

Developer

/ Documentation

/ Unreal Engine ▾

/ Unreal Engine 5.4 Documentation

/ Testing and Optimizing Your Content

/ Unreal Insights

/ Unreal Insights Reference

# Unreal Insights Reference

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



## Trace Channels

Trace is capable of emitting large amounts of data. Channels provide you with control of the data rate when tracing. Each event type is tied to one or more channels, if the required channels are not enabled then the event will not be emitted to the trace stream.


Channel presets groups many channels and provides scenario-based entry points. Users can define their own presets using config files added to the [Trace.ChannelPresets] category.

Available trace channels are included in the table below. You can observe a list of channels that include details on each Channel's **File Path**, **Supported Platforms** and **Notes**:




The plural forms of the name are automatically deduced, "bookmark" and "bookmarks" are treated identically.

Channel	Notes	Supported Platforms	File Path
<b>Log</b>	Logs Messages.		<code>Runtime\Core\Private\Logging\LogTrace.cpp</code>
<b>BookMark</b>	Low frequency markers signifying important transitions, providing a quick overview for users. Such as level loading or engine boot phases.		<code>Runtime\Core\Private\Profiling\Debugging\MiscTrace.cpp</code>
<b>Frame</b>	Game and Rendering frames.		<code>Runtime\Core\Private\Profiling\Debugging\MiscTrace.cpp</code>
<b>Stats</b>	Stats counters. Based on the Stats system.		<code>Runtime\Core\Private\Stats\StatsTrace.cpp</code>
<b>Counter</b>	Generic counters. Allows tracing of values (float, integer) over time.		<code>Runtime\Core\Private\Profiling\Debugging\CountersTrace.cpp</code>

Channel	Notes	Supported Platforms	File Path
	Counters Trace API. It also enables the CSV Profiler Trace.		
<b>CPU</b>	<p>Named cpu timers.</p> <div>  <p>Additional timers can be added using <code>-statnamedevents</code> command line argument.</p> </div>		<code>Runtime\Core\Private\Profiling\Debugging\CpuProfilerTrace.cpp</code>
<b>ContextSwitch</b>	Trace context switch events. On Windows, game/editor runtime should be "run as administrator".	<ul style="list-style-type: none"> <li>• Win64</li> <li>• XBox</li> <li>• PS4/PS5</li> </ul>	<code>Runtime\Core\Private\Profiling\Debugging\PlatformEvents.cpp</code>
<b>Task</b>	Task Graph trace channel.		<code>Runtime\Core\Private\Async\TaskTrace.cpp</code>
<b>GPU</b>	Named gpu timers. Based on GpuProfiler data.		<code>Runtime\RHI\Private\GpuProfilerTrace.cpp</code>
<b>RHICommands</b>	Cpu/Gpu named timers for RHI commands.		<code>Runtime\RHI\Private\RHICommandList.cpp</code>

Channel	Notes	Supported Platforms	File Path
<b>RenderCommands</b>	Cpu/Gpu named timers for commands executed on the rendering thread.		<code>Runtime\RenderCore\Private\RenderingThread.cpp</code>
<b>File</b>	File I/O trace channel (Open, ReOpen, Read, Write, Close events).	<ul style="list-style-type: none"> <li>• Win64</li> <li>• Mac,</li> <li>• XBox</li> <li>• PS4.</li> </ul>	<code>Runtime\Core\Private\Profiling\Debugging\PlatformFileTrace.cpp</code>
<b>LoadTime</b>	Asset Loading Insights trace channel. Package load times (specific trace events) + named cpu timers specific to package loading. Requires "cpu" channel. Also adds the "LoadPackageInternal" cpu timing events (with metadata). See usage of SCOPED_LOADTIMER and SCOPED_CUSTOM_LOADTIMER macros.		<code>Runtime\Core\Public\Profiling\Debugging\LoadTimeTracker.h</code> <code>Runtime\Core\Private\Serialization\LoadTimeTrace.cpp</code>
<b>SaveTime</b>	Named CPU timers specific to package saving. This requires the CPU channel. Adds cpu timers named with <code>UPackage_Save_</code> prefix. See the usage of the		<code>Runtime\CoreUObject\Private\UObject\SavePackage\SavePackageUtilities.h / .cpp</code>

Channel	Notes	Supported Platforms	File Path
	SCOPED_SAVETIMER macro for additional details.		
<b>Cook</b>	Displays named CPU timers specific to cooking. This requires the CPU channel. It will add the "CookByTheBook" and "SaveCookedPackage" cpu timing events (with metadata).		Editor\UnrealEd\Private\Cooker\CookProfiling.h/.cpp
<b>Net</b>	Networking trace channel. Requires -NetTrace=1 (it enables the "Frame" channel)		Runtime\Net\Core\Private\Net\Core\Trace\Reporters\NetTraceReporter.cpp
<b>Module</b>	Module loading information	<ul style="list-style-type: none"> <li>Win64</li> <li>XBox</li> <li>PS4/PS5</li> <li>Switch</li> </ul>	Runtime\Core\Private\Profiling\Debugging\ModuleDiagnostics.cpp
<b>Callstack</b>	<p>Callstack descriptions. Allows allocations to be associated with callstacks.</p> <div>  </div>	<ul style="list-style-type: none"> <li>Win64</li> <li>XBox</li> <li>PS4/PS5</li> <li>Switch</li> </ul>	Runtime\Core\Private\Profiling\Debugging\CallstackTrace.cpp

Channel	Notes	Supported Platforms	File Path
	Requires Module channel to be enabled for symbol resolution to be possible.		
<b>MemAlloc</b>	Memory allocations. Uses Module and Callstack.	<ul style="list-style-type: none"> <li>Win64</li> <li>XBox</li> <li>PS4/PS5</li> <li>Switch</li> </ul>	Runtime\Core\Private\Profiling\Debugging\MemoryAllocationTrace.cpp
<b>MemTag</b>	Memory tag statistics. Traces snapshots of memory usage per tag at regular rate. Relies on LLM subsystem for tracing. Implies "-llm". Available after Init().		Runtime\Core\Private\HAL\LowLevelMemTracker.cpp
<b>VisualLogger</b>	Visual Logger starts recording to file.		Runtime\Engine\Private\VisualLogger\VisualLoggerTraceDevice.cpp
<b>Physics</b>	Chaos Visual Debugger.		Runtime\Experimental\Chaos\Private\ChaosVisualDebugger\ChaosVisualDebuggerTrace.cpp
<b>Animation</b>	Animation Insights(GameplayInsights/Rewind		Runtime\Engine\Private\Animation\AnimTrace.cpp

Channel	Notes	Supported Platforms	File Path
	Debugger) plugin.		
<b>Object</b>	GameplayInsights/RewindDebugger plugin. UObject classes, worlds, instances and events.		Runtime\Engine\Private\ObjectTrace.cpp
<b>ObjectProperties</b>	GameplayInsights/RewindDebugger plugin.		Plugins\Animation\GameplayInsights\Source\GameplayInsights\Private\ObjectPropertyTrace.cpp
<b>NetworkPrediction</b>	Network Prediction Insights plugin.		Plugins\Runtime\NetworkPrediction\Source\NetworkPrediction\Private\NetworkPredictionTrace.cpp
<b>RDG</b>	RDG Insights plugin.		Runtime\RenderCore\Private\RenderGraphTrace.cpp
<b>Slate</b>	Slate Insights plugin.		Runtime\SlateCore\Private\Trace\SlateTrace.cpp
<b>Niagara</b>	Niagara plugin.		Plugins\FX\Niagara\Source\Niagara\Private\NiagaraTrace.cpp

Channel	Notes	Supported Platforms	File Path
<b>Messaging</b>	UDP Messaging plugin.		Plugins\Messaging\UdpMessaging\Source\UdpMessaging\Private\UdpMessagingTracing.cpp
<b>TraceSourceFilters</b>	Trace Source Filtering plugin.		Plugins\Developer\TraceSourceFiltering\Source\SourceFilteringTrace\Private\SourceFilterTrace.cpp
<b>PoseSearch</b>	PoseSearch plugin.		Plugins\Experimental\Animation\PoseSearch\Source\Runtime\Private\Trace\PoseSearchTraceLogger.cpp
<b>Concert</b>	Concert plugin.		Plugins\Developer\Concert\ConcertMain\Source\ConcertTransport\Private\ConcertLogGlobal.cpp
<b>LevelSnapshots</b>	Level Snapshots plugin.		Plugins\VirtualProduction\LevelSnapshots\Source\LevelSnapshots\Private\LevelSnapshotsLog.cpp



Channel	Notes	Supported Platforms	File Path
<b>Screenshot</b>	Captures screenshots triggered with Trace.Screenshot console command or using the <code>TRACE_SCREENSHOT()</code> API.	All platforms.	<code>Runtime\Core\Public\ProfilingDebugging\MiscTrace.h</code>
<b>Asset Metadata</b>	Asset Names and Class Names as metadata for memory allocations. Requires Metadata channel. Used by the MemAlloc channel.	<ul style="list-style-type: none"> <li>• Win64</li> <li>• XB1/XSX</li> <li>• PS4/PS5</li> <li>• Switch</li> <li>• Linux</li> </ul>	<code>Engine\Source\Runtime\Core\Public\ProfilingDebugging\MetadataTrace.h</code>
<b>Metadata</b>	Support for generic metadata scopes.	All platforms.	<code>Engine\Source\Runtime\Core\Public\ProfilingDebugging\MetadataTrace.h</code>
<b>Regions</b>	Long-duration timing events.	All platforms.	<code>Runtime\Core\Public\ProfilingDebugging\MiscTrace.h</code>
<b>AssetLoadTime</b>	Named CPU timers for <code>UObject::Serialize</code> . Also toggles on Blueprint names. Requires CPU channel and <code>statnamedevents</code> .	All platforms.	<code>Runtime\Core\Private\Serialization\LoadTimeTrace.cpp</code> <code>Runtime\Core\UObject\Private\UObject\LinkerLoad.cpp</code>

Some plugins or runtime code enables or disables channels automatically, they are listed in the table below:


Plugin	Channel Description
Gameplay Insights / Rewind Debugger	Toggles the <b>Object</b> channel when Play in Editor (PIE) is started, and toggles <b>ObjectProperties</b> , <b>Animation</b> , and <b>Frame</b> when a recording is started.
TraceSourceFiltering	When the plugin is enabled, the <b>TraceSourceFilters</b> channel is toggled on.
NetworkPrediction	When the plugin is enabled, the <b>NetworkPrediction</b> channel is toggled on.
PoseSearch	When the plugin is enabled, the <b>Pose Search</b> channel is toggled on.
Visual Logger	When recording the <b>Visual Logger</b> channel is toggled on.
-NetTrace=1	When enabled from the command line, the <b>Net</b> and <b>Frame</b> channels are toggled on.

# Controlling Runtime

## Macros

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. Additionally, mouse and keyboard input shortcuts provide users with the capability to navigate Insights windows more efficiently. You can reference the table below to determine which macros you may want to adjust based on the data that you want to see in Unreal Insights:


Macro name	Default State	Source File	Area Description
<code>UE_TRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/TraceLog/Public/Trace/Config.h</code>	Master control for the entire system.
<code>UE_TASK_TRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/Async/TaskTrace.h</code>	Controls tracing of Task Graph events. <code>-trace=default,task</code>
<code>LOGTRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/Logging/LogTrace.h</code>	Controls whether log messages are reported to Unreal Insights. <code>-trace=log</code>
<code>MISCTRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/ProfilingDebugging/MiscTrace.h</code>	Controls tracing for bookmarks, frames, threads, and thread groups. <code>-trace=bookmark,frame</code>
<code>CPUPROFILERTRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/ProfilingDebugging/CpuProfilerTrace.h</code>	Controls tracing for CPU timers and timing events. <code>-trace=cpu</code>
<code>LOADTIMEPROFILERTRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/CoreUObject/Public/Serialization/LoadTimeTrace.h</code>	Controls tracing for events related to load assets. <code>-trace=loadtime</code>
<code>STATSTRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/Stats/StatsTrace.h</code>	Controls tracing for stats counters. <code>-trace=stats</code>


Macro name	Default State	Source File	Area Description
PLATFORMFILETRACE_ENABLED	On	<div><p>The default enabled behavior only applies for Windows, Mac, and PS4 platforms.</p></div> <div>Engine/Source/Runtime/Core/Public/ProfilingDebugging/PlatformFileTrace.h</div>	Controls tracing for file activity, such as opening, closing, reading, and writing files. -trace=file
PLATFORM_SUPPORTS_PLATFORM_EVENTS	On	Engine/Source/Runtime/Core/Private/ProfilingDebugging/PlatformEvents.h	On for platforms with support for platform events (context switch events). -trace=ContextSwitch
GPUPROFILERTRACE_ENABLED	On	Engine/Source/Runtime/RHI/Public/GpuProfilerTrace.h	Controls GPU timers and timing events. -trace=gpu
UE_NET_TRACE_ENABLED	On	Engine/Source/Runtime/Net/Core/Public/Net/Core/Trace/NetTrace.h Engine/Source/Runtime/Net/Core/Public/Net/Core/Trace/Config.h	Controls tracing of network packet content. -trace=net

Macro name	Default State	Source File	Area Description
<code>UE_MEMORY_TAGS_TRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/ProfilingDebugging/TagTrace.h</code>	Enables tracing of tag scopes. If disabled traced allocation will not have an associated tag. <code>-trace=memtag</code>
<code>UE_CALLSTACK_TRACE_ENABLED</code>	On	<code>Engine/Source/Runtime/Core/Public/ProfilingDebugging/CallstackTrace.h</code>	Enables tracing of callstacks. If disabled traced allocations will not contain callstacks. <code>-trace=callstack</code>

## Command-Line Options

When combined with macros, the following command-line options enable trace data:

Command-Line Option	Description
<code>-trace</code> <code>-trace=&lt;channel1&gt;</code>	<p>Provides you the capability to manage how much trace data to generate by categorizing events into named groups.</p> <div>  <p>For example, <code>-trace=cpu,frame,bookmark</code> enables CPU profiler events, frame markers, and bookmarks.</p> </div> <p>Trace data for enabled channels is cached in the "Always on" buffers.</p>

Command-Line Option	Description
<code>-tracehost=&lt;ip&gt;</code>	<p>Sends a trace to the specified IP address. Will default to a local host address.</p> <p> To learn about <code>-tracehost</code>, refer to the <a href="#">Unreal Insights</a>.</p>
<code>-tracefile, -tracefile =&lt;filename&gt;</code>	Performs a "WriteTo" file trace control command. This will run the trace default command.
<code>-tracefiletrunc</code>	
<code>-tracetailmb= N</code>	N specifies the size of the trace tail buffer in MiB. The default size of the trace tail buffer is 4 MiB.
<code>-notraceserver</code>	If specified, the game or editor will not start the local trace server.
<code>-statnamedevents</code>	<p>logs trace data for stat named events. Can be enabled by <code>GCycleStatsShouldEmitNamedEvents = 1;</code> or from calling "Stat NamedEvents" as a console command.</p>
<code>-verbosenamedevents</code>	<p>logs trace data for verbose named events. Can be enabled by <code>GCycleStatsShouldEmitVerboseNamedEvents = 1;</code> or from calling "Stat VerboseNamedEvents" as a console command. An example of a verbose cycle status: "CPU Stall - [...]", "Other TaskGraph Stalls", "FQueuedThread::Run.WaitForWork".</p>

# Console Commands

Command	Description
<code>Trace.Send &lt;Host&gt; [ChannelSet]</code>	Starts tracing to a trace store. is the IP address or hostname of the trace store. [ChannelSet] is an optional comma-separated list of trace channels/presets to be enabled.
<code>Trace.SnapshotSend &lt;Host&gt; &lt;Port&gt;</code>	Send a snapshot of the current in-memory trace buffer to a trace store.
<code>Trace.File [&lt;File&gt;] [ChannelSet]</code>	<p>Begins tracing events to a file. If a file path is not specified, then the default path directory for the utrace file is: <code>YourProject/Saved/Profiling</code>. If a name is not specified, the name of the utrace file is generated automatically based on the current timestamp. Supports Late Connect.</p> <p>Use Examples:</p> <ul style="list-style-type: none"><li>• <code>trace.file [ChannelSet]</code> paths to <code>YourProject}/Saved/Profiling/</code> the <code>*.utrace. Filename</code> is generated automatically.</li><li>• <code>trace.file MyFile.utrace [ChannelSet]</code> paths to <code>YourProject/Saved/Profiling/MyFile.utrace</code>.</li><li>• <code>trace.file C:/Path/ [ChannelSet]</code> is an absolute path, the filename is generated automatically.</li><li>• <code>trace.file Path/ [ChannelSet]</code> is a relative path to <code>YourProject/Saved/Profiling/</code>, the filename is generated automatically.</li></ul>
<code>Trace.Start[ChannelSet]</code>	Keeps the default enabled channels.


Command	Description
<code>Trace.Stop</code>	Stops tracing profiling events.
<code>Trace.Enable[ChannelSet]</code>	Enables a set of channels.
<code>Trace.Disable[ChannelSet]</code>	Disables a set of channels. If no channel set is specified, all channels will be disabled.
<code>Trace.Pause</code>	Pauses all trace channels currently sending events.
<code>Trace.Resume</code>	Resumes tracing that was previously paused (re-enables the pushed channels).
<code>Trace.Status</code>	Prints the trace status to the console.

# Controlling Unreal Insights

## Command-Line Options

Command-Line Option	Usage	Description
<code>-OpenTraceId</code>	<code>-OpenTraceId=id</code>	Forces Unreal Insights to start analyzing the specified ID in "Viewer mode"



Command-Line Option	Usage	Description
<code>-OpenTraceFile</code>	<code>-OpenTraceFile=file.utrace</code>	Forces Unreal Insights to start analyzing the specified file in "Viewer mode"
<code>-Store</code>	<code>-Store=&lt;ip&gt;:port</code>	<p>Connects Browser with specified trace store</p> <div>  <p><code>-StoreHost=address</code> and <code>-StorePort=port</code> are also available.</p> </div>
<code>-TraceAutoStart</code>	<code>-TraceAutoStart=[0 1]</code>	Start tracing to the local trace server automatically at start if Unreal Insights is running or wait to initiate via console command. Default is on.
<code>-NoTraceThreading</code>	<code>-NoTraceThreading</code>	Disable the trace worker thread.

## Keyboard Input Shortcuts




All input operations and control references are limited and subject to change.


# Timing Insights Window




The following mouse and keyboard commands help you navigate Timing Insights more efficiently:

## Frames Panel

Keyboard Input Shortcut	Function
Left mouse click	Select a frame <div> The timing view centers on the selected frame.</div>
Right mouse click	Open context menu
Left/right mouse drag	Horizontal zoom
Mouse wheel	Horizontal zoom
Shift + mouse wheel	Vertical zoom



## Timing Panel



Input Shortcut	Function
<b>Left/right mouse drag</b>	Horizontal or vertical pan
<b>Ctrl + left/right mouse drag</b>	Horizontal pan
<b>Shift + left/right mouse drag</b>	Vertical pan
<b>Mouse wheel</b>	Zoom
<b>Ctrl + mouse wheel</b>	Horizontal scroll
<b>Shift + mouse wheel</b>	Vertical scroll
<b>Right mouse click</b>	Open context menu <div>  Options available for Graph track. </div>
<b>Left mouse click on Timing Event</b>	Select the timing event
<b>Left mouse click on empty space</b>	Clear selection

Input Shortcut	Function
<b>Ctrl + double left mouse click</b>	Select time range of selected timing event <div>  Pressing ENTER performs the same function.         </div>
<b>Left/right mouse drag over Time Ruler</b>	Select time region <div>  Aggregated states automatically update in Timers and Counters views.         </div>
<b>Ctrl+F</b>	Toggles the Quick Find menu.
<b>F</b>	Frame the last selection and then alternate between framing the time range or the timing event <div>  The last selection can be a time range or a timing event.         </div>
<b>G</b>	Toggle Graph track visibility, showing Game and Rendering frames, including a placeholder for Counter series
<b>Y</b>	Toggle GPU Timing track visibility
<b>U</b>	Toggle CPU Timing track visibility

## Input Shortcut

## Function

<b>I</b>	Toggle I/O Overview and Activity track visibility
<b>O</b>	Toggle visibility of background events for file activities in the I/O Activity track
<b>L</b>	Toggle Asset Loading track visibility
<b>C</b>	Switch between normal and compact mode, affecting how timing events are visualized
<b>V</b>	Auto hide empty timelines <div> Empty timelines are ones without timing events.</div>
<b>X</b>	Chooses the next event depth limit: Single Lane, 4 Lanes, Unlimited.
<b>B</b>	Toggle visibility of bookmarks <div> Bookmarks display by default.</div>
<b>M</b>	Toggle visibility of time markers (all trace logs instead of just bookmarks)

Input Shortcut	Function
	<div>  Time markers do not display by default. </div>
<b>**-/ +</b>	Zoom
<b>Ctrl + left/right/Up/Down arrows</b>	Horizontal and vertical pan
<b>Left/right/Up/Down arrows</b>	Select timing event that is its previous or next sibling/parent/largest child
<b>Enter</b>	Select time range of selected timing event  <div>  Pressing Ctrl + Double Left Mouse Click performs the same function. </div>

## Networking Insights Window

The following mouse and keyboard commands help you navigate Networking Insights more efficiently:

Input Shortcut	Function
<b>Left mouse click on packet</b>	Selects the respective packet
<b>Shift + left mouse click on different packet</b>	Selects a range of packets

Input Shortcut

Function

<b>Ctrl + A</b>	Select all packets
<b>Left/right arrows</b>	Selects previous/next package
<b>Shift + left/right arrows</b>	Extends selection (multiple packages) toward Left/right side
<b>Ctrl + shift + left/right arrows</b>	Shrink selection (multiple packages) from Left/right side
<b>Double click an Event</b>	Highlights all packets containing the event in the Packet Content view while masking out all other events