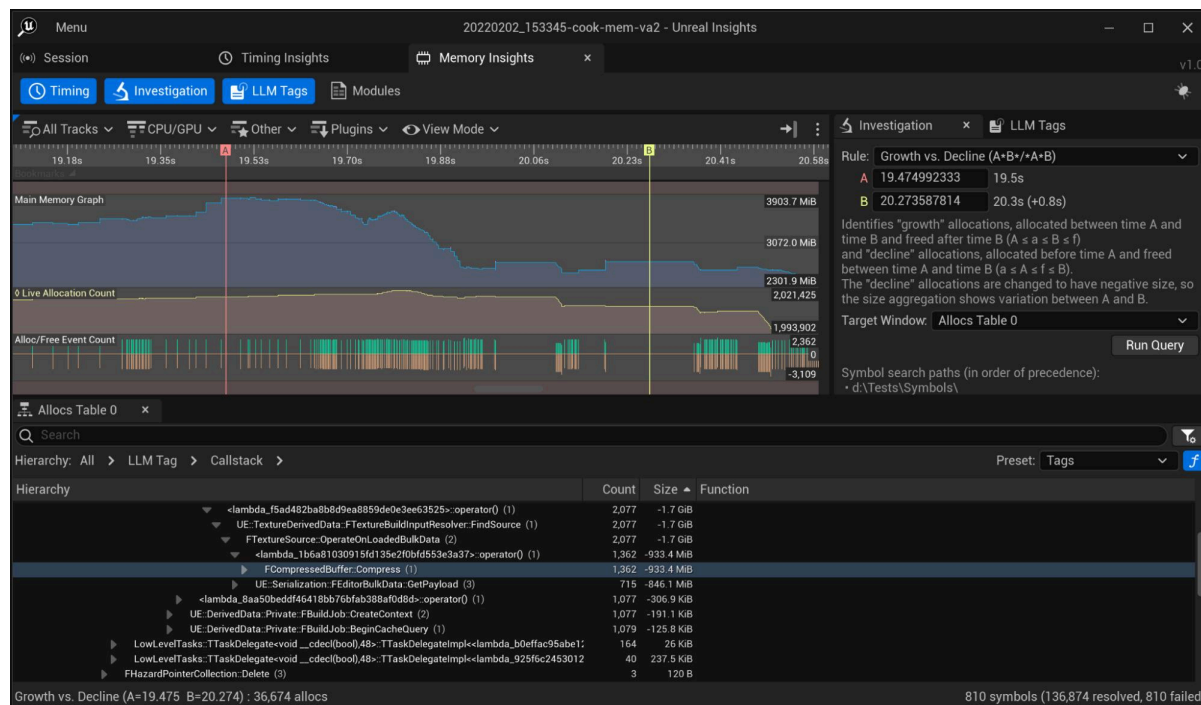


The Timing Insights window features the Frames panel (1), the Timing panel filters (2), the Timing panel (3), the Log panel (4), Timers and Counters tabs (5) and the Callers and Callees panels (6).

See the [Timing Insights](#)

Memory Insights

The **Memory Insights** component allows you to investigate memory usage and call stack tracing in your project.



Memory Insights traces events for every allocation, reallocation, or free event that occurs during runtime, then reconstructs that memory usage pattern during analysis.

Hierarchy	Count	Size	LLM Tag	Function
All (8)	1,413,040	1 GiB		
0x9DBC7034 (2)	1,363,107	327.1 MiB		
_scri_common_main_seh (1)	1,045,808	267.6 MiB		
WinMain (1)	1,045,808	267.6 MiB		
LaunchWindowsStartup (2)	1,045,808	267.6 MiB		
GuardedMainWrapper (1)	1,045,807	267.6 MiB		
GuardedMain (1)	1,045,807	267.6 MiB		
FEngineLoop::Prelim (2)	1,045,807	267.6 MiB		
FEngineLoop::PrelimPostStartupScreen (15)	565,707	113.5 MiB		
FEngineLoop::PrelimPreStartupScreen (63)	480,100	154.1 MiB		
InitEngineTextLocalization (4)	369,481	29.7 MiB		
FTextLocalizationManager::LoadLocalizationResourcesForCulture	369,371	29.7 MiB		
ApplyDefaultCultureSettings (3)	88	7.9 KiB		
FInternationalization::SetCurrentLanguage (2)	79	7.1 KiB		
FICUInternationalization::FindOrMakeCanonizedCulture (4)	73	6.7 KiB		
FICUCultureImplementation::FICUCultureImplementation (3)	62	5.7 KiB		
FCulture::Create (2)	9	308 B		
TSet<TTuple<FString,TSharedRef<FCulture,T>>,TDefaultMap	1	6 B		
FMemory::Malloc (1)	1	712 B		
alloc_4229746	1	712 B	Localization	UnrealEditor-Core.dll\FICUInternationalization::FindOrMakeCanonizedCulture
FICUInternationalization::HandleLanguageChanged (3)	6	426 B		
<lambda_f95c47a1313bbabd6035c4d0d38252c>::operator() (1)	7	594 B		
FTextLocalizationManager::GetNativeCultureName (1)	2	172 B		
FICUInternationalization::RefreshCultureDisplayNames (1)	20	712 B		
FThreadSafeStaticBase::DoSetup (1)	2	81 B		
FModuleManager::LoadModule (1)	57,204	5.9 MiB		
FEngineLoop::Applint (17)	40,913	20.8 MiB		
InitializeShaderTypes (3)	4,938	19.3 MiB		
EKeys::Initialize (12)	1,612	425.9 KiB		
FSlateApplication::InitializeCoreStyle (3)	965	403.1 KiB		

Active Allocs (A=10) : 1,413,040 allocs

685 symbols (132,256 resolved, 679 failed)

See the [Memory Insights](#) documentation for instructions on how to set up, trace, query, and sort your data.

Networking Insights

Unreal Insights includes **Networking Insights** to analyze, optimize and debug network traffic.

Refer to the [Networking Insights](#) for additional documentation.

Slate Insights

Slate Insights extends Unreal Insights to help developers improve the performance of their UI. it provides tools to identify the root cause of a specific Slate and UMG update.

See [Slate Insights](#) for additional documentation.

Asset Loading Insights

Asset Loading Insights provides a way to profile the amount of time it takes to load a project's assets into UnrealEngine. Asset Loading Insights is based on the data traced from the AssetLoadTime trace channel.

This profiling tool is useful in several ways, including the following:

- Viewing details about data sets per asset type.
- Assessing the load order of packages
- Determining if AsyncLoading is respecting package priorities as expected.
- Filtering the Asset Loading tracks.

Cooking Insights

Unreal Cooking Insights allows you to gather and display information about the way packages are cooked in your project. Long cooking times can significantly affect the productivity of teams that are working on larger projects. By displaying the time it takes to cook each package, you can observe which packages to focus your investigation into optimizing. See [Cooking Insights](#) for additional documentation.

Reference

To get the most out of the many features that ship with **Unreal Insights**, You can customize your project's output with macros and command-line options.

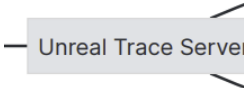
Refer to the [Reference](#) for additional documentation.

Topics



Session Browser

Browse trace sessions to analyze with Unreal Insights.

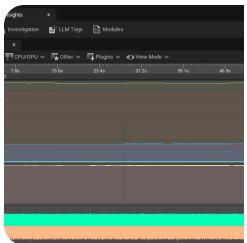


Trace

An Overview of using the Trace logging framework in Unreal Insights.

Timing Insights

An Overview of the Timing Insights Window in Unreal Insights.



Memory Insights

An Overview of Memory Insights



Networking Insights

Overview of Networking Insights, the network performance profiling tool



Slate Insights

Overview of Slate Insights, an extension of Unreal Insights that helps users debug Slate and Unreal Motion Graphics (UMG).



Unreal Insights Reference

Unreal Insights reference covering input shortcuts, macros, and command-line options