

# Namespace Awperative

## Classes

### [Awperative](#)

Initiating class of Awperative. Call Start() to start the kernel.

### [Base](#)

Base class of Awperative. Carries events from MonoGame into scenes and hooks.

### [Body](#)

#### [BodyComponent](#)

#### [BodyCreateEvent](#)

#### [BodyDestroyEvent](#)

### [Component](#)

The lowest level scripting class in Awperative. Components are scene level and provide access to all scene level methods, can be applied to any docker and inherited Sadly component does not have excessive access to specific types. Anything that inherits Component is built to work in any DockerEntity, which leads to generic Assumptions. If you want to make a body specific or scene specific component both classes are available.

#### [ComponentCreateEvent](#)

#### [ComponentDestroyEvent](#)

### [Debug](#)

### [DockerEntity](#)

Base class for all Awperative entities, manages components as a requirement because that is the job of all entities.

### [Scene](#)

#### [SceneComponent](#)

#### [SceneCreateEvent](#)

#### [SceneDestroyEvent](#)

### [Transform](#)

#### [TransformModifyEvent](#)

# Interfaces

## [AwperativeHook](#)

Awperative hooks are the source of entry for scripts using Awperative. Create a hook and send into Start() to be recognized by the engine.

# Class Awperative

Namespace: [Awperative](#)

Assembly: Awperative.dll

Initiating class of Awperative. Call Start() to start the kernel.

```
public static class Awperative
```

## Inheritance

[object](#) ← Awperative

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## Base

Bottom class of Awperative. Contains the MonoGame instance.

```
public static Base Base { get; }
```

## Property Value

[Base](#)

## ContentManager

Handles loading content through MonoGame.

```
public static ContentManager ContentManager { get; }
```

## Property Value

## EventHooks

List of all event hooks currently loaded in the kernel.

```
public static List<AwperativeHook> EventHooks { get; }
```

### Property Value

[List](#)<[AwperativeHook](#)>

## GraphicsDeviceManager

Handles graphics settings through MonoGame.

```
public static GraphicsDeviceManager GraphicsDeviceManager { get; }
```

### Property Value

GraphicsDeviceManager

## LoadedScenes

List of all scenes currently loaded in the kernel.

```
public static List<Scene> LoadedScenes { get; }
```

### Property Value

[List](#)<[Scene](#)>

## SpriteBatch

Handles drawing sprites to the screen through MonoGame.

```
public static SpriteBatch SpriteBatch { get; }
```

## Property Value

SpriteBatch

## Methods

### Start(List<AwperativeHook>)

Start() begins the game; and begins communication with all event hooks.

```
public static void Start(List<AwperativeHook> __hooks)
```

#### Parameters

\_\_hooks [List](#)<AwperativeHook>

List of all event hooks you wish to use.

#### Remarks

You cannot add new hooks later; so make sure to register all of them in the Start() method.

# Interface AwperativeHook

Namespace: [Awperative](#)

Assembly: Awperative.dll

Awperative hooks are the source of entry for scripts using Awperative. Create a hook and send into Start() to be recognized by the engine.

```
public interface AwperativeHook
```

## Methods

### Load()

Called when the program starts; It is not recommended you load assets here.

```
void Load()
```

### Unload()

Called when the program closes.

```
void Unload()
```

# Class Base

Namespace: [Awperative](#)

Assembly: Awperative.dll

Base class of Awperative. Carries events from MonoGame into scenes and hooks.

```
public sealed class Base : Game, IDisposable
```

## Inheritance

[object](#) ← Game ← Base

## Implements

[IDisposable](#)

## Inherited Members

Game.Dispose() , Game.Exit() , Game.ResetElapsedTime() , Game.SuppressDraw() ,  
Game.RunOneFrame() , Game.Run() , Game.Run(GameRunBehavior) , Game.Tick()  
Game.LaunchParameters , Game.Components , Game.InactiveSleepTime , Game.MaxElapsedTime ,  
Game.IsActive , Game.IsMouseVisible , Game.TargetElapsedTime , Game.IsFixedTimeStep ,  
Game.Services , Game.Content , Game.GraphicsDevice , Game.Window , Game.Activated ,  
Game.Deactivated , Game.Disposed , Game.Exiting , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Methods

## Draw(GameTime)

Draw() is called every frame; after Update(). Override Draw() in scripting tools to call from this event.

```
protected override void Draw(GameTime __gameTime)
```

## Parameters

[\\_\\_gameTime](#) GameTime

## Remarks

Hooks are unable to receive both Update() and Draw()

## EndRun()

EndRun() is called if the program closes. Override Terminate() in scripting tools or use hooks to call from this event.

```
protected override void EndRun()
```

### Remarks

This event may not trigger if the program is force closed.

## Initialize()

Initialize() is called when the program starts. Goes before LoadContent(). And prepares the kernel for use.

```
protected override void Initialize()
```

### Remarks

It is recommended not to load content in Initialize()

## LoadContent()

LoadContent() is called when the program starts; right after Initialize(). Override Load() in scripting tools or use hooks to call from this event.

```
protected override void LoadContent()
```

### Remarks

It is recommended to load content during LoadContent()

## Update(GameTime)

Update() is called every frame; before Draw(). Override Update() in scripting tools to call from this event.

```
protected override void Update(GameTime __gameTime)
```

## Parameters

\_\_gameTime GameTime

## Remarks

Hooks are unable to receive both Update() and Draw()

# Class Body

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed class Body : DockerEntity
```

## Inheritance

[object](#) ← [DockerEntity](#) ← Body

## Inherited Members

[DockerEntity.AddComponent<Generic>\(\)](#) , [DockerEntity.AddComponent<Generic>\(object\[\]\)](#) ,  
[DockerEntity.GetComponent<Generic>\(\)](#) , [DockerEntity.GetComponents<Generic>\(\)](#) ,  
[DockerEntity.RemoveComponent\(Component\)](#) , [DockerEntity.RemoveComponent<Generic>\(\)](#) ,  
[DockerEntity.RemoveComponents<Generic>\(\)](#) , [object.Equals\(object\)](#) ,  
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,  
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Properties

## Components

All components attached to the body

```
public List<Component> Components { get; }
```

## Property Value

[List](#)<[Component](#)>

## Scene

Current scene the body exists in

```
public Scene Scene { get; }
```

Property Value

[Scene](#)

Tags

All tags attached to the body

```
public List<string> Tags { get; }
```

Property Value

[List<string>](#)

transform

Position of the body

```
public Transform transform { get; }
```

Property Value

[Transform](#)

# Class BodyComponent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public class BodyComponent : Component
```

## Inheritance

[object](#) ← [Component](#) ← BodyComponent

## Inherited Members

[Component.Scene](#) , [Component.Unload\(\)](#) , [Component.Load\(\)](#) , [Component.Update\(GameTime\)](#) ,  
[Component.Draw\(GameTime\)](#) , [Component.Create\(\)](#) , [Component.Destroy\(\)](#) , [Component.AddBody\(\)](#) ,  
[Component.AddBody\(Transform\)](#) , [Component.GetBody\(string\)](#) , [Component.GetBodies\(string\)](#) ,  
[Component.DestroyBody\(Body\)](#) , [Component.AddComponent<Generic>\(\)](#) ,  
[Component.AddComponent<Generic>\(object\[\]\)](#) , [Component.GetComponent<Generic>\(\)](#) ,  
[Component.GetComponents<Generic>\(\)](#) , [Component.RemoveComponent<Generic>\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### Body

```
public Body Body
```

## Field Value

[Body](#)

## Properties

### Transform

```
public Transform Transform { get; }
```

Property Value

[Transform](#)

# Class BodyCreateEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record BodyCreateEvent : IEquatable<BodyCreateEvent>
```

## Inheritance

[object](#) ← BodyCreateEvent

## Implements

[IEquatable](#)<[BodyCreateEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Fields

### body

```
public readonly Body body
```

### Field Value

[Body](#)

### scene

```
public readonly Scene scene
```

### Field Value

[Scene](#)

# Class BodyDestroyEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record BodyDestroyEvent : IEquatable<BodyDestroyEvent>
```

## Inheritance

[object](#) ← BodyDestroyEvent

## Implements

[IEquatable](#)<[BodyDestroyEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Fields

### body

```
public readonly Body body
```

### Field Value

[Body](#)

### scene

```
public readonly Scene scene
```

### Field Value

[Scene](#)

# Class Component

Namespace: [Awperative](#)

Assembly: Awperative.dll

The lowest level scripting class in Awperative. Components are scene level and provide access to all scene level methods, can be applied to any docker and inherited Sadly component does not have excessive access to specific types. Anything that inherits Component is built to work in any DockerEntity, which leads to generic Assumptions. If you want to make a body specific or scene specific component both classes are available.

```
public abstract class Component
```

## Inheritance

[object](#) ← Component

## Derived

[BodyComponent](#), [SceneComponent](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## Scene

```
public Scene Scene { get; set; }
```

## Property Value

[Scene](#)

# Methods

## AddBody()

```
protected Body AddBody()
```

### Returns

[Body](#)

## AddBody(Transform)

```
protected Body AddBody(Transform __transform)
```

### Parameters

\_\_transform [Transform](#)

### Returns

[Body](#)

## AddComponent<Generic>()

```
public Component AddComponent<Generic>() where Generic : Component
```

### Returns

[Component](#)

### Type Parameters

[Generic](#)

## AddComponent<Generic>(object[])

```
public Component AddComponent<Generic>(object[] __args) where Generic : Component
```

Parameters

\_\_args object[]

Returns

Component

Type Parameters

Generic

Create()

```
public virtual void Create()
```

Destroy()

```
public virtual void Destroy()
```

DestroyBody(Body)

```
protected void DestroyBody(Body __body)
```

Parameters

\_\_body Body

Draw(GameTime)

```
public virtual void Draw(GameTime __gameTime)
```

Parameters

\_\_gameTime GameTime

## GetBodies(string)

protected Body[] GetBodies(string \_\_tag)

### Parameters

\_\_tag string ↗

### Returns

[Body](#)[]

## GetBody(string)

protected Body GetBody(string \_\_tag)

### Parameters

\_\_tag string ↗

### Returns

[Body](#)

## GetComponent<Generic>()

public Component GetComponent<Generic>() where Generic : Component

### Returns

[Component](#)

### Type Parameters

Generic

## GetComponents<Generic>()

```
public Component[] GetComponents<Generic>() where Generic : Component
```

Returns

[Component\[\]](#)

Type Parameters

Generic

## Load()

```
public virtual void Load()
```

## RemoveComponent<Generic>()

```
public void RemoveComponent<Generic>() where Generic : Component
```

Type Parameters

Generic

## Unload()

```
public virtual void Unload()
```

## Update(GameTime)

```
public virtual void Update(GameTime __gameTime)
```

## Parameters

\_\_gameTime GameTime

# Class ComponentCreateEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record ComponentCreateEvent : IEquatable<ComponentCreateEvent>
```

## Inheritance

[object](#) ← ComponentCreateEvent

## Implements

[IEquatable](#)<[ComponentCreateEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Fields

### body

```
public readonly Body body
```

### Field Value

[Body](#)

### component

```
public readonly Component component
```

### Field Value

[Component](#)

## scene

```
public readonly Scene scene
```

Field Value

[Scene](#)

# Class ComponentDestroyEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record ComponentDestroyEvent : IEquatable<ComponentDestroyEvent>
```

## Inheritance

[object](#) ← ComponentDestroyEvent

## Implements

[IEquatable](#)<[ComponentDestroyEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Fields

### body

```
public readonly Body body
```

#### Field Value

[Body](#)

### component

```
public readonly Component component
```

#### Field Value

[Component](#)

## scene

```
public readonly Scene scene
```

Field Value

[Scene](#)

# Class Debug

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public static class Debug
```

## Inheritance

[object](#) ← Debug

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Properties

## LogFilePath

True path of the log file Awperative dumps to.

```
public static string LogFilePath { get; }
```

## Property Value

[string](#)

# Methods

## AssertAction(bool, string)

Writes the current message to the log file if the condition is true.

```
public static void AssertAction(bool __condition, string __message)
```

## Parameters

## \_\_condition [bool](#)

Condition to debug

## \_\_message [string](#)

Message to debug

# AssertError(bool, string)

Writes the current message to the log file if the condition is true.

```
public static void AssertError(bool __condition, string __message)
```

## Parameters

### \_\_condition [bool](#)

Condition to debug

### \_\_message [string](#)

Message to debug

# AssertGeneric(bool, string, string[], string[], string[])

Writes the current message to the log file if the condition is true. With any given call sign.

```
public static void AssertGeneric(bool __condition, string __message, string __callSign, string[] __parameters, string[] __values)
```

## Parameters

### \_\_condition [bool](#)

Condition to debug

### \_\_message [string](#)

Message to debug

\_\_callSign [string](#)

Message identifier

\_\_parameters [string](#)[]

Names of values to debug

\_\_values [string](#)[]

Values to debug

## AssertState(bool, string)

Writes the current message to the log file if the condition is true.

```
public static void AssertState(bool __condition, string __message)
```

### Parameters

\_\_condition [bool](#)

Condition to debug

\_\_message [string](#)

Message to debug

## AssertValue(bool, string)

Writes the current message to the log file if the condition is true.

```
public static void AssertValue(bool __condition, string __message)
```

### Parameters

\_\_condition [bool](#)

Condition to debug

\_\_message [string](#)

Message to debug

## AssertWarning(bool, string)

Writes the current message to the log file if the condition is true.

```
public static void AssertWarning(bool __condition, string __message)
```

### Parameters

\_\_condition [bool](#)

Condition to debug

\_\_message [string](#)

Message to debug

## LogAction(string)

Writes the current message to the log file.

```
public static void LogAction(string __message)
```

### Parameters

\_\_message [string](#)

Message to debug

## LogAction(string, string[], string[])

Writes the current message to the log file. With any given call sign.

```
public static void LogAction(string __message, string[] __parameters,  
string[] __values)
```

## Parameters

\_\_message [string](#)

Message to debug

\_\_parameters [string](#)[]

Names of values to debug

\_\_values [string](#)[]

Values to debug

## .LogError(string)

Writes the current message to the log file.

```
public static void LogError(string __message)
```

## Parameters

\_\_message [string](#)

Message to debug

## .LogError(string, string[], string[])

Writes the current message to the log file. With any given call sign.

```
public static void LogError(string __message, string[] __parameters,
string[] __values)
```

## Parameters

\_\_message [string](#)

Message to debug

\_\_parameters [string](#)[]

Names of values to debug

\_\_values [string\[\]](#) []

Values to debug

## LogGeneric(string, string, string[], string[])

Writes the current message to the log file. With any given call sign.

```
public static void LogGeneric(string __message, string __callSign, string[]
__parameters, string[] __values)
```

### Parameters

\_\_message [string\[\]](#)

Message to debug

\_\_callSign [string\[\]](#)

Message identifier

\_\_parameters [string\[\]](#) []

Names of values to debug

\_\_values [string\[\]](#) []

Values to debug

## LogState(string)

Writes the current message to the log file.

```
public static void LogState(string __message)
```

### Parameters

\_\_message [string\[\]](#)

Message to debug

## LogState(string, string[], string[])

Writes the current message to the log file. With any given call sign.

```
public static void LogState(string __message, string[] __parameters,  
string[] __values)
```

### Parameters

\_\_message [string](#)

Message to debug

\_\_parameters [string](#)[]

Names of values to debug

\_\_values [string](#)[]

Values to debug

## LogValue(string)

Writes the current message to the log file.

```
public static void LogValue(string __message)
```

### Parameters

\_\_message [string](#)

Message to debug

## LogValue(string, string[], string[])

Writes the current message to the log file. With any given call sign.

```
public static void LogValue(string __message, string[] __parameters,  
string[] __values)
```

## Parameters

\_\_message [string](#)

Message to debug

\_\_parameters [string](#)[]

Names of values to debug

\_\_values [string](#)[]

Values to debug

## LogWarning(string)

Writes the current message to the log file.

```
public static void LogWarning(string __message)
```

## Parameters

\_\_message [string](#)

Message to debug

## LogWarning(string, string[], string[])

Writes the current message to the log file. With any given call sign.

```
public static void LogWarning(string __message, string[] __parameters,  
string[] __values)
```

## Parameters

\_\_message [string](#)

Message to debug

parameters [string](#)[]

Names of values to debug

values [string](#)[]

Values to debug

# Class DockerEntity

Namespace: [Awperative](#)

Assembly: Awperative.dll

Base class for all Awperative entities, manages components as a requirement because that is the job of all entities.

```
public abstract class DockerEntity
```

## Inheritance

[object](#) ← DockerEntity

## Derived

[Body](#), [Scene](#)

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### Scene

```
public Scene Scene
```

### Field Value

[Scene](#)

## Methods

### AddComponent<Generic>()

```
public Component AddComponent<Generic>() where Generic : Component
```

Returns

[Component](#)

Type Parameters

**Generic**

AddComponent<Generic>(object[])

```
public Component AddComponent<Generic>(object[] __args) where Generic : Component
```

Parameters

**\_\_args** [object](#)[]

Returns

[Component](#)

Type Parameters

**Generic**

GetComponent<Generic>()

```
public Component GetComponent<Generic>() where Generic : Component
```

Returns

[Component](#)

Type Parameters

**Generic**

## GetComponents<Generic>()

```
public Component[] GetComponents<Generic>() where Generic : Component
```

Returns

[Component\[\]](#)

Type Parameters

Generic

## RemoveComponent(Component)

```
public void RemoveComponent(Component __component)
```

Parameters

\_\_component [Component](#)

## RemoveComponent<Generic>()

```
public void RemoveComponent<Generic>() where Generic : Component
```

Type Parameters

Generic

## RemoveComponents<Generic>()

```
public void RemoveComponents<Generic>() where Generic : Component
```

Type Parameters

## Generic

# Class Scene

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed class Scene : DockerEntity
```

## Inheritance

[object](#) ← [DockerEntity](#) ← Scene

## Inherited Members

[DockerEntity.AddComponent<Generic>\(\)](#) , [DockerEntity.AddComponent<Generic>\(object\[\]\)](#) ,  
[DockerEntity.Scene](#) , [DockerEntity.GetComponent<Generic>\(\)](#) ,  
[DockerEntity.GetComponents<Generic>\(\)](#) , [DockerEntity.RemoveComponent\(Component\)](#) ,  
[DockerEntity.RemoveComponent<Generic>\(\)](#) , [DockerEntity.RemoveComponents<Generic>\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Properties

## bodies

```
public List<Body> bodies { get; }
```

## Property Value

[List](#)<[Body](#)>

# Methods

## AddBody()

```
public Body AddBody()
```

## Returns

## Body

### AddBody(Transform)

```
public Body AddBody(Transform __transform)
```

#### Parameters

\_\_transform [Transform](#)

#### Returns

## Body

### DestroyBody(Body)

```
public void DestroyBody(Body __body)
```

#### Parameters

\_\_body [Body](#)

### Draw(GameTime)

```
public void Draw(GameTime __gameTime)
```

#### Parameters

\_\_gameTime [GameTime](#)

### GetBodies(string)

```
public Body[] GetBodies(string tag)
```

Parameters

**tag** [string](#)

Returns

[Body](#)[]

## GetBody(string)

```
public Body GetBody(string tag)
```

Parameters

**tag** [string](#)

Returns

[Body](#)

## Load()

```
public void Load()
```

## Unload()

```
public void Unload()
```

## Update(GameTime)

```
public void Update(GameTime __gameTime)
```

Parameters

\_\_gameTime GameTime

## Events

### BodyCreatedEvent

```
public event EventHandler<BodyCreateEvent> BodyCreatedEvent
```

Event Type

[EventHandler](#)<[BodyCreateEvent](#)>

### BodyDestroyedEvent

```
public event EventHandler<BodyDestroyEvent> BodyDestroyedEvent
```

Event Type

[EventHandler](#)<[BodyDestroyEvent](#)>

# Class SceneComponent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public class SceneComponent : Component
```

## Inheritance

[object](#) ← [Component](#) ← SceneComponent

## Inherited Members

[Component.Scene](#) , [Component.Unload\(\)](#) , [Component.Load\(\)](#) , [Component.Update\(GameTime\)](#) ,  
[Component.Draw\(GameTime\)](#) , [Component.Create\(\)](#) , [Component.Destroy\(\)](#) , [Component.AddBody\(\)](#) ,  
[Component.AddBody\(Transform\)](#) , [Component.GetBody\(string\)](#) , [Component.GetBodies\(string\)](#) ,  
[Component.DestroyBody\(Body\)](#) , [Component.AddComponent<Generic>\(\)](#) ,  
[Component.AddComponent<Generic>\(object\[\]\)](#) , [Component.GetComponent<Generic>\(\)](#) ,  
[Component.GetComponents<Generic>\(\)](#) , [Component.RemoveComponent<Generic>\(\)](#) ,  
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

# Class SceneCreateEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record SceneCreateEvent : IEquatable<SceneCreateEvent>
```

## Inheritance

[object](#) ← SceneCreateEvent

## Implements

[IEquatable](#)<[SceneCreateEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

# Fields

## scene

```
public Scene scene
```

## Field Value

[Scene](#)

# Class SceneDestroyEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record SceneDestroyEvent : IEquatable<SceneDestroyEvent>
```

## Inheritance

[object](#) ← SceneDestroyEvent

## Implements

[IEquatable](#)<[SceneDestroyEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Fields

### scene

```
public Scene scene
```

### Field Value

[Scene](#)

# Class Transform

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed class Transform
```

## Inheritance

[object](#) ← Transform

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Constructors

Transform()

```
public Transform()
```

Transform(Vector2, Vector2, float, float, Vector2)

```
public Transform(Vector2 __origin, Vector2 __position, float __depth, float  
__rotation, Vector2 __scale)
```

## Parameters

\_\_origin Vector2

\_\_position Vector2

\_\_depth float

\_\_rotation float

\_\_scale Vector2

# Properties

## Depth

```
public float Depth { get; set; }
```

### Property Value

[float](#) ↗

## Origin

```
public Vector2 Origin { get; set; }
```

### Property Value

Vector2

## Position

```
public Vector2 Position { get; set; }
```

### Property Value

Vector2

## Rotation

```
public float Rotation { get; set; }
```

### Property Value

[float](#) ↗

## Scale

```
public Vector2 Scale { get; set; }
```

### Property Value

Vector2

## Methods

### Clone()

```
public Transform Clone()
```

### Returns

[Transform](#)

### Set(Vector2, Vector2, float, float, Vector2)

```
public void Set(Vector2 __origin, Vector2 __position, float __depth, float  
__rotation, Vector2 __scale)
```

### Parameters

\_\_origin Vector2

\_\_position Vector2

\_\_depth [float](#)

\_\_rotation [float](#)

\_\_scale Vector2

### ToMatrix()

```
public Matrix ToMatrix()
```

Returns

Matrix

## Events

### OnTransformChangedEvent

```
public event EventHandler<TransformModifyEvent> OnTransformChangedEvent
```

Event Type

[EventHandler](#) <[TransformModifyEvent](#)>

# Class TransformModifyEvent

Namespace: [Awperative](#)

Assembly: Awperative.dll

```
public sealed record TransformModifyEvent : IEquatable<TransformModifyEvent>
```

## Inheritance

[object](#) ← TransformModifyEvent

## Implements

[IEquatable](#)<[TransformModifyEvent](#)>

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

## Fields

### after

```
public readonly Transform after
```

### Field Value

[Transform](#)

### before

```
public readonly Transform before
```

### Field Value

[Transform](#)

# Namespace Awperative.Kernel.Communication.Config

## Classes

[Config](#)

# Class Config

Namespace: [Awperative.Kernel.Communication.Config](#)

Assembly: Awperative.dll

```
public static class Config
```

## Inheritance

[object](#) ← Config

## Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) ,  
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) ,  
[object.ToString\(\)](#)

## Fields

### logFileName

```
public static string logFileName
```

## Field Value

[string](#)