dotProfile Documentation

Release 0

Terry Magnus Drever

Aug 24, 2018

General

Welcome to the official documentation for .Profile; the session-based, deterministic performance data reporting tool for your applications built on unity3d. We recommend you read the *introduction page* to get an overview of what this documentation has to offer.

The table of contents below and in the sidebar should let you easily access the documentation for your topic of interest. You can also use the search function in the top left corner.

Note: Notice something wrong with our documentation? Feel free to submit a pull request.

If you have a technical question, please feel free to contact us through our keybase team wellfiredltd.technicalsupport

---

The main documentation for the site is organized into the following sections:

CHAPTER 1

About

1.1 Introduction

This page aims at giving a broad presentation of the tool and of the contents of this documentation, so that you know where to start if you are a beginner or where to look if you need info on a specific feature.

1.1.1 About the documentation

This documentation is continuously written, corrected, edited and revamped by members of the .Profile team and community. It is edited via text files in the reStructuredText markup language and then compiled into a static website/offline document using the open source Sphinx and ReadTheDocs tools.

Note: You can contribute to .Profiles's documentation by opening issues through YouTrack or sending patches via pull requests on its GitHub source repository.

1.1.2 Organisation of the documentation

This documentation is organised in five sections, the way it is split up should be relatively intuitive:

- The *General* section contains this introduction as well as general information on the tool It also contains the *Frequently asked questions*.
- The *Getting Started* section is the the main entry point of this documentation, as it contains all the necessary information on using the tool. It starts with the *Step by step* tutorial which should be the entry point for all new users.
- Finally, the *Class API reference* is the documentation of the .Profile API. It is generated automatically from a files in the main repository, and the generated files of the documentation are therefore not meant to be modified.

1.2 Frequently asked questions

1.2.1 Someone asks a question?

We should answer it here

CHAPTER 2

Step By Step

2.1 Installing

2.1.1 Package Contents

Each .unitypackage downloaded from the AssetStore or from the WellFired website will have the same contents.

- **/WellFired** Here you're going to find all things related to the .Profile project.
- **/link.xml** This file contains a list of assemblies that might be needed if your Unity Project enables stripping. This will ensure Unity doesn't remove needed assemblies.
- **/Bootstrap.cs** This is a simple script that you can attach to a game object and have instant profile support. Feel free to investigate this script.

2.1.2 Dependencies

.Profile has two dependencies, both come included with the installation, however they might conflict with your already existing project, especially if it's a large project, the following assemblies are included with .Profile.

- **NewtonSoft.Json** This isn't the typical NewtonSoft.Json package, it's a custom build package that doesn't use JIT compilation, making it the preferred choice if you plan to target none desktop platforms. It's prepared and developed by WellFired, but it runs against all of Newtonsoft.Jsons unit tests. Prefer this over your installation if you don't want NewtonSoft.Json to use to use JIT compilation.
- **WellFired.Promise** A lightweight promise library.

Tip: You can safely remove either of these assemblies if they conflict with your project, removing them won't harm .Profile, .Profile will simply default to using the versions contained in your project.

2.1.3 Installing

1. Import the .unitypackage into your unity project.
2. Add the scene WellFired/WellFired.Profile/VisualAssets/Scenes/DotProfileUI to your build settings if you'd like to recieve our automatic performance visualisation.

2.1.4 To be continued. . . ..

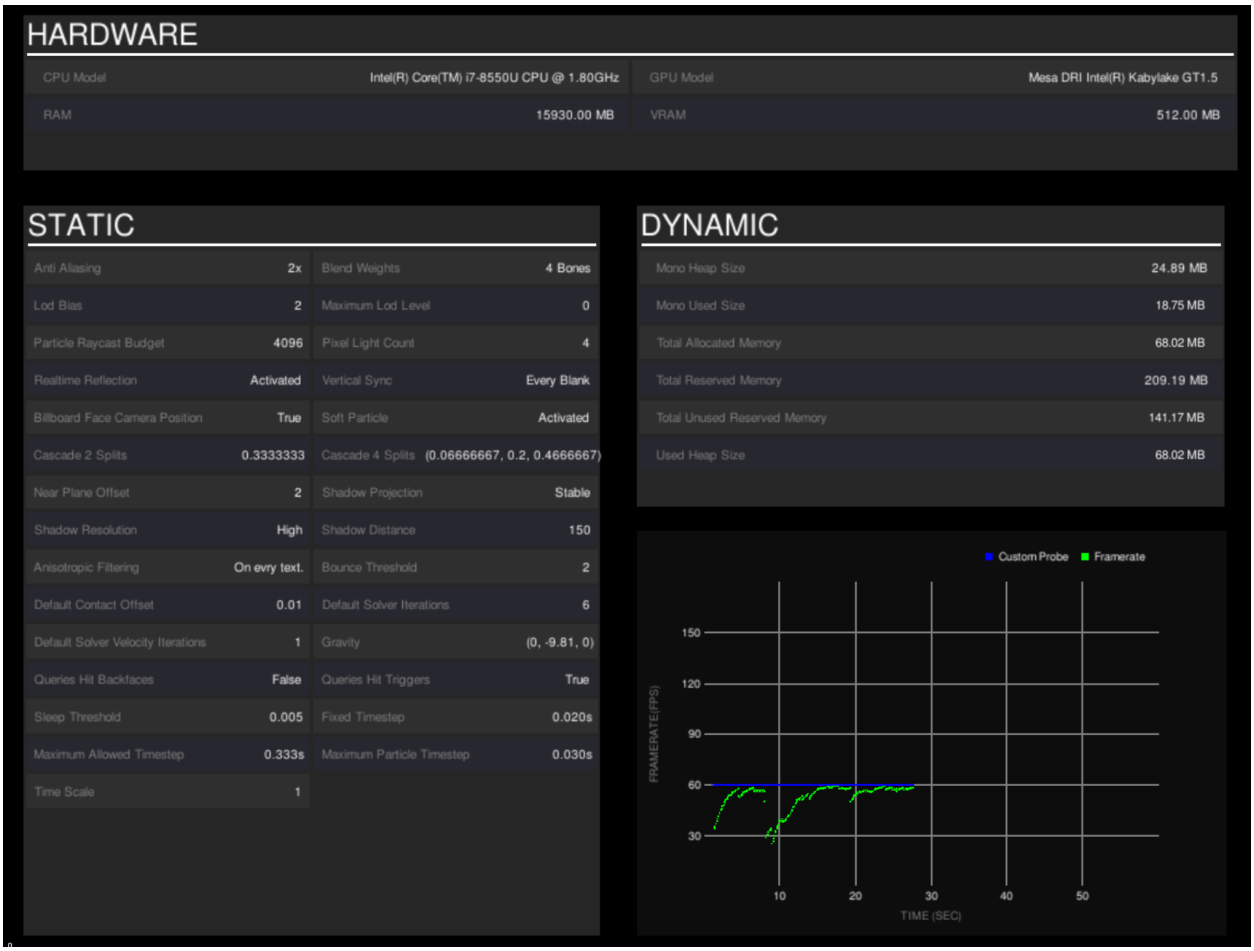
This section walked you through installing .Profile, in the next section you'll get to use .Profile with our pre made script.

2.2 Quick Start

.Profile is built to be incredibly easy to use, in here, you're going to find two sections, one will get you running with no work, and the other is a bit more manual, but provides you more control.

2.2.1 Quickest Start

Simply add the script Bootstrap.cs to a game object in your scene and press play! This will automatically create a fully functioning Profile, that tracks a lot of data, you can toggle the profile visual display by pressing the A key, you should be presented with an overview of performance metrics . . .



If that's all you need, you're now done, you can ship your product to customers and they'll be able to visualise performance data, your internal QA team can use it to track performance across a number of devices, or you can use it yourself to get high on your supremely optimised code.

2.2.2 Slightly less Quick Start

For those of you who want to know more about what's going on under the hood, you can follow this quick start. If you're not interested in going into more detail about managing profiling sessions, you can skip this section.

We'll take you through the basics of setting up a simple profiling session from scratch. During this quick start, we'll assume that you already have some code that you'd like to use to setup a profiling session, this could be anything from a simple cs script with an OnStart Method to a complex FSM.

1. Decide where you'd like to create and maintain your profile session
2. Add the required using

```
using WellFired.Profile.Probes;
using WellFired.Profile.ProfileProcessor;
using WellFired.Profile.Unity.Runtime;
```

3. Create a new Profiling Session

```
// Create a new session.
var session = DotProfileSession.New();
```

4. Tell your session what data you'd like it to track using one of the provided defaults

```
// Track Continuous framerate
session.Track(Defaults.Continuous);
```

5. Tell your session how to process the recorded data

```
// Process recorded data using the built in visual display extension
session.ProcessData(new VisualProcessor(() => Input.GetKeyDown(KeyCode.
↪A)));
```

6. Tell the session to start recording

```
session.StartRecording();
```

7. Optionally tell the session to stop recording

```
session.StopRecording();
```

8. Execute your code and you will have a complete profiling session with instant display.

2.2.3 And Then?

Although we glossed over most core concepts of .Profile, you've learned the basic interface for a session, how to maintain one, track and display certain aspects. In the following sections, we'll go into more detail about how exactly Tracking and Processing works, as well as allowing you to completely customise .Profile.

If you want more information now, feel free to check out our API, specifically our built in tracker *Defaults*.

2.3 Session

2.3.1 Introduction

A session is a simple construct that refers to the start and end of a recording session. This recording session takes Probes as inputs and Processors as outputs (More on that in the coming pages).

2.3.2 Managing a session

A Session object is required to manage a session, a session can be constructed by hand, but we provide a simple interface for users to easily initialise a session. You can do this with the following steps.

1. Add the required using

```
using WWellFired.Profile.Unity.Runtime;
```

2. Tell your session to track the data you require.

```
var session = DotProfileSession.New();
```

Now you've got a session, you should start it when you want to start recording data.

```
session.StartRecording();
```

You can also tell your session to stop recording when you don't require it to record anymore.

```
session.StopRecording();
```

Tip: In these examples, we show you how to use and manage a single session, but it's important to mention you're not limited to just one, you can create as many as you want / need.

Sessions will automatically stop recording when your application terminates, but we provide the functionality for you to stop them, in case you want to manage your sessions in a custom manor.

2.3.3 Next up

Sessions are a small topic, with not many functions, but they're the cornerstone of your recording process. We're going to teach you how to make them more useful in the coming pages.

2.4 Probes

2.4.1 Introduction

A probe is a simple construct that allows you to extract data from your game and give it to .Profile. A single .Profile session can having multiple probes into your application. This allows you as a developer a way to keep your game code an your .Profile code separated from one another, imposing a semi strict seperation of concerns.

.Profile ships with a whole bunch of Default *Probes* that allow you to instantly probe most functionality that Unity provides. These Probes are available out of the box and can be used.

All Probes used by .Profile fall into one of two categories

- **Continuous probes** These probes continually probe your application at a specified interval, these probes can be useful for data that changes over time, such as framerate or memory usage
- **One shot probes** One shot probes will probe your app once and once only, these probes can be useful for data that won't change at runtime, such as hardware stats.

2.4.2 Provided Probes

.Profiles Default *Probes* are split fairly evenly between continuous and one shot probes. Out of the box, .Profile provides probes for the following data.

Name	Probe Type	Tracks
Defaults.Continuous	Continuous	CPU load and framerate
Defaults.ContinuousMemory	Continuous	Memory usage across many systems
Defaults.ContinuousObjectMemory	Continuous	Memory in use by various GameObjects
Defaults.Graphics	One Shot	The current graphics settings
Defaults.GraphicsParticles	One Shot	The current graphics settings for particles
Defaults.GraphicsShadow	One Shot	The current graphics settings for shadows
Defaults.GraphicsTexture	One Shot	The current graphics settings for textures
Defaults.Hardware	One Shot	The current hardware settings
Defaults.Time	One Shot	The current time settings

To use one of these Default *Probes*, simply

1. Add the required using

```
using WellFired.Profile.ProfileProcessor;
```

2. Tell your session to track the data you require.

```
// Track Continuous memory usage
session.Track(Defaults.ContinuousMemory);
```

2.4.3 Custom Probes

.Profile provides a simple API for users who want to provide custom probes, allowing users to track any custom data in their application.

Building custom probes is as simple as implementing the *IProbe* interface.

consider the following simple custom probe that doesn't do anything other than return the float 60.

```
public class CustomProbe : IProbe
{
    /// <summary>
    /// Return a value
    /// </summary>
    /// <returns>Always returns 60</returns>
    public object Probe()
    {
        return 60.0f;
    }
}
```

The *IProbe* interface requires you implement the Probe method. From this method you are required to return your custom tracked data, this can be anything you want it to be, it doesn't even have to be performance related (maybe a custom in-game currency spent over time).

Further to this simple example, you can also implement the *IFormattedName* interface, this gives you a method for naming your probe, but is entirely optional.

consider the following extensions:

```
public class CustomProbe : IProbe, IFormattedName
{
    /// <summary>
    /// Return a value
    /// </summary>
    /// <returns>Always returns 60</returns>
    public object Probe()
    {
        return 60.0f;
    }

    /// <summary>
    /// The IFormattedName is optional
    /// </summary>
    public string Name { get { return "Custom Probe"; } }
}
```

Now that you have your custom probe, the last step is to tell your .Profile session to track it.

```
session.Track(new CustomProbe(), RecordMode.Continous, 100);
```

When Tracking a custom probe, you are required to tell your session if it should be probed continuously or only once and at what interval. This corresponds to the second and third parameter respectively.

A single session can have any number of probes.

2.4.4 Next up

Now we've covered tracking data using built in probes and writing custom probes to track custom application logic and data, we'll cover reporting that data.

2.5 Processors

2.5.1 Introduction

A processor facilitates .Profiles processing of data, if probes extract data, processors process them into a usable format. .Profile comes with a selection of processors built in, allowing you to get up and running with little to no effort.

think of probes as the input and processors as the output, with a .Profile sessions sitting in the middle, controlling the whole thing.

2.5.2 Provided Processors

.Profile provides a selection of processors out of the box, ranging from writing to file, to displaying on screen data tracked by each session

FileDumpProcessor	Dump a file after the session is stopped
FileStreamerProcessor	Streams the sessions data to disk
NetworkDumpProcessor	Dumps a file over a network connection when a session is stopped
NetworkStreamerProcessor	Streams the session data to a location over a network
VisualProcessor	A simple built in visualiser for .Profile
DebugLogProcessor	Will Log all tracked data to the Unity Console

To use one of these built in processors, simply

1. Add the required using

```
using WellFired.Profile.ProfileProcessor;
using WellFired.Profile.Unity.Runtime.ProfileProcessor
```

2. Tell your session to use the required Processor

```
// Process recorded data using the built in visual display extension
session.ProcessData(new VisualProcessor(() => Input.GetKeyDown(KeyCode.
↪A)));
```

2.5.3 Custom Processor

.Profile provides a simple API for users who want to provide custom processors. This could be useful for customers who wish to handle processing .Profiles data in a custom manor, for example sending tracked data to an analytics server, or a custom endpoint.

Building custom probes is as simple as implementing the *IProfileProcessor* interface.

Consider the following extension:

```
public class CustomProcessor : IProfileProcessor
{
    private readonly List<TimeValueTable> _timeValueTables = new List<TimeValueTable>
↪();
    private ProbeRecorder[] _probeRecorders;

    public void RecordingStarted(ProbeRecorder[] probeRecorders)
    {
        _probeRecorders = probeRecorders;
        foreach (var probeRecorder in probeRecorders)
            _timeValueTables.Add(new TimeValueTable(probeRecorder.ProbeName()));
    }

    public void RecordingUpdated()
    {
        for (var i = 0; i < _probeRecorders.Length; i++)
        {
            if (_probeRecorders[i].Updated)
                _timeValueTables[i].AddPoint(_probeRecorders[i].
↪GetLastRecordedValue());
        }
    }

    public void RecordingStopped()
    {

    }
}
```

Now that you have your custom processor, the last step is to tell your .Profile session to use it.

```
session.ProcessData(new CustomProcessor());
```

A single session is capable of having multiple processors.

CHAPTER 3

Advanced

3.1 Introduction

One of the more useful, but slightly more advanced functionality of .Profile is the ability to launch your application from the command line with a selection of command line arguments allowing you to automatically start a custom recording session. This doesn't require any code on your side, which is precisely why this functionality is so amazing, it just works out of the box.

3.2 Usage

When you install .Profile, you also install a pre configured command line profile configuration. It's located in your project and called Framerate.pcfg. Feel free to open this and have a look at the format (it's plain Json). We'll use this bundled configuration to show you how .Profile can leverage out of the box command line profiling.

> **Warning:** .Profile ships with a link.xml file, this file tells unity not to strip out .Profile's assemblies when it builds a new player, it's important you have this link.xml file in your project prior to building your player.

1. After installing .Profile, build a player for your desired platform.

2) Run your player with the following command line arguments

```
path/to/your/player.exe  -DotProfileConfig "Framerate"
```

And that's it, your application should be launched and a session will automatically be created using the Framerate.pcfg config.

Note: You can create your own profile configuration files and ship them with your product, if you want to enable a user facing command line interface, allowing your customers the option to profile their existing game. Similar to how Valve ship a profiler with their hammer based products.

CHAPTER 4

.Profile API

4.1 Classes

4.1.1 CommandRunner

Namespace: WellFired.Profile

Description

Public Static Methods

void	*StartRecording* ( )

Public Methods

	CommandRunner (*IGameEventSender* eventSender, *IJSONConfigLoader* jsonConfigLoader, string[] args)
void	*Start* ( )
void	*Stop* ( )

Breakdown

- void **StartRecording** ( )
- **CommandRunner** (*IGameEventSender* eventSender, *IJSONConfigLoader* jsonConfigLoader, string[] args)
- void **Start** ( )
- void **Stop** ( )

4.1.2 ProbeNotFoundException

Namespace: WellFired.Profile.Config

Description

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.3 ProcessorNotFoundException

Namespace: WellFired.Profile.Config

Description

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.4 ParameterConfig

Namespace: WellFired.Profile.Config

Description

Public Properties

readonly string	*ParameterName*
readonly object	*Value*

Public Methods

	ParameterConfig (string parameterName, object value)

Breakdown

- readonly string **ParameterName**
- readonly object **Value**
- **ParameterConfig** (string parameterName, object value)

4.1.5 ProcessorConfig

Namespace: WellFired.Profile.Config

Description

Public Properties

readonly string	*ProcessorName*
readonly :ref:‘ParameterConfig<classwellfired_profile_config_processor_parameterconfig>‘[]	*Parameters*

Public Methods

	ProcessorConfig (string processorName, :ref:‘ParameterConfig<classwellfired_profile_config_processor_parameterconfig>‘[] parameters)

Breakdown

- readonly string **ProcessorName**
- readonly :ref:‘ParameterConfig<classwellfired_profile_config_processor_parameterconfig>‘[] **Parameters**
- **ProcessorConfig** (string processorName, :ref:‘ParameterConfig<classwellfired_profile_config_processor_parameterconfig>‘[] parameters)

4.1.6 ProfileConfig

Namespace: WellFired.Profile

Description

Public Properties

readonly :ref:‘TrackerConfig<classwellfired_profile_config_tracker_trackerconfig>‘[]	*Trackers*
readonly :ref:‘ProcessorConfig<classwellfired_profile_config_processor_processorconfig>‘[]	*Processors*

Public Methods

	ProfileConfig (:ref:‘TrackerConfig<classwellfired_profile_config_tracker_trackerconfig>‘[] trackers, :ref:‘ProcessorConfig<classwellfired_profile_config_processor_processorconfig>‘[] processors)

Breakdown

- readonly :ref:‘TrackerConfig<classwellfired_profile_config_tracker_trackerconfig>‘[] **Trackers**
- readonly :ref:‘ProcessorConfig<classwellfired_profile_config_processor_processorconfig>‘[] **Processors**
- **ProfileConfig** (:ref:‘TrackerConfig<classwellfired_profile_config_tracker_trackerconfig>‘[] trackers, :ref:‘ProcessorConfig<classwellfired_profile_config_processor_processorconfig>‘[] processors)

4.1.7 ProfileConfigRunner

Namespace: WellFired.Profile

Description

Public Methods

	ProfileConfigRunner (*ISession* session, *ProfileConfigLoader* profileConfigLoader, string configName)
void	*RunSession* ( )
void	*StopSession* ( )

Breakdown

- **ProfileConfigRunner** (*ISession* session, *ProfileConfigLoader* profileConfigLoader, string configName)
- void **RunSession** ( )
- void **StopSession** ( )

4.1.8 TrackerConfig

Namespace: WellFired.Profile.Config

Description

Properties

string	*TrackerName* { get; set; }
int	*Interval* { get; set; }

Public Methods

	TrackerConfig (string trackerName, int interval)

Breakdown

- string **TrackerName** { get; set; }
- int **Interval** { get; set; }
- **TrackerConfig** (string trackerName, int interval)

4.1.9 AssemblyHelper

Namespace: WellFired.Profile.Config

Description

Public Static Methods

T	GetImplementationOf (string specificTypeName = null)
Type	GetTypeOfImplementationOf (string specificTypeName = null)

Breakdown

- T **GetImplementationOf< T >** (string specificTypeName = null)
- Type **GetTypeOfImplementationOf< T >** (string specificTypeName = null)

4.1.10 JSONToProfileConfig

Namespace: WellFired.Profile.Config

Description

Public Static Methods

ProfileConfig	*GetProfileConfig* (string json)

Breakdown

- *ProfileConfig* **GetProfileConfig** (string json)

4.1.11 ProfileConfigHelper

Namespace: WellFired.Profile.Config

Description

Public Static Methods

:ref:‘ProbeSetting<structwellfired_profile_config_utils_profileconfighelper_probesetting>’[]	*GetProbeSettings* (*ProfileConfig* profileConfig)
:ref:‘IProfileProcessor<interfacewellfired_profile_profileprocessor_iprofileprocessor>’[]	*GetProcessors* (*ProfileConfig* profileConfig)

Breakdown

- ProbeSetting profileConfig)
- IProfileProcessor profileConfig)

4.1.12 ProfileConfigLoader

Namespace: WellFired.Profile.Config

Description

Public Methods

	ProfileConfigLoader (*IJSONConfigLoader* jsonConfigLoader)
ProfileConfig	*Load* (string configName)

Breakdown

- **ProfileConfigLoader** (*IJSONConfigLoader* jsonConfigLoader)
- *ProfileConfig* **Load** (string configName)

4.1.13 NoImplementationFound

Namespace: WellFired.Profile

Description

Public Properties

override string	*Message*

Public Methods

	NoImplementationFound (Type interfaceType)

Breakdown

- override string **Message**
- **NoImplementationFound** (Type interfaceType)

4.1.14 ProbeAlreadyAdded

Namespace: WellFired.Profile

Description

This will be thrown if you attempt to track too many probes.

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.15 ProfileAlreadyRecording

Namespace: WellFired.Profile

Description

This exception will be called if you've already attempted to start recording with this profile

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.16 ProfileNotRecording

Namespace: WellFired.Profile

Description

This exception will be thrown if you have not yet started recording, but you should have.

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.17 CustomProbe

Namespace: WellFired.Profile

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*

Description

A custom Probe can be used if you want to add custom data to your profiling session.

Properties

string	*Name* { get; set; }

Public Methods

	CustomProbe (Func< object > method, string name)
object	*Probe* ( )

Breakdown

- string **Name** { get; set; }
- **CustomProbe** (Func< object > method, string name)

  Description

  Constructs a new instance of *CustomProbe*

  Parameters

method	The method that will return your custom data
name	Name this probe something sensible

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.18 DefaultProbe

Namespace: WellFired.Profile

Description

A default probe can be provided by the system, a default probe will have the probe, the record mode and the interval.

Properties

IProbe	*Probe* { get; set; }
RecordMode	*RecordMode* { get; set; }
int	*Interval* { get; set; }

Breakdown

- *IProbe* **Probe** { get; set; }
- RecordMode **RecordMode** { get; set; }
- int **Interval** { get; set; }

4.1.19 ProbeEqualityComparer

Namespace: WellFired.Profile

Description

Public Methods

bool	*Equals* (*IProbe* x, *IProbe* y)
int	*GetHashCode* (*IProbe* obj)

Breakdown

- bool **Equals** (*IProbe* x, *IProbe* y)
- int **GetHashCode** (*IProbe* obj)

4.1.20 ProbeHashSet

Namespace: WellFired.Profile

Implements: *WellFired.Profile.Probes.IProbeCollection*

Description

Public Properties

int	*ProbeCount*

Public Methods

bool	*AddProbe* (*IProbe* probe)
bool	*RemoveProbe* (*IProbe* probe)

Breakdown

- int **ProbeCount**
- bool **AddProbe** (*IProbe* probe)

  Description

  Adds a new probe to the *IProbeCollection*

  Parameters

probe	The probe to add

- bool **RemoveProbe** (*IProbe* probe)

  Description

  Removes a passed probe from the *IProbeCollection*

  Parameters

probe	The probe to remove

4.1.21 ProbeRecorder

Namespace: WellFired.Profile

Description

A probe recorder contains a Probe and a Record Mode. This is used to determine what the *ProbeRecorder* will record and also, at what interval the recorder will refresh it's probe

Properties

bool	*Updated* { get; set; }

Public Properties

readonly RecordMode	*RecordMode*

Public Methods

string	*ProbeName* ( )
Type	*ProbeType* ( )
	ProbeRecorder (*IProbe* probe, RecordMode recordMode = RecordMode.Continous, int interval = 0)
void	*RecordSample* (long deltaTime)
TimeValue	*GetLastRecordedValue* ( )
TimeValue	*GetLastFormattedRecordedValue* ( )

Breakdown

- bool **Updated** { get; set; }

  Description

  This flag is reset if the probe has recorded using the RecordSample method.

- readonly RecordMode **RecordMode**

  Description

  The RecordMode that states how this Recorder will record data

- string **ProbeName** ( )

  Description

  Returns the probe name for this recorder. This Probe name will either be specified on the probe or it will be specified using an *IFormattedName* on the probe object

- Type **ProbeType** ( )

  Description

  Returns the raw type of the probe.

- **ProbeRecorder** (*IProbe* probe, RecordMode recordMode = RecordMode.Continous, int interval = 0)

  Description

  Constructs a new instance of *ProbeRecorder*, by default every probe will be continuous, but if you want to optimise your probes, make sure to use Continuous only when needed. You can optionally specify the record interval, this works in conjunction with the RecordMode.Continuous, but is ignored when recordMode is OneShot

  Parameters

probe	The probe which we will get our data from
recordMode	How we will record this data
interval	The interval that we will record this data

- void **RecordSample** (long deltaTime)

Description

Tells the recorder to attempt to record a new sample, note this might not actually record anything, it's up to the Recorder to do the recording, calling this method is simply a way of flagging this recorder to record, if needed

Parameters

deltaTime

- TimeValue **GetLastRecordedValue ( )**

  Description

  Returns the last recorded value as a TimeValue

- TimeValue **GetLastFormattedRecordedValue ( )**

  Description

  Rrturns the last recorded formatted value as a TimeValue

4.1.22 Timer

Namespace: WellFired.Profile

Implements: *WellFired.Profile.Probes.ITimer*

Description

Public Methods

	Timer ( )
long	*GetTime* ( )

Breakdown

- **Timer ( )**
- long **GetTime ( )**

4.1.23 ProfileExtension

Namespace: WellFired

Description

Public Static Methods

void	*InjectConfig* (this *ISession* session, *ProfileConfig* profileConfig)

Breakdown

- void **InjectConfig** (this *ISession* session, *ProfileConfig* profileConfig)

4.1.24 ConnectionFailedException

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Public Properties

override string	*Message*

Public Methods

	ConnectionFailedException (IPAddress address, int port)

Breakdown

- override string **Message**
- **ConnectionFailedException** (IPAddress address, int port)

4.1.25 FailedDumpFileException

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Properties

override string	*Message* { get; set; }

Public Methods

	FailedDumpFileException (string message)

Breakdown

- override string **Message** { get; set; }
- **FailedDumpFileException** (string message)

4.1.26 IPInvalidException

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.27 ContentWriter

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Public Methods

	ContentWriter (Stream stream)
void	*Write* (string content)
void	*WriteLine* (string line)

Breakdown

- **ContentWriter** (Stream stream)
- void **Write** (string content)
- void **WriteLine** (string line)

4.1.28 FileDumpProcessor

Namespace: WellFired.Profile

Inherits: *WellFired.Profile.ProfileProcessor.FileProcessor*

Description

A custom Processor that will dump all recorded data to disk once the recording has stopped. If you want to get all data regardless of application success or failure, you might consider using the FileStreamProcessor

Public Methods

	FileDumpProcessor (string filepath)
	FileDumpProcessor (Stream stream)
override void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
override void	*RecordingUpdated* ( )
override void	*RecordingStopped* ( )

Breakdown

- **FileDumpProcessor** (string filepath)

  Description

  Constructs a new *FileDumpProcessor*.

  Parameters

filepath	The file you’d like to dump your recording data into

- **FileDumpProcessor** (Stream stream)

  Description

  Constructs a new *FileDumpProcessor*, use this variant if you want to manage the Stream yourself.

  Parameters

stream

- override void **RecordingStarted** (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- override void **RecordingUpdated** ( )

  Description

  The recording data has been updated.

- override void **RecordingStopped** ( )

  Description

  The recording has stopped.

4.1.29 FileProcessor

Namespace: WellFired.Profile

Implements: *WellFired.Profile.ProfileProcessor.IProfileProcessor*

Description

This is an abstract implementation of a FileStreamProcessor

protected-attrib

readonly Stream	*Stream*
:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[]	*ProbeRecorders*
readonly bool	*StreamSetManually*

protected-func

	FileProcessor (Stream stream)

Public Methods

void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
void	*RecordingUpdated* ( )
abstract void	*RecordingStopped* ( )

Breakdown

- readonly Stream **Stream**
- :ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] **ProbeRecorders**
- readonly bool **StreamSetManually**
- **FileProcessor** (Stream stream)
- void **RecordingStarted** (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- void **RecordingUpdated** ( )

Description

The recording data has been updated.

- abstract void **RecordingStopped** ( )

Description

The recording has stopped.

4.1.30 FileStreamerProcessor

Namespace: WellFired.Profile

Inherits: *WellFired.Profile.ProfileProcessor.FileProcessor*

Description

A custom Profile Processor that will stream your profiled data to disk. Use this variation if you're worried about your application not exiting cleanly, otherwise, use the *FileProcessor*

Public Methods

	FileStreamerProcessor (string filepath)
	FileStreamerProcessor (Stream stream)
override void	*RecordingUpdated* ( )
override void	*RecordingStopped* ( )

Breakdown

- **FileStreamerProcessor** (string filepath)

Description

Constructs a new instance of a *FileStreamerProcessor*.

Parameters

filepath	The file you'd like to dump your recording data into

- **FileStreamerProcessor** (Stream stream)

Description

Constructs a new instance of *FileStreamerProcessor*, use this variation if you'd like to manage your own stream.

Parameters

stream

- override void **RecordingUpdated** ( )

Description

The recording data has been updated.

- override void **RecordingStopped** ( )

  Description

  The recording has stopped.

4.1.31 NetworkDumpProcessor

Namespace: WellFired.Profile

Implements: *WellFired.Profile.ProfileProcessor.IProfileProcessor*

Description

A custom Processor that will dump your data over a network.

Public Methods

	NetworkDumpProcessor (string address, Protocol protocol = Protocol.TCP)
void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
void	*RecordingUpdated* ( )
void	*RecordingStopped* ( )

Breakdown

- **NetworkDumpProcessor** (string address, Protocol protocol = Protocol.TCP)

  Description

  Creates a new instance of *NetworkDumpProcessor*

  Parameters

address	The Ip you’d like to dump to.
protocol	The Protocol you want to use when dumping

- void **RecordingStarted** (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- void **RecordingUpdated** ( )

  Description

  The recording data has been updated.

- void **RecordingStopped** ( )

  Description

  The recording has stopped.

4.1.32 NetworkStreamerProcessor

Namespace: WellFired.Profile

Implements: *WellFired.Profile.ProfileProcessor.IProfileProcessor*

Description

Public Methods

	NetworkStreamerProcessor (string address, Protocol protocol = Protocol.TCP)
void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
void	*RecordingUpdated* ( )
void	*RecordingStopped* ( )

Breakdown

- **NetworkStreamerProcessor** (string address, Protocol protocol = Protocol.TCP)
- void **RecordingStarted** (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- void **RecordingUpdated** ( )

  Description

  The recording data has been updated.

- void **RecordingStopped** ( )

  Description

  The recording has stopped.

4.1.33 ArrayNode

Namespace: *WellFired.Profile.ProfileProcessor*

Implements: *WellFired.Profile.ProfileProcessor.Node.IComposedNode*

Description

Properties

string	*Name* { get; set; }

Public Properties

int	*ChildrenCount*

Public Methods

	ArrayNode (string name)
INode	*AddChild* (*INode* node)
void	*AddChildren* (IEnumerable< *INode* > nodes)
IEnumerable< *INode* >	*GetChildren* ( )

Breakdown

- string **Name** { get; set; }
- int **ChildrenCount**
- **ArrayNode** (string name)
- *INode* **AddChild** (*INode* node)
- void **AddChildren** (IEnumerable< *INode* > nodes)
- IEnumerable< *INode* > **GetChildren** ( )

4.1.34 NameIsNullOrEmpty

Namespace: WellFired.Profile.ProfileProcessor.Node

Description

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.35 NodeConverter

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Public Methods

	NodeConverter ( )
string	*ConvertToJson* (*INode* node, bool singleLine = false)
string	*ConvertRecordsTo* (IEnumerable< TimeValueTable > timeValueTables)
INode	*ConvertToArrayNode* (TimeValueTable timeValueTable)

Breakdown

- **NodeConverter** ( )
- string **ConvertToJson** (*INode* node, bool singleLine = false)
- string **ConvertRecordsTo** (IEnumerable< TimeValueTable > timeValueTables)
- *INode* **ConvertToArrayNode** (TimeValueTable timeValueTable)

4.1.36 ObjectNode

Namespace: *WellFired.Profile.ProfileProcessor*

Implements: *WellFired.Profile.ProfileProcessor.Node.IComposedNode*

Description

Properties

string	*Name* { get; set; }

Public Properties

int	*ChildrenCount*

Public Methods

	ObjectNode (string name)
INode	*AddChild* (*INode* node)
void	*AddChildren* (IEnumerable< *INode* > nodes)
IEnumerable< *INode* >	*GetChildren* ( )
INode	*GetChild* (string name)

Breakdown

- string **Name** { get; set; }
- int **ChildrenCount**
- **ObjectNode** (string name)
- *INode* **AddChild** (*INode* node)
- void **AddChildren** (IEnumerable< *INode* > nodes)
- IEnumerable< *INode* > **GetChildren** ( )
- *INode* **GetChild** (string name)

4.1.37 ValueNode

Namespace: *WellFired.Profile.ProfileProcessor*

Implements: *WellFired.Profile.ProfileProcessor.Node.INode*

Description

Properties

string	*Name* { get; set; }
object	*Value* { get; set; }

Public Methods

	ValueNode (string name, object value)

Breakdown

- string **Name** { get; set; }
- object **Value** { get; set; }
- **ValueNode** (string name, object value)

4.1.38 GraphCanvasDrawer

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

	GraphCanvasDrawer (*IGraphDisplayer* graphDisplayer, *IVisualObjectFactory* visualObjectFactory)
void	*DrawVerticalLine* (float scaleXInSec, float intervalInSec)
void	*DrawHorizontalLine* (float scaleY, float originY, float interval)

Breakdown

- **GraphCanvasDrawer** (*IGraphDisplayer* graphDisplayer, *IVisualObjectFactory* visualObjectFactory)
- void **DrawVerticalLine** (float scaleXInSec, float intervalInSec)
- void **DrawHorizontalLine** (float scaleY, float originY, float interval)

4.1.39 GraphConfig

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Properties

float	*ScaleX*
float	*ScaleY*
float	*OriginY*
float	*MarkerXInterval*
float	*MarkerYInterval*
string	*XLabel*
string	*YLabel*

Breakdown

- float **ScaleX**

  Description

  Number of seconds that should be displayed on the X axis.

- float **ScaleY**

  Description

  Number of unity that should be displayed on the Y axis. Starting from 0.

- float **OriginY**

  Description

  The value on Y axes at the graph origin. To visualize values between 450 and 400 you would set *ScaleY* to 50, and *OriginY* to 450.

- float **MarkerXInterval**

Description

Distance in second between each vertical lines on the X axis.

- float **MarkerYInterval**

  Description

  Distance between each horizontal lines on the Y axis.

- string **XLabel**

  Description

  Label displayed under the X axis (ex : Time (sec)).

- string **YLabel**

  Description

  Label displayed on the left of the Y axis (ex : Framerate (FPS))

4.1.40 GraphController

Namespace: WellFired.Profile.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.UI.IUIController*

Description

Public Properties

const float	*ReferenceSize*

Public Methods

	GraphController **(** *IUIGraph* graph, *GraphConfig* graphConfig, *IVisualObjectFactory* visualObjectFactory **)**
void	*Update* **( )**
void	*AddProbeRecorders* **(** IEnumerable< *ProbeRecorder* > probeRecorders **)**
void	*DrawPoint* **(** Vect2 point, Color color **)**

Breakdown

- const float **ReferenceSize**
- **GraphController (** *IUIGraph* graph, *GraphConfig* graphConfig, *IVisualObjectFactory* visualObjectFactory **)**
- void **Update ( )**
- void **AddProbeRecorders (** IEnumerable< *ProbeRecorder* > probeRecorders **)**
- void **DrawPoint (** Vect2 point, Color color **)**

4.1.41 GraphLegendDrawer

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

	GraphLegendDrawer (*IUIGraph* graph, *IVisualObjectFactory* visualObjectFactory)
void	*DrawAxesLegend* (string xLabel, string yLabel)
void	*DrawColorLegend* (Color color, string label)

Breakdown

- **GraphLegendDrawer** (*IUIGraph* graph, *IVisualObjectFactory* visualObjectFactory)
- void **DrawAxesLegend** (string xLabel, string yLabel)
- void **DrawColorLegend** (Color color, string label)

4.1.42 GraphRecorder

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

	GraphRecorder (*ProbeRecorder* probeRecorder, *GraphController* graphController, Color color)
void	*Update* ( )

Breakdown

- **GraphRecorder** (*ProbeRecorder* probeRecorder, *GraphController* graphController, Color color)
- void **Update** ( )

4.1.43 PointDrawer

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

	PointDrawer (*IGraphDisplayer* graphDisplayer, *PointPool* pointPool)
void	*DrawPoint* (int x, int y, Color color)

Breakdown

- **PointDrawer** (*IGraphDisplayer* graphDisplayer, *PointPool* pointPool)
- void **DrawPoint** (int x, int y, Color color)

4.1.44 PointPool

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

	PointPool (int count, float pointSize, *IGraphDisplayer* graphDisplayer, *IVisualObjectFactory* visualObjectFactory)
IRectangle	*GetPoint* (Color color)
void	*RecyclePoint* (*IRectangle* point)

Breakdown

- **PointPool** (int count, float pointSize, *IGraphDisplayer* graphDisplayer, *IVisualObjectFactory* visualObjectFactory)
- *IRectangle* **GetPoint** (Color color)
- void **RecyclePoint** (*IRectangle* point)

4.1.45 UIDataView

Namespace: WellFired.Profile.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Group.IUIDataView*

Description

Properties

string	*ProbeName* { get; set; }
string	*ProbeValue* { get; set; }
Vect2	*Size* { get; set; }
Vect2	*Position* { get; set; }
Color	*Color* { get; set; }

Public Methods

	UIDataView (*IRectangle* background, *IText* name, *IText* value)
void	*SetParent* (object parent)
void	*Show* ( )
void	*Recycle* ( )

Breakdown

- string **ProbeName** { get; set; }
- string **ProbeValue** { get; set; }
- Vect2 **Size** { get; set; }
- Vect2 **Position** { get; set; }
- Color **Color** { get; set; }
- **UIDataView** (*IRectangle* background, *IText* name, *IText* value)
- void **SetParent** (object parent)
- void **Show** ( )
- void **Recycle** ( )

4.1.46 UIDataViewController

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

	UIDataViewController (*IUIDataView* uiDataView, *ProbeRecorder* probeRecorder)
void	*Update* ( )

Breakdown

- **UIDataViewController** (*IUIDataView* uiDataView, *ProbeRecorder* probeRecorder)
- void **Update** ( )

4.1.47 UIDataViewInstantiator

Namespace: WellFired.Profile.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Group.IUIDataViewInstantiator*

Description

Public Methods

	UIDataViewInstantiator (*IVisualObjectFactory* visualObjectFactory)
IUIDataView	*Instantiate* ( )

Breakdown

- **UIDataViewInstantiator** (*IVisualObjectFactory* visualObjectFactory)
- *IUIDataView* **Instantiate** ( )

4.1.48 UIGroupController

Namespace: WellFired.Profile.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.UI.IUIController*

Description

public-static-attrib

readonly Color	*OddLineColor*
readonly Color	*PairLineColor*

Public Methods

	UIGroupController (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, *IUIDataViewInstantiator* instanciator, string title)
void	*Update* ( )
void	*AddProbeRecorders* (IEnumerable< *ProbeRecorder* > enumerableProbeRecorders)

Breakdown

- readonly Color **OddLineColor**
- readonly Color **PairLineColor**
- **UIGroupController** (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, *IUIDataViewInstantiator* instanciator, string title)
- void **Update** ( )
- void **AddProbeRecorders** (IEnumerable< *ProbeRecorder* > enumerableProbeRecorders)

4.1.49 MultipleInstantiationException

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Public Properties

override string	*Message*

Breakdown

- override string **Message**

4.1.50 ObjectBuilder

Namespace: *WellFired.Profile.ProfileProcessor*

Implements: *WellFired.Profile.ProfileProcessor.Visual.IObjectBuilder*

Description

Properties

IObjectBuilder	*Instance* { get; set; }

Public Methods

IUIController	*GetStaticUIGroupController* (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
IUIController	*GetDynamicUIGroupController* (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
IUIController	*GetGraphController* (*IUIGraph* graph, *GraphConfig* graphConfig, *IVisualObjectFactory* visualObjectFactory)

Breakdown

- *IObjectBuilder* **Instance** { get; set; }
- *IUIController* **GetStaticUIGroupController** (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
- *IUIController* **GetDynamicUIGroupController** (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
- *IUIController* **GetGraphController** (*IUIGraph* graph, *GraphConfig* graphConfig, *IVisualObjectFactory* visualObjectFactory)

4.1.51 VisualProcessor

Namespace: WellFired.Profile

Implements: *WellFired.Profile.ProfileProcessor.IProfileProcessor*

Description

A custom processor that has a visual representation

Properties

GraphConfig	*GraphConfig* { get; set; }

Public Methods

	VisualProcessor (Func< bool > toggleUIFunc, IEnumerable< Type > additionalProbeTypesToProbeToGraph)
	VisualProcessor (IEnumerable< Type > additionalProbeTypesToProbeToGraph)
	VisualProcessor ( )
	VisualProcessor (Func< bool > toggleUIFunc)
	VisualProcessor (*IUILoader* uiLoader, Func< bool > toggleUIFunc, *IRuntimeTaskLooper* runtimeTaskLooper)
void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
void	*RecordingUpdated* ( )
void	*RecordingStopped* ( )

Breakdown

- *GraphConfig* **GraphConfig** { get; set; }

  Description

  This gives you access to the different parameters to configure your graph. Note that once your Profile session started, any modification to the graph config won't have any impact.

- **VisualProcessor** (Func< bool > toggleUIFunc, IEnumerable< Type > additionalProbeTypesToProbeToGraph)

  Description

  Constructs a new instance of *VisualProcessor*.

  Parameters

toggleUIFunc	An optional delegate that controls the display of the Visual Processor. This delegate should return true whenever you want to toggle the display state of the UI.
additionalProbeTypesToProbeToGraph	An option IEnumerable of additional probe types to add to the graph

- **VisualProcessor** (IEnumerable< Type > additionalProbeTypesToProbeToGraph)

  Description

  Constructs a new instance of *VisualProcessor*.

Parameters

additionalProbeTypesToProbeToGraph	An option IEnumerable of additional probe types to add to the graph

- **VisualProcessor ( )**

  Description

  Constructs a new instance of *VisualProcessor*.

- **VisualProcessor (** Func< bool > toggleUIFunc **)**

  Description

  Constructs a new instance of *VisualProcessor*.

  Parameters

toggleUIFunc	An optional delegate that controls the display of the Visual Processor. This delegate should return true whenever you want to toggle the display state of the UI.

- **VisualProcessor (** *IUILoader* uiLoader, Func< bool > toggleUIFunc, *IRuntimeTaskLooper* runtimeTaskLooper **)**

  Description

  Constructs a new instance of *VisualProcessor*

  Parameters

uiLoader	The object which deals with loading the UI
toggleUIFunc	A delegate that will deal with toggling the visual display. This delegate should return true whenever you want to toggle the display state of the UI.
runtimeTaskLooper	

- void **RecordingStarted (** :ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders **)**

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- void **RecordingUpdated ( )**

  Description

  The recording data has been updated.

- void **RecordingStopped ( )**

  Description

The recording has stopped.

4.1.52 ProfileSharedResources

Namespace: WellFired

Description

Properties

IRuntimeConnector	*RuntimeConnector* { get; set; }

public-static-attrib

ILogger	*Logger*

Public Static Methods

void	*Initialize* (*IRuntimeConnector* runtimeConnector)

Breakdown

- *IRuntimeConnector* **RuntimeConnector** { get; set; }
- *ILogger* **Logger**
- void **Initialize** (*IRuntimeConnector* runtimeConnector)

4.1.53 Session

Namespace: WellFired

Implements: *WellFired.Profile.ISession*

Description

Our basic profiler object

Public Methods

	Session ( )
	Session (*ITimer* timer, *IRuntimeTaskLooper* runtimeTaskLooper)
void	*Track* (*IProbe* probe, RecordMode recordMode = RecordMode.Continous, int interval = 0)
void	*Track* (Func< object > method, RecordMode recordMode = RecordMode.Continous, int interval = 0)
void	*Track* (IEnumerable< *DefaultProbe* > probes)
void	*ProcessData* (*IProfileProcessor* processor)
void	*StartRecording* ( )
void	*StopRecording* ( )

Breakdown

- **Session** ( )

  Description

  Creates a new instance of Profile, with default settings, you'll likely only need to call this

- **Session** (*ITimer* timer, *IRuntimeTaskLooper* runtimeTaskLooper)

  Description

  Creates a new instance of Profile allowing you to provide a custom timer and task looper

  Parameters

timer
runtimeTaskLooper

- void **Track** (*IProbe* probe, RecordMode recordMode = RecordMode.Continous, int interval = 0)

  Description

  Tracks a given probe, with the passed data

  Parameters

probe	The probe that we should track
recordMode	The record mode to use when we're getting tracked data
interval	The interval that we would like to use when retrieving tracked data

- void **Track** (Func< object > method, RecordMode recordMode = RecordMode.Continous, int interval = 0)

  Description

  Allows you to track custom data on this profiler, simply pass the method that will extract the custom data as the first parameter

  Parameters

method	A delegate that will be used to extract custom data
recordMode	The record mode to use when we're getting tracked data
interval	The interval that we would like to use when retrieving tracked data

- void **Track** (IEnumerable< *DefaultProbe* > probes)

Description

Tracks a collection of probes.

Parameters

probes

- void **ProcessData** (*IProfileProcessor* processor)

Description

How do you want to process the data. We provide many default processors including the VisualProcessor, which will display data to the screen

Parameters

processor

- void **StartRecording** ( )

Description

Start Recording Data

- void **StopRecording** ( )

Description

Stop recording data

4.1.54 JsonConfigLoader

Namespace: WellFired.Profile.Unity.Runtime

Implements: *WellFired.Profile.Config.Utils.IJSONConfigLoader*

Description

Loads a config file from the DotProfileConfig directory

Public Methods

string	*LoadJsonConfig* (string fileName)

Breakdown

- string **LoadJsonConfig** (string fileName)

Description

Load the actual config, returning the loaded data before deserialization

Parameters

fileName	The file you'd like to load, the file does not need to have the .pcfg extension. the file also does not need to be a full path, the fileName provided should be relative to the DotProfileConfig directory. I.E. If you want SomeDirectory/AnotherDirectory/DotProfileCOnfig/ConfigFile.pcfg, the fileName used should be ConfigFile.

4.1.55 Defaults

Namespace: WellFired.Profile.Unity

Description

Defaults provide a static access to different group of probes (All, Texture, etc...) that can be directly tracked by a IProfile.

public-static-attrib

readonly IList< *DefaultProbe* >	*Continuous*
readonly IList< *DefaultProbe* >	*ContinuousMemory*
readonly IList< *DefaultProbe* >	*ContinuousObjectMemory*
readonly IList< *DefaultProbe* >	*Graphics*
readonly IList< *DefaultProbe* >	*GraphicsParticles*
readonly IList< *DefaultProbe* >	*GraphicsShadow*
readonly IList< *DefaultProbe* >	*GraphicsTexture*
readonly IList< *DefaultProbe* >	*Hardware*
readonly IList< *DefaultProbe* >	*Physics*
readonly IList< *DefaultProbe* >	*Time*

Breakdown

- readonly IList< *DefaultProbe* > **Continuous**

  Description

  Static Helper allowing you to quickly track CpuLoad and Framerate

- readonly IList< *DefaultProbe* > **ContinuousMemory**

  Description

  Static Helper allowing you to quickly track generic memory usage

- readonly IList< *DefaultProbe* > **ContinuousObjectMemory**

  Description

  Static Helper allowing you to quickly track object memory usage (Can be quite slow, especially in editor), better to only used this when you need it.

- readonly IList< *DefaultProbe* > **Graphics**

  Description

  Static Helper allowing you to quickly track Graphic Settings

- readonly IList< *DefaultProbe* > **GraphicsParticles**

  Description

  Static Helper allowing you to quickly track Graphic Particle Settings

- readonly IList< *DefaultProbe* > **GraphicsShadow**

  Description

  Static Helper allowing you to quickly track Graphic Shadow Settings

- readonly IList< *DefaultProbe* > **GraphicsTexture**

  Description

  Static Helper allowing you to quickly track Graphic Texture Settings

- readonly IList< *DefaultProbe* > **Hardware**

  Description

  Static Helper allowing you to quickly track Hardware Statistics

- readonly IList< *DefaultProbe* > **Physics**

  Description

  Static Helper allowing you to quickly track Physics Settings

- readonly IList< *DefaultProbe* > **Time**

  Description

  Static Helper allowing you to quickly track Time Settings

4.1.56 DotProfileSession

Namespace: WellFired.Profile.Unity

Description

A Utility interface that allows users to quickly obtain and manage Sessions.

Public Static Methods

ISession	*New* ( )
void	*StartConfigProfile* ( )
void	*ToggleLog* (bool enabled)

Breakdown

- *ISession* **New ( )**

  Description

  Return a new *ISession* allowing the caller to specify details about a what to record, when to record it and how to process that data.

- void **StartConfigProfile ( )**

Description

Whenever it is called, it will load *.pcfg files in StreamingAssets/DotProfileConfig/ which are files specifying what needs to be tracked and how to process it.

- void **ToggleLog** (bool enabled)

Description

Toggle logs in Unity Debug console. For debug purpose only.

Parameters

enabled

4.1.57 CPULoadProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*

Description

Probes the CPU Load

Public Properties

string	*Name*

Public Methods

	CPULoadProbe ( )
object	*Probe* ( )

Breakdown

- string **Name**
- **CPULoadProbe** ( )
- object **Probe** ( )

Description

Returns the probed data.

4.1.58 FramerateProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

Probes the current Framerate, which is equal to the inverse of the time elapsed in second to coplete the last frame

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.59 AnimationMemoryProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the amount of memory used by AnimationClips

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.60 AudioMemoryProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the amount of memory used by AudioClips

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.61 MonoHeapSizeProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the memory in use by the MonoHeap

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.62 MonoUsedSizeProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the memory in use by the MonoUseddSize

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.63 TextureMemoryProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the Memory used by Textures

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.64 TotalAllocatedMemoryProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the Total Allocated Memory

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.65 TotalReservedMemoryProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the Total Reserved Memory

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.66 TotalUnusedReservedMemoryProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes the total Unused Reserved Memory

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.67 UsedHeapSizeProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

If you Track the Used Heap size and profiling is enabled, this value will be reported and show you the used heap size in MB

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Will return the used heap size in MB

- string **FormattedValue** (object value)

  Description

  Will return the used heap size as a formatted string in MBs

  Parameters

value

4.1.68 ActiveDynamicProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.ContinuousData

Implements: *WellFired.Profile.Probes.IProbe*

Description

Probes the number of active dynamic RigidBodies

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.69 AntiAliasingProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current AntiAliasing Setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.70 AsyncUploadBufferSizeProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Async Upload Buffer Size Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.71 AsyncUploadTimeSliceProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

Profiles Unity's current Async Upload Time Slice Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.72 BlendWeightsProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

Profiles Unity's current Blend Weight Setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.73 LodBiasProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current LOD Bias setting.

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.74 MaximumLodLevelProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's Maximum Load Level Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.75 ParticleRaycastBudgetProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's Particle Raycast Budget Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.76 BillboardFaceCameraPositionProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Billboard Face Camera Position setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.77 SoftParticleProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current Soft Particles settings

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.78 PixelLightCountProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Pixel Light Count Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.79 RealtimeReflectionProbesProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current Realtime Reflection Probes Setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.80 Cascade2SplitsProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Cascade 2 split Shadow settings

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.81 Cascade4SplitsProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current Cascade 4 split setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.82 NearPlaneOffsetProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Shadow Near Plane Offset

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.83 QualityProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Shadow Quality Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.84 ShadowDistanceProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Shadow Distance Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.85 ShadowProjectionProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current Shadow Projection Setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.86 ShadowResolutionProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Shadow Resolution Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.87 AnisotropicFilteringProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current TExture Anisotropic filtering setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.88 QualityProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes.GraphicSettings

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Texture Quality setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.89 VSyncCountProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current VSync Count Setting

Public Properties

string	*Name*

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- string **Name**
- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

From the probed data, return a formatted value.

Parameters

value	The object returned from a probe

4.1.90 CPUModelProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*

Description

Probes our current CPUModel

Public Properties

string	*Name*

Public Methods

object	*Probe* ( )

Breakdown

- string **Name**
- object **Probe** ( )

  Description

  Returns the probed data.

4.1.91 GPUModelProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*

Description

Probes your current GPUModel

Public Properties

string	*Name*

Public Methods

object	*Probe* ( )

Breakdown

- string **Name**
- object **Probe** ( )

  Description

  Returns the probed data.

4.1.92 RAMProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes your current system memory size.

Public Properties

string	*Name*

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- string **Name**
- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.93 VRAMProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedName*, *WellFired.Profile.Probes.IFormattedValue*

Description

Probes your current VRAM Size

Public Properties

string	*Name*

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- string **Name**
- object **Probe** ( )

  Description

  Returns the probed data.
- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.94 BounceThresholdProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Bounce Threshold Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.95 DefaultContactOffsetProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's Default Contact Offest Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.96 DefaultSolverIterationsProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's Default Solver Iterations Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.97 DefaultSolverVelocityIterationsProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Default Solver Velocity Iterations Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.98 GravityProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's Current Gravity Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.99 QueriesHitBackfacesProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's Queries Hit Backfaces Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.100 QueriesHitTriggersProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

Probe Unity's Queries Hit Triggers Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.101 SleepThresholdProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Sleep Threshold Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.102 FixedTimestepProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's Fixed Timestep setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.103 MaximumAllowedTimestepProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current Maximum Allows Timestep setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.104 MaximumParticleTimestepProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*, *WellFired.Profile.Probes.IFormattedValue*

Description

ProbesUnity's current Maximum Particle Timestep setting

Public Methods

object	*Probe* ( )
string	*FormattedValue* (object value)

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.1.105 TimeScaleProbe

Namespace: WellFired.Profile.Unity.Runtime.Probes

Implements: *WellFired.Profile.Probes.IProbe*

Description

ProbesUnity's current Time Scale Setting

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.1.106 DebugLogProcessor

Namespace: WellFired.Profile.Unity.Runtime

Implements: *WellFired.Profile.ProfileProcessor.IProfileProcessor*

Description

A custom Processor that prints all profiler information to the debug console.

Public Methods

void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
void	*RecordingUpdated* ( )
void	*RecordingStopped* ( )

Breakdown

- void **RecordingStarted** (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- void **RecordingUpdated** ( )

  Description

  The recording data has been updated.

- void **RecordingStopped** ( )

  Description

  The recording has stopped.

4.1.107 GraphDisplayer

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Graph.IGraphDisplayer*

Description

Public Properties

int	*SizeX*
int	*SizeY*

Public Methods

void	*Initialize* ( )
object	*GetCanvasParent* ( )
object	*GetPointsParent* ( )
void	*TranslateCoordinate* (Vect2 vector)
IVisualObjectFactory	*GetGraphObjectFactory* ( )

Breakdown

- int **SizeX**
- int **SizeY**
- void **Initialize** ( )
- object **GetCanvasParent** ( )
- object **GetPointsParent** ( )
- void **TranslateCoordinate** (Vect2 vector)
- *IVisualObjectFactory* **GetGraphObjectFactory** ( )

4.1.108 UIGraph

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Group.IUIGraph*

Description

Public Properties

int	*SizeX*
int	*SizeY*
IGraphDisplayer	*GraphDisplayer*

Public Methods

void	*Initialize* ( )

Breakdown

- int **SizeX**
- int **SizeY**
- *IGraphDisplayer* **GraphDisplayer**
- void **Initialize ( )**

4.1.109 UIGroup

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Group.IUIGroup*

Description

Public Properties

Vect2	*Size*

Public Methods

void	*Initialize* ( )
object	*GetTransform* ( )

Breakdown

- Vect2 **Size**
- void **Initialize ( )**
- object **GetTransform ( )**

4.1.110 DefaultUIToggler

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.UI.IDefaultUIToggler*

Description

Properties

Func< bool >	*DefaultFunction* { get; set; }

Breakdown

- Func< bool > **DefaultFunction** { get; set; }

4.1.111 Rectangle

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.UI.IRectangle*

Description

Properties

Vect2	*Size* { get; set; }
Vect2	*Position* { get; set; }
Color	*Color* { get; set; }

Public Methods

void	*SetParent* (object parent)
object	*GetTransform* ( )
void	*Activate* ( )
void	*Deactivate* ( )
void	*Initialize* ( )

Breakdown

- Vect2 **Size** { get; set; }
- Vect2 **Position** { get; set; }
- Color **Color** { get; set; }
- void **SetParent** (object parent)
- object **GetTransform** ( )
- void **Activate** ( )
- void **Deactivate** ( )
- void **Initialize** ( )

4.1.112 Text

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.UI.IText*

Description

Properties

Color	*Color* { get; set; }
string	*Content* { get; set; }
int	*FontSize* { get; set; }
Vect2	*Position* { get; set; }

Public Properties

float	*TextWidth*
float	*TextHeight*

Public Methods

void	*Initialize* ( )
void	*SetParent* (object parent)
void	*Rotate* (float angleInDegree)

Breakdown

- Color **Color** { get; set; }
- string **Content** { get; set; }
- int **FontSize** { get; set; }
- Vect2 **Position** { get; set; }
- float **TextWidth**
- float **TextHeight**
- void **Initialize** ( )
- void **SetParent** (object parent)
- void **Rotate** (float angleInDegree)

4.1.113 UILoader

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Group.IUILoader*

Description

Public Methods

IPromise< *IVisualProcessorUI* >	*LoadUI* ( )
bool	*IsUILoaded* ( )

Breakdown

- IPromise< *IVisualProcessorUI* > **LoadUI** ( )
- bool **IsUILoaded** ( )

4.1.114 VisualObjectFactory

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.UI.IVisualObjectFactory*

Description

Public Methods

IRectangle	*GetRectangle* ( )
IText	*GetText* ( )

Breakdown

- *IRectangle* **GetRectangle** ( )
- *IText* **GetText** ( )

4.1.115 VisualProcessorUI

Namespace: WellFired.Profile.Unity.Runtime.ProfileProcessor.Visual

Implements: *WellFired.Profile.ProfileProcessor.Visual.Group.IVisualProcessorUI*

Description

Public Properties

IUIGroup	*HardwareGroup*
IUIGroup	*StaticGroup*
IUIGroup	*DynamicGroup*
IUIGraph	*Graph*
Type[]	*HardwareProbes*
Type[]	*GraphProbes*

Public Methods

void	*Initialize* ( )
void	*Toggle* ( )
IVisualObjectFactory	*GetVisualObjectFactory* ( )

Breakdown

- *IUIGroup* **HardwareGroup**
- *IUIGroup* **StaticGroup**
- *IUIGroup* **DynamicGroup**
- *IUIGraph* **Graph**
- Type[] **HardwareProbes**
- Type[] **GraphProbes**
- void **Initialize** ( )
- void **Toggle** ( )
- *IVisualObjectFactory* **GetVisualObjectFactory** ( )

4.1.116 RuntimeConnector

Namespace: WellFired.Profile.Unity

Implements: *WellFired.Profile.IRuntimeConnector*

Description

Public Methods

IGameEventSender	*GetGameEventSender* ( )
IJSONConfigLoader	*GetConfigJsonLoader* ( )
IRuntimeTaskLooper	*GetTaskLooper* ( )
IUILoader	*GetUILoader* ( )
IDefaultUIToggler	*GetDefaultUIToggler* ( )
ILogger	*GetLogger* ( )

Breakdown

- *IGameEventSender* **GetGameEventSender** ( )
- *IJSONConfigLoader* **GetConfigJsonLoader** ( )
- *IRuntimeTaskLooper* **GetTaskLooper** ( )
- *IUILoader* **GetUILoader** ( )
- *IDefaultUIToggler* **GetDefaultUIToggler** ( )
- *ILogger* **GetLogger** ( )

4.1.117 ColorConvertor

Namespace: WellFired.Profile.Unity.Runtime

Description

Public Static Methods

Color	*Convert* (Profile.ProfileProcessor.Visual.UI.Color color)

Breakdown

- Color **Convert** (Profile.ProfileProcessor.Visual.UI.Color color)

4.1.118 GameEventSender

Namespace: WellFired.Profile.Unity.Runtime

Implements: *WellFired.Profile.Utils.IGameEventSender*

Description

Events

Action	*OnGameStart*
Action	*OnGameEnd*

Breakdown

- Action **OnGameStart**
- Action **OnGameEnd**

4.1.119 GameEventSenderCreator

Namespace: WellFired.Profile.Unity.Runtime

Implements: *WellFired.Profile.Utils.IGameEventSenderCreator*

Description

Public Methods

IGameEventSender	*Create* ( )

Breakdown

- *IGameEventSender* **Create** ( )

4.1.120 ObjectsTracker

Namespace: WellFired.Profile.Unity.Runtime

Description

Public Static Methods

IEnumerable< Object >	*GetCollectedObjects* ( )

Breakdown

- IEnumerable< Object > **GetCollectedObjects** ( )

4.1.121 ProfileLogger

Namespace: WellFired.Profile.Unity.Runtime

Description

Public Methods

void	*Toggle* (bool enabled)
void	*Log* (string message)
void	*LogWarning* (string message)
void	*LogError* (string message)

Breakdown

- void **Toggle** (bool enabled)
- void **Log** (string message)
- void **LogWarning** (string message)
- void **LogError** (string message)

4.1.122 RuntimeTaskLooper

Namespace: WellFired.Profile.Unity.Runtime

Implements: *WellFired.Profile.Utils.IRuntimeTaskLooper*

Description

Public Methods

void	*Play* (Action a, float interval)
void	*Stop* ( )

Breakdown

- void **Play** (Action a, float interval)

  Description

  Allows to repeat an action over time at specified interval.

  Parameters

a
interval

- void **Stop** ( )

4.1.123 Task

Namespace: WellFired.Profile.Unity.Runtime

Description

Public Properties

bool	*Running*
bool	*Paused*

Events

FinishedHandler	*Finished*

Public Methods

delegate void	*FinishedHandler* (bool manual)
	Task (IEnumerator c, bool autoStart = true)
void	*Pause* ( )
void	*Unpause* ( )
void	*Start* ( )
void	*Stop* ( )
IEnumerator	*CallWrapper* ( )

Breakdown

- bool **Running**
- bool **Paused**
- FinishedHandler **Finished**
- delegate void **FinishedHandler** (bool manual)
- **Task** (IEnumerator c, bool autoStart = true)
- void **Pause** ( )
- void **Unpause** ( )
- void **Start** ( )
- void **Stop** ( )
- IEnumerator **CallWrapper** ( )

4.1.124 TaskManager

Namespace: WellFired.Profile.Unity.Runtime

Description

Public Static Methods

void	*RunTask* (*Task* task)

Public Methods

void	*Update* ( )

Breakdown

- void **RunTask** (*Task* task)
- void **Update** ( )

4.1.125 UnitySystemConsoleRedirector

Namespace: WellFired.Profile.Unity.Runtime

Description

Public Static Methods

void	*Redirect* ( )

Breakdown

- void **Redirect** ( )

4.1.126 UnityTextWriter

Namespace: WellFired.Profile.Unity.Runtime.Utils

Description

Properties

override Encoding	*Encoding* { get; set; }

Public Methods

override void	*Flush* ( )
override void	*Write* (string value)
override void	*Write* (char value)
override void	*Write* (char[] value, int index, int count)

Breakdown

- override Encoding **Encoding** { get; set; }
- override void **Flush** ( )
- override void **Write** (string value)
- override void **Write** (char value)
- override void **Write** (char[] value, int index, int count)

4.1.127 Constants

Namespace: WellFired.Profile

Description

A useful utility class that provides constants

Public Properties

const float	*MbConversion*

Breakdown

- const float **MbConversion**

  Description

  Conversion to Mb

4.1.128 Debug

Namespace: WellFired.Profile

Description

Public Static Methods

void	*Log* (string message)
void	*LogWarning* (string message)
void	*LogError* (string message)

Breakdown

- void **Log** (string message)
- void **LogWarning** (string message)
- void **LogError** (string message)

4.1.129 NameFormatter

Namespace: WellFired.Profile

Description

Public Static Methods

string	*Nicify* (string originalName)

Breakdown

- string **Nicify** (string originalName)

4.1.130 NetworkUtilities

Namespace: WellFired.Profile

Description

Public Static Methods

bool	*IsIpValid* (string ip)

Breakdown

- bool **IsIpValid** (string ip)

4.1.131 TaskLooper

Namespace: WellFired.Profile

Description

Properties

bool	*Playing* { get; set; }

Public Methods

	TaskLooper (*IRuntimeTaskLooper* runtimeTaskLooper, float interval = 0f)
void	*Play* (Action a)
void	*Stop* ( )

Breakdown

- bool **Playing** { get; set; }
- **TaskLooper** (*IRuntimeTaskLooper* runtimeTaskLooper, float interval = 0f)
- void **Play** (Action a)
- void **Stop** ( )

4.2 Interfaces

4.2.1 IJSONConfigLoader

Namespace: WellFired.Profile.Config

Description

Public Methods

string	*LoadJsonConfig* (string fileName)

Breakdown

- string **LoadJsonConfig** (string fileName)

4.2.2 IRuntimeConnector

Namespace: WellFired

Description

Public Methods

IGameEventSender	*GetGameEventSender* ( )
IJSONConfigLoader	*GetConfigJsonLoader* ( )
IRuntimeTaskLooper	*GetTaskLooper* ( )
IUILoader	*GetUILoader* ( )
IDefaultUIToggler	*GetDefaultUIToggler* ( )
ILogger	*GetLogger* ( )

Breakdown

- *IGameEventSender* **GetGameEventSender** ( )
- *IJSONConfigLoader* **GetConfigJsonLoader** ( )
- *IRuntimeTaskLooper* **GetTaskLooper** ( )
- *IUILoader* **GetUILoader** ( )
- *IDefaultUIToggler* **GetDefaultUIToggler** ( )
- *ILogger* **GetLogger** ( )

4.2.3 ISession

Namespace: WellFired

Description

Implement this interface if you'd like to create an object that tracks probes.

Public Methods

void	*StartRecording* ( )
void	*StopRecording* ( )
void	*Track* (*IProbe* probe, RecordMode recordMode = RecordMode.Continous, int interval = 0)
void	*Track* (Func< object > method, RecordMode recordMode = RecordMode.Continous, int interval = 0)
void	*Track* (IEnumerable< *DefaultProbe* > probes)
void	*ProcessData* (*IProfileProcessor* processor)

Breakdown

- void **StartRecording** ( )

  Description

  Will start this profilers recording session. What is tracked and how it is process should be specify beforehand

- void **StopRecording** ( )

  Description

  Will Stop this profilers recording session.

- void **Track** (*IProbe* probe, RecordMode recordMode = RecordMode.Continous, int interval = 0)

  Description

  Call this to specify the probes you want to use to track data. You can add custom probes here, or any of the many pre-created probes. You can record probes continuously or only once when the session is just started (one-shot mode). If you just want an easy to use interface, you can prefer the to use Track(IEnumerable probes)

  Parameters

probe	Here you can pass any probes you want to track.
recordMode	Continous or One-Shot recording
interval	Time interval between each data sampling. Note that the time used is based on the system clock, therefore it is independent from the game time scale

- void **Track** (Func< object > method, RecordMode recordMode = RecordMode.Continous, int interval = 0)

  Description

  Call this to track the data returned by one of your method.

  Parameters

method
recordMode
interval

- void **Track** (IEnumerable< *DefaultProbe* > probes)

  Description

This method works similar to the Track method but record mode and interval are specified by default in the Default *Probes*. You can pass one of the groups of default probes we already provide, like Defaults.All. You should prefer this method if you don't need 100% control over your probes.

Parameters

probes	You can also pass one of the provided utilities like Defaults.All

- void **ProcessData (** *IProfileProcessor* processor **)**

  Description

  How do you want to process the data. We provide many default processors including the VisualProcessor, which will display data to the screen

  Parameters

processor

4.2.4 IFormattedName

Namespace: WellFired.Profile

Description

Use this interface on a custom probe if you want to return a formatted name to the probe system

Properties

string	*Name* { get; set; }

Breakdown

- string **Name** { get; set; }

4.2.5 IFormattedValue

Namespace: WellFired.Profile

Description

Optionally implement this interface if you'd like your custom probe to display data with a custom format

Public Methods

string	*FormattedValue* (object value)

Breakdown

- string **FormattedValue** (object value)

  Description

  From the probed data, return a formatted value.

  Parameters

value	The object returned from a probe

4.2.6 IProbe

Namespace: WellFired.Profile

Description

The interface for a probe, this is a simple interface that provides just one method, this method should return the probed data.

Public Methods

object	*Probe* ( )

Breakdown

- object **Probe** ( )

  Description

  Returns the probed data.

4.2.7 IProbeCollection

Namespace: WellFired.Profile

Description

A probe collection is a collection of probes.

Properties

int	*ProbeCount* { get; set; }

Public Methods

bool	*AddProbe* (*IProbe* probe)
bool	*RemoveProbe* (*IProbe* probe)

Breakdown

- int **ProbeCount** { get; set; }

  Description

  How many probes are in this collection.

- bool **AddProbe** (*IProbe* probe)

  Description

  Adds a new probe to the *IProbeCollection*

  Parameters

probe	The probe to add

- bool **RemoveProbe** (*IProbe* probe)

  Description

  Removes a passed probe from the *IProbeCollection*

  Parameters

probe	The probe to remove

4.2.8 ITimer

Namespace: WellFired.Profile

Description

Public Methods

long	*GetTime* ( )

Breakdown

- long **GetTime** ( )

4.2.9 IProfileProcessor

Namespace: WellFired.Profile

Description

Implement this interface if you want to provide custom profile processors. For example, you could use this functionality if you wanted to create a custom analytics processor, sending all recorded data to an analytics provider, for instance.

Public Methods

void	*RecordingStarted* (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)
void	*RecordingUpdated* ( )
void	*RecordingStopped* ( )

Breakdown

- void **RecordingStarted** (:ref:‘ProbeRecorder<classwellfired_profile_probes_proberecorder>‘[] probeRecorders)

  Description

  The Record has been started

  Parameters

probeRecorders	This recording session will record these probes

- void **RecordingUpdated** ( )

  Description

  The recording data has been updated.

- void **RecordingStopped** ( )

  Description

  The recording has stopped.

4.2.10 IComposedNode

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Properties

int	*ChildrenCount* { get; set; }

Public Methods

INode	*AddChild* (*INode* node)
void	*AddChildren* (IEnumerable< *INode* > nodes)
IEnumerable< *INode* >	*GetChildren* ( )

Breakdown

- int **ChildrenCount** { get; set; }
- *INode* **AddChild** (*INode* node)
- void **AddChildren** (IEnumerable< *INode* > nodes)
- IEnumerable< *INode* > **GetChildren** ( )

4.2.11 INode

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Properties

string	*Name* { get; set; }

Breakdown

- string **Name** { get; set; }

4.2.12 IGraphDisplayer

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

int	*SizeX* { get; set; }
int	*SizeY* { get; set; }

Public Methods

void	*Initialize* ( )
object	*GetCanvasParent* ( )
object	*GetPointsParent* ( )
void	*TranslateCoordinate* (Vect2 vector)

Breakdown

- int **SizeX** { get; set; }
- int **SizeY** { get; set; }
- void **Initialize** ( )
- object **GetCanvasParent** ( )
- object **GetPointsParent** ( )
- void **TranslateCoordinate** (Vect2 vector)

4.2.13 IUIDataView

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

string	*ProbeName* { get; set; }
string	*ProbeValue* { get; set; }
Vect2	*Size* { get; set; }
Vect2	*Position* { get; set; }
Color	*Color* { get; set; }

Public Methods

void	*SetParent* (object parent)
void	*Show* ( )
void	*Recycle* ( )

Breakdown

- string **ProbeName** { get; set; }
- string **ProbeValue** { get; set; }
- Vect2 **Size** { get; set; }
- Vect2 **Position** { get; set; }

- Color **Color** { get; set; }
- void **SetParent** (object parent)
- void **Show** ( )
- void **Recycle** ( )

4.2.14 IUIDataViewInstantiator

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

IUIDataView	*Instantiate* ( )

Breakdown

- *IUIDataView* **Instantiate** ( )

4.2.15 IUIGraph

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

int	*SizeX* { get; set; }
int	*SizeY* { get; set; }
IGraphDisplayer	*GraphDisplayer* { get; set; }

Public Methods

void	*Initialize* ( )

Breakdown

- int **SizeX** { get; set; }
- int **SizeY** { get; set; }
- *IGraphDisplayer* **GraphDisplayer** { get; set; }
- void **Initialize** ( )

4.2.16 IUIGroup

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

Vect2	*Size* { get; set; }

Public Methods

void	*Initialize* ( )
object	*GetTransform* ( )

Breakdown

- Vect2 **Size** { get; set; }
- void **Initialize** ( )
- object **GetTransform** ( )

4.2.17 IUILoader

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

IPromise< *IVisualProcessorUI* >	*LoadUI* ( )
bool	*IsUILoaded* ( )

Breakdown

- IPromise< *IVisualProcessorUI* > **LoadUI** ( )
- bool **IsUILoaded** ( )

4.2.18 IVisualProcessorUI

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

IUIGroup	*HardwareGroup* { get; set; }
IUIGroup	*StaticGroup* { get; set; }
IUIGroup	*DynamicGroup* { get; set; }
IUIGraph	*Graph* { get; set; }
Type[]	*HardwareProbes* { get; set; }
Type[]	*GraphProbes* { get; set; }

Public Methods

void	*Initialize* ( )
void	*Toggle* ( )
IVisualObjectFactory	*GetVisualObjectFactory* ( )

Breakdown

- *IUIGroup* **HardwareGroup** { get; set; }
- *IUIGroup* **StaticGroup** { get; set; }
- *IUIGroup* **DynamicGroup** { get; set; }
- *IUIGraph* **Graph** { get; set; }
- Type[] **HardwareProbes** { get; set; }
- Type[] **GraphProbes** { get; set; }
- void **Initialize** ( )
- void **Toggle** ( )
- *IVisualObjectFactory* **GetVisualObjectFactory** ( )

4.2.19 IObjectBuilder

Namespace: *WellFired.Profile.ProfileProcessor*

Description

Public Methods

IUIController	*GetStaticUIGroupController* (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
IUIController	*GetDynamicUIGroupController* (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
IUIController	*GetGraphController* (*IUIGraph* graph, *GraphConfig* graphConfig, *IVisualObjectFactory* visualObjectFactory)

Breakdown

- *IUIController* **GetStaticUIGroupController** (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
- *IUIController* **GetDynamicUIGroupController** (*IUIGroup* group, int columnCount, *IVisualObjectFactory* visualObjectFactory, string title)
- *IUIController* **GetGraphController** (*IUIGraph* graph, *GraphConfig* graphConfig, *IVisualObjectFactory* visualObjectFactory)

4.2.20 IDefaultUIToggler

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Used by *VisualProcessor*. It define what should be the default condition under which UI should be displayed.

Properties

Func< bool >	*DefaultFunction* { get; set; }

Breakdown

- Func< bool > **DefaultFunction** { get; set; }

4.2.21 IRectangle

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

Color	*Color* { get; set; }
Vect2	*Size* { get; set; }
Vect2	*Position* { get; set; }

Public Methods

void	*SetParent* (object parent)
object	*GetTransform* ( )
void	*Activate* ( )
void	*Deactivate* ( )

Breakdown

- Color **Color** { get; set; }
- Vect2 **Size** { get; set; }
- Vect2 **Position** { get; set; }
- void **SetParent** (object parent)
- object **GetTransform** ( )
- void **Activate** ( )
- void **Deactivate** ( )

4.2.22 IText

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Properties

Color	*Color* { get; set; }
string	*Content* { get; set; }
int	*FontSize* { get; set; }
Vect2	*Position* { get; set; }
float	*TextWidth* { get; set; }
float	*TextHeight* { get; set; }

Public Methods

void	*SetParent* (object parent)
void	*Rotate* (float angleInDegree)

Breakdown

- Color **Color** { get; set; }
- string **Content** { get; set; }
- int **FontSize** { get; set; }
- Vect2 **Position** { get; set; }
- float **TextWidth** { get; set; }
- float **TextHeight** { get; set; }
- void **SetParent** (object parent)
- void **Rotate** (float angleInDegree)

4.2.23 IUIController

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

void	*AddProbeRecorders* (IEnumerable< *ProbeRecorder* > enumerableProbeRecorders)
void	*Update* ( )

Breakdown

- void **AddProbeRecorders** (IEnumerable< *ProbeRecorder* > enumerableProbeRecorders)
- void **Update** ( )

4.2.24 IVisualObjectFactory

Namespace: WellFired.Profile.ProfileProcessor.Visual

Description

Public Methods

IRectangle	*GetRectangle* ( )
IText	*GetText* ( )

Breakdown

- *IRectangle* **GetRectangle** ( )
- *IText* **GetText** ( )

4.2.25 IGameEventSender

Namespace: WellFired.Profile

Description

Events

Action	*OnGameStart*
Action	*OnGameEnd*

Breakdown

- Action **OnGameStart**
- Action **OnGameEnd**

4.2.26 IGameEventSenderCreator

Namespace: WellFired.Profile

Description

Public Methods

IGameEventSender	*Create* ( )

Breakdown

- *IGameEventSender* **Create** ( )

4.2.27 ILogger

Namespace: WellFired.Profile

Description

Public Methods

void	*Toggle* (bool enabled)
void	*Log* (string message)
void	*LogWarning* (string message)
void	*LogError* (string message)

Breakdown

- void **Toggle** (bool enabled)
- void **Log** (string message)
- void **LogWarning** (string message)
- void **LogError** (string message)

4.2.28 IRuntimeTaskLooper

Namespace: WellFired.Profile

Description

A test here that will work with a Probe

Public Methods

void	*Play* (Action a, float interval)
void	*Stop* ( )

Breakdown

- void **Play** (Action a, float interval)

  Description

  Allows to repeat an action over time at specified interval.

  Parameters

a
interval

- void **Stop** ( )

4.3 Namespaces

4.3.1 GraphicSettings

Namespace: WellFired.Profile

Description

Breakdown

4.3.2 Shadow

Namespace: WellFired.Profile.Data

Description

Breakdown

4.3.3 Texture

Namespace: WellFired.Profile.Data

Description

Breakdown

4.3.4 Probes

Namespace: WellFired

Description

Breakdown

4.3.5 ProfileProcessor

Namespace: WellFired

Description

Breakdown

4.4 Enums

4.4.1 BlendWeights

Namespace: *WellFired.Profile.Data.GraphicSettings*

Description

Internal representation of Unity's Blend Weights settings

OneBone
TwoBones
FourBones
NoSync
EveryVBlank
EverySecondVBlank

4.4.2 Projection

Namespace: *WellFired.Profile.Data.GraphicSettings.Shadow*

Description

Internal representation of Unity's *Shadow* Projection settings

CloseFit
StableFit
Disable
HardOnly
All
Low
Medium
High
VeryHigh

4.4.3 AnisotropicFiltering

Namespace: *WellFired.Profile.Data.GraphicSettings.Texture*

Description

Internal representation of Unity's *Texture* Filetering options

Disable
Enable
ForceEnable
FullRes
HalfRes
QuarterRes
EighthRes

4.4.4 RecordMode

Namespace: *WellFired.Profile.Probes*

Description

The record mode for a given Probe

OneShot	This probe mode will grab data only once and it won't be updated
Continous	This probe mode will continually grab data from it's probe.

4.4.5 Protocol

Namespace: *WellFired.Profile.ProfileProcessor*

Description

TCP
UDP