

Namespace S1API

Namespaces

[S1API.AssetBundles](#)

[S1API.Avatar](#)

[S1API.Building](#)

[S1API.Carte](#)

[S1API.Casino](#)

[S1API.Conditions](#)

[S1API.Console](#)

[S1API.DeadDrops](#)

[S1API.Dialogues](#)

[S1API.Economy](#)

[S1API.Entities](#)

[S1API.GameTime](#)

[S1API.Graffiti](#)

[S1API.Growing](#)

[S1API.Input](#)

[S1API.Internal](#)

[S1API.Items](#)

[S1API.Law](#)

[S1API.Leveling](#)

[S1API.Lifecycle](#)

[S1API.Logging](#)

[S1API.Map](#)

[S1API.Messaging](#)

[S1API.Money](#)

[S1API.PhoneApp](#)

[S1API.PhoneCalls](#)

[S1API.Products](#)

[S1API.Properties](#)

[S1API.Property](#)

[S1API.Quests](#)

[S1API.Rendering](#)

[S1API.Saveables](#)

[S1API.Shops](#)

[S1API.Storage](#)

[S1API.Storages](#)

[S1API.UI](#)

[S1API_Utils](#)

[S1API.Vehicles](#)

Classes

[MyPluginInfo](#)

[S1API](#)

S1API root MelonMod. Provides lifecycle hooks for internal systems.

[VersionChecker](#)

Version checking functionality for MelonLoader compatibility. Credits: estonia__ and k073l (S1 Modding Discord)

Namespace S1API.AssetBundles

Classes

[AssetLoader](#)

The asset bundle manager

[WrappedAssetBundle](#)

INTERNAL: Wrapper around UnityEngine.AssetBundle instance.

[WrappedAssetBundleRequest](#)

INTERNAL: Wrapper around UnityEngine.AssetBundleRequest instance.

Class AssetLoader

Namespace: [S1API.AssetBundles](#)

Assembly: S1API.dll

The asset bundle manager

```
public static class AssetLoader
```

Inheritance

[object](#) ← AssetLoader

Inherited Members

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

Methods

EasyLoad<T>(string, string)

Loads an asset of type [T](#) from an embedded AssetBundle using the executing assembly.

```
public static T EasyLoad<T>(string bundleName, string objectName) where T : Object
```

Parameters

bundleName [string](#)

The name of the embedded AssetBundle resource.

objectName [string](#)

The name of the asset to load within the AssetBundle.

Returns

T

The loaded asset of type `T`.

Type Parameters

T

The type of asset to load (must derive from `UnityEngine.Object`).

EasyLoad<T>(string, string, out WrappedAssetBundle)

Loads an asset of type `T` from an embedded AssetBundle using the executing assembly and outputs the loaded bundle.

```
public static T EasyLoad<T>(string bundleName, string objectName, out  
WrappedAssetBundle bundle) where T : Object
```

Parameters

`bundleName` `string`

The name of the embedded AssetBundle resource.

`objectName` `string`

The name of the asset to load within the AssetBundle.

`bundle` `WrappedAssetBundle`

The output parameter containing the loaded `WrappedAssetBundle`.

Returns

T

The loaded asset of type `T`.

Type Parameters

T

The type of asset to load (must derive from UnityEngine.Object).

EasyLoad<T>(string, string, Assembly)

Loads an asset of type T from an embedded AssetBundle using a specified assembly.

```
public static T EasyLoad<T>(string bundleName, string objectName, Assembly assemblyOverride) where T : Object
```

Parameters

bundleName [string](#)

The name of the embedded AssetBundle resource.

objectName [string](#)

The name of the asset to load within the AssetBundle.

assemblyOverride [Assembly](#)

The assembly from which to load the embedded AssetBundle resource.

Returns

T

The loaded asset of type T.

Type Parameters

T

The type of asset to load (must derive from UnityEngine.Object).

EasyLoad<T>(string, string, Assembly, out WrappedAssetBundle)

Loads an asset of type `T` from an embedded AssetBundle using a specified assembly and outputs the loaded bundle.

```
public static T EasyLoad<T>(string bundleName, string objectName, Assembly assemblyOverride, out WrappedAssetBundle bundle) where T : Object
```

Parameters

`bundleName` [string](#)

The name of the embedded AssetBundle resource.

`objectName` [string](#)

The name of the asset to load within the AssetBundle.

`assemblyOverride` [Assembly](#)

The assembly from which to load the embedded AssetBundle resource.

`bundle` [WrappedAssetBundle](#)

The output parameter containing the loaded [WrappedAssetBundle](#).

Returns

`T`

The loaded asset of type `T`.

Type Parameters

`T`

The type of asset to load (must derive from `UnityEngine.Object`).

GetAssetBundleFromStream(string, Assembly)

Load a [WrappedAssetBundle](#) instance by [string](#) resource name.

```
public static WrappedAssetBundle GetAssetBundleFromStream(string full resourceName,  
Assembly overrideAssembly)
```

Parameters

fullResourceName [string](#)

The full embedded resource name (including namespace path);

overrideAssembly [Assembly](#)

The assembly to load the embedded resource from.

Returns

[WrappedAssetBundle](#)

The loaded AssetBundle instance

Class WrappedAssetBundle

Namespace: [S1API.AssetBundles](#)

Assembly: S1API.dll

INTERNAL: Wrapper around UnityEngine.AssetBundle instance.

```
public class WrappedAssetBundle
```

Inheritance

[object](#) ← WrappedAssetBundle

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WrappedAssetBundle(AssetBundle)

Initializes a new instance of the [WrappedAssetBundle](#) class.

```
public WrappedAssetBundle(AssetBundle realBundle)
```

Parameters

realBundle AssetBundle

The actual Unity AssetBundle to wrap.

Properties

IsStreamedAssetBundle

Gets a value indicating whether this asset bundle is a streamed scene asset bundle.

```
public bool IsStreamedAssetBundle { get; }
```

Property Value

[bool](#)

Methods

Contains(string)

Checks if the asset bundle contains a specific asset by name.

```
public bool Contains(string name)
```

Parameters

[name](#) [string](#)

Returns

[bool](#)

GetAllAssetNames()

Returns all asset names contained in the asset bundle.

```
public string[] GetAllAssetNames()
```

Returns

[string](#) []

GetAllScenePaths()

Returns all scene paths contained in the asset bundle.

```
public string[] GetAllScenePaths()
```

Returns

[string](#) []

Load(string)

Loads an asset by name as a generic UnityEngine.Object.

```
public Object Load(string name)
```

Parameters

name [string](#)

Returns

Object

Load(string, Type)

Loads an asset by name using a [Type](#) object.

```
public Object Load(string name, Type type)
```

Parameters

name [string](#)

type [Type](#)

Returns

Object

LoadAll()

Loads all assets from the bundle as UnityEngine.Object instances.

```
public Object[] LoadAll()
```

Returns

Object[]

LoadAllAssets()

Loads all assets from the bundle as UnityEngine.Object instances.

```
public Object[] LoadAllAssets()
```

Returns

Object[]

LoadAllAssets(Type)

Loads all assets from the bundle using a [Type](#) object.

```
public Object[] LoadAllAssets(Type type)
```

Parameters

type [Type](#)

Returns

Object[]

LoadAllAssets<T>()

Loads all assets from the bundle and casts them to the specified type.

```
public T[] LoadAllAssets<T>() where T : Object
```

Returns

T[]

Type Parameters

T

LoadAsset(string)

Loads an asset by name as a generic UnityEngine.Object.

```
public Object LoadAsset(string name)
```

Parameters

name [string](#)

Returns

Object

LoadAsset(string, Type)

Loads an asset by name using a [Type](#) object.

```
public Object LoadAsset(string name, Type type)
```

Parameters

name [string](#)

type [Type](#)

Returns

Object

LoadAssetAsync(string)

Asynchronously loads an asset by name as a generic UnityEngine.Object.

```
public WrappedAssetBundleRequest LoadAssetAsync(string name)
```

Parameters

name [string](#)

Returns

[WrappedAssetBundleRequest](#)

LoadAssetAsync(string, Type)

Asynchronously loads an asset by name using a [Type](#) object.

```
public WrappedAssetBundleRequest LoadAssetAsync(string name, Type type)
```

Parameters

name [string](#)

type [Type](#)

Returns

[WrappedAssetBundleRequest](#)

LoadAssetAsync<T>(string)

Asynchronously loads an asset by name and casts it to the specified type.

```
public WrappedAssetBundleRequest LoadAssetAsync<T>(string name) where T : Object
```

Parameters

name [string](#)

Returns

[WrappedAssetBundleRequest](#)

Type Parameters

T

LoadAssetWithSubAssets(string)

Loads an asset and all of its sub-assets by name as UnityEngine.Object instances.

```
public Object[] LoadAssetWithSubAssets(string name)
```

Parameters

name [string](#)

Returns

[Object\[\]](#)

LoadAssetWithSubAssets(string, Type)

Loads an asset and its sub-assets by name using a [Type](#) object.

```
public Object[] LoadAssetWithSubAssets(string name, Type type)
```

Parameters

name [string](#)

type [Type](#)

Returns

[Object\[\]](#)

LoadAssetWithSubAssets<T>(string)

Loads an asset and its sub-assets by name and casts them to the specified type.

```
public T[] LoadAssetWithSubAssets<T>(string name) where T : Object
```

Parameters

name [string](#)

Returns

T[]

Type Parameters

T

LoadAsset<T>(string)

Loads an asset by name and casts it to the specified type.

```
public T LoadAsset<T>(string name) where T : Object
```

Parameters

name [string](#)

Returns

T

Type Parameters

T

Load<T>(string)

Loads an asset by name and casts it to the specified type.

```
public T Load<T>(string name) where T : Object
```

Parameters

name [string](#)

Returns

T

Type Parameters

T

Unload(bool)

Unloads the asset bundle and optionally unloads all loaded objects.

```
public void Unload(bool unloadAllLoadedObjects)
```

Parameters

unloadAllLoadedObjects [bool](#)

Whether to unload all loaded objects as well.

Class WrappedAssetBundleRequest

Namespace: [S1API.AssetBundles](#)

Assembly: S1API.dll

INTERNAL: Wrapper around UnityEngine.AssetBundleRequest instance.

```
public class WrappedAssetBundleRequest
```

Inheritance

[object](#) ← WrappedAssetBundleRequest

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

AllAssets

All Assets in the UnityEngine.AssetBundleRequest.

```
public Object[] AllAssets { get; }
```

Property Value

Object[]

Asset

The requested UnityEngine.Object asset instance.

```
public Object Asset { get; }
```

Property Value

Object

Namespace S1API.Avatar

Classes

[Avatar](#)

Modder-facing wrapper for an Avatar component in the game. Provides access to avatar appearance and state without exposing game types.

[AvatarSettings](#)

Modder-facing wrapper for AvatarSettings ScriptableObject. Provides access to avatar appearance configuration without exposing game types.

[AvatarSettings.AccessorySetting](#)

Accessory setting structure.

[AvatarSettings.EyeLidConfiguration](#)

Eye lid configuration structure.

[AvatarSettings.LayerSetting](#)

Layer setting structure.

[BasicAvatarSettings](#)

Simplified avatar settings format used by the character creator. Provides a more user-friendly interface for basic character customization compared to the full AvatarSettings.

[Seat](#)

Provides metadata about a world `AvatarSeat` for modders. Seats are registered automatically when `AvatarSeat.Awake` runs and cleared on scene unload via `S1API.Internal.Lifecycle.SceneStateCleaner`.

Class Avatar

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Modder-facing wrapper for an Avatar component in the game. Provides access to avatar appearance and state without exposing game types.

```
public sealed class Avatar
```

Inheritance

[object](#) ← Avatar

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

GameObject

The GameObject associated with this avatar.

```
public GameObject GameObject { get; }
```

Property Value

GameObject

IsActive

Whether the avatar GameObject is active.

```
public bool IsActive { get; }
```

Property Value

[bool](#) ↗

Methods

FindInScene(bool)

Finds all Avatar instances in the current scene.

```
public static Avatar[] FindInScene(bool includeInactive = false)
```

Parameters

[includeInactive](#) [bool](#) ↗

Whether to include inactive GameObjects in the search.

Returns

[Avatar](#) ↗

An array of Avatar wrappers found in the scene.

LoadAvatarSettings(AvatarSettings)

Loads avatar settings onto this avatar.

```
public void LoadAvatarSettings(AvatarSettings settings)
```

Parameters

`settings` [AvatarSettings](#)

The avatar settings to load.

Class AvatarSettings

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Modder-facing wrapper for AvatarSettings ScriptableObject. Provides access to avatar appearance configuration without exposing game types.

```
public sealed class AvatarSettings
```

Inheritance

[object](#) ← AvatarSettings

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

AccessoryCount

Accessory settings count.

```
public int AccessoryCount { get; }
```

Property Value

[int](#)

BodyLayerCount

Body layer settings count.

```
public int BodyLayerCount { get; }
```

Property Value

[int ↗](#)

EyeBallTint

Eye ball tint color.

```
public Color EyeBallTint { get; set; }
```

Property Value

Color

EyeballMaterialIdentifier

Eyeball material identifier.

```
public string EyeballMaterialIdentifier { get; set; }
```

Property Value

[string ↗](#)

EyebrowRestingAngle

Eyebrow resting angle value.

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

Property Value

[float](#)

EyebrowRestingHeight

Eyebrow resting height value.

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

Property Value

[float](#)

EyebrowScale

Eyebrow scale value.

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

Property Value

[float](#)

EyebrowThickness

Eyebrow thickness value.

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

Property Value

[float](#)

FaceLayerCount

Face layer settings count.

```
public int FaceLayerCount { get; }
```

Property Value

[int](#)

Gender

Gender value (0.0 to 1.0).

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

Property Value

[float](#)

HairColor

Hair color.

```
public Color HairColor { get; set; }
```

Property Value

Color

HairPath

Hair path/identifier.

```
public string HairPath { get; set; }
```

Property Value

[string](#)

Height

Height value (typically 0.0 to 1.0).

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

Property Value

[float](#)

LeftEyeLidColor

Left eye lid color.

```
public Color32 LeftEyeLidColor { get; set; }
```

Property Value

Color32

LeftEyeRestingState

Left eye resting state configuration.

```
public AvatarSettings.EyeLidConfiguration LeftEyeRestingState { get; set; }
```

Property Value

[AvatarSettings.EyeLidConfiguration](#)

PupilDilation

Pupil dilation value (typically 0.0 to 1.0).

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

Property Value

[float](#)

RightEyeLidColor

Right eye lid color.

```
public Color32 RightEyeLidColor { get; set; }
```

Property Value

Color32

RightEyeRestingState

Right eye resting state configuration.

```
public AvatarSettings.EyeLidConfiguration RightEyeRestingState { get; set; }
```

Property Value

[AvatarSettings.EyeLidConfiguration](#)

SkinColor

Skin color.

```
public Color32 SkinColor { get; set; }
```

Property Value

Color32

Weight

Weight value (typically 0.0 to 1.0).

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

Property Value

[float](#)

Methods

AddAccessory(string, Color)

Adds an accessory setting.

```
public void AddAccessory(string path, Color color)
```

Parameters

path [string](#)

color [Color](#)

AddBodyLayer(string, Color)

Adds a body layer setting.

```
public void AddBodyLayer(string layerPath, Color layerTint)
```

Parameters

layerPath [string](#)

layerTint [Color](#)

AddFaceLayer(string, Color)

Adds a face layer setting.

```
public void AddFaceLayer(string layerPath, Color layerTint)
```

Parameters

layerPath [string](#)

layerTint [Color](#)

Create()

Creates a new AvatarSettings instance.

```
public static AvatarSettings Create()
```

Returns

[AvatarSettings](#)

GetAccessories()

Gets all accessory settings.

```
public List<AvatarSettings.AccessorySetting> GetAccessories()
```

Returns

[List](#)<[AvatarSettings](#).AccessorySetting>

GetBodyLayers()

Gets all body layer settings.

```
public List<AvatarSettings.LayerSetting> GetBodyLayers()
```

Returns

[List](#)<[AvatarSettings](#).LayerSetting>

GetFaceLayers()

Gets all face layer settings.

```
public List<AvatarSettings.LayerSetting> GetFaceLayers()
```

Returns

[List](#)<[AvatarSettings.LayerSetting](#)>

SetAccessories(List<AccessorySetting>)

Sets accessory settings from a list.

```
public void SetAccessories(List<AvatarSettings.AccessorySetting> accessories)
```

Parameters

accessories [List](#)<[AvatarSettings.AccessorySetting](#)>

SetBodyLayers(List<LayerSetting>)

Sets body layer settings from a list.

```
public void SetBodyLayers(List<AvatarSettings.LayerSetting> layers)
```

Parameters

layers [List](#)<[AvatarSettings.LayerSetting](#)>

SetFaceLayers(List<LayerSetting>)

Sets face layer settings from a list.

```
public void SetFaceLayers(List<AvatarSettings.LayerSetting> layers)
```

Parameters

layers [List](#)<[AvatarSettings](#).[LayerSetting](#)>

Class AvatarSettings.AccessorySetting

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Accessory setting structure.

```
public sealed class AvatarSettings.AccessorySetting
```

Inheritance

[object](#) ← AvatarSettings.AccessorySetting

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

AccessorySetting()

```
public AccessorySetting()
```

Properties

Color

Accessory color.

```
public Color Color { get; set; }
```

Property Value

Color

Path

Accessory path/identifier.

```
public string Path { get; set; }
```

Property Value

[string](#)

Class AvatarSettings.EyeLidConfiguration

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Eye lid configuration structure.

```
public sealed class AvatarSettings.EyeLidConfiguration
```

Inheritance

[object](#) ← AvatarSettings.EyeLidConfiguration

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

EyeLidConfiguration()

```
public EyeLidConfiguration()
```

Properties

BottomLidOpen

Bottom lid open value (0.0 to 1.0).

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

Property Value

[float](#)

TopLidOpen

Top lid open value (0.0 to 1.0).

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

Property Value

[float](#)

Class AvatarSettings.LayerSetting

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Layer setting structure.

```
public sealed class AvatarSettings.LayerSetting
```

Inheritance

[object](#) ← AvatarSettings.LayerSetting

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

LayerSetting()

```
public LayerSetting()
```

Properties

LayerPath

Layer path/identifier.

```
public string LayerPath { get; set; }
```

Property Value

[string](#)

LayerTint

Layer tint color.

```
public Color LayerTint { get; set; }
```

Property Value

Color

Class BasicAvatarSettings

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Simplified avatar settings format used by the character creator. Provides a more user-friendly interface for basic character customization compared to the full AvatarSettings.

```
public sealed class BasicAvatarSettings
```

Inheritance

[object](#) ← BasicAvatarSettings

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Bottom

Bottom clothing layer path/identifier.

```
public string Bottom { get; set; }
```

Property Value

[string](#)

BottomColor

Bottom clothing color.

```
public Color BottomColor { get; set; }
```

Property Value

Color

EyeballColor

Eyeball color.

```
public Color EyeballColor { get; set; }
```

Property Value

Color

EyebrowRestingAngle

Eyebrow resting angle value.

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

Property Value

[float](#)

EyebrowRestingHeight

Eyebrow resting height value.

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

PropertyValue

[float](#)

EyebrowScale

Eyebrow scale value.

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

PropertyValue

[float](#)

EyebrowThickness

Eyebrow thickness value.

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

PropertyValue

[float](#)

Eyewear

Eyewear accessory path/identifier.

```
public string Eyewear { get; set; }
```

PropertyValue

[string](#)

EyewearColor

Eyewear color.

```
public Color EyewearColor { get; set; }
```

Property Value

Color

FacialDetails

Facial details layer path/identifier (e.g., scars, makeup).

```
public string FacialDetails { get; set; }
```

Property Value

[string](#)

FacialDetailsIntensity

Facial details intensity (0.0 to 1.0).

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

Property Value

[float](#)

FacialHair

Facial hair layer path/identifier.

```
public string FacialHair { get; set; }
```

Property Value

[string](#)

Gender

Gender value (0 = male, 1 = female).

```
public int Gender { get; set; }
```

Property Value

[int](#)

HairColor

Hair color.

```
public Color HairColor { get; set; }
```

Property Value

Color

HairStyle

Hair style path/identifier.

```
public string Hairstyle { get; set; }
```

Property Value

[string](#)

Headwear

Headwear accessory path/identifier.

```
public string Headwear { get; set; }
```

Property Value

[string](#)

HeadwearColor

Headwear color.

```
public Color HeadwearColor { get; set; }
```

Property Value

Color

LowerEyeLidRestingPosition

Lower eye lid resting position (0.0 to 1.0).

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

Property Value

[float](#)

Mouth

Mouth layer path/identifier.

```
public string Mouth { get; set; }
```

Property Value

[string](#)

PupilDilation

Pupil dilation value (typically 0.0 to 1.0).

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

Property Value

[float](#)

Shoes

Shoes accessory path/identifier.

```
public string Shoes { get; set; }
```

PropertyValue

[string](#)

ShoesColor

Shoes color.

```
public Color ShoesColor { get; set; }
```

PropertyValue

Color

SkinColor

Skin color.

```
public Color SkinColor { get; set; }
```

PropertyValue

Color

Top

Top clothing layer path/identifier.

```
public string Top { get; set; }
```

PropertyValue

[string](#)

TopColor

Top clothing color.

```
public Color TopColor { get; set; }
```

Property Value

Color

UpperEyeLidRestingPosition

Upper eye lid resting position (0.0 to 1.0).

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

Property Value

[float](#)

Weight

Weight value (typically 0.0 to 1.0).

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

Property Value

[float](#)

Methods

AddTattoo(string)

Adds a tattoo layer path to the tattoos list.

```
public void AddTattoo(string tattooPath)
```

Parameters

tattooPath [string](#)

The tattoo layer path to add.

Create()

Creates a new BasicAvatarSettings instance.

```
public static BasicAvatarSettings Create()
```

Returns

[BasicAvatarSettings](#)

GetTattoos()

Gets a list of tattoo layer paths.

```
public List<string> GetTattoos()
```

Returns

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

GetValue<T>(string)

Gets a field value by name using reflection.

```
public T GetValue<T>(string fieldName)
```

Parameters

fieldName [string](#)

The name of the field.

Returns

T

The field value, or default(T) if not found.

Type Parameters

T

The type of the value.

SetTattoos(List<string>)

Sets the tattoo layer paths.

```
public void SetTattoos(List<string> tattoos)
```

Parameters

tattoos [List](#)<[string](#)>

List of tattoo layer paths.

SetValue<T>(string, T)

Sets a field value by name using reflection.

```
public void SetValue<T>(string fieldName, T value)
```

Parameters

fieldName [string](#)

The name of the field.

value [T](#)

The value to set.

Type Parameters

T

The type of the value.

ToAvatarSettings()

Converts this BasicAvatarSettings to a full AvatarSettings instance.

```
public AvatarSettings ToAvatarSettings()
```

Returns

[AvatarSettings](#)

A new AvatarSettings instance with the converted settings.

Class Seat

Namespace: [S1API.Avatar](#)

Assembly: S1API.dll

Provides metadata about a world [AvatarSeat](#) for modders. Seats are registered automatically when [AvatarSeat.Awake](#) runs and cleared on scene unload via [S1API.Internal.Lifecycle.SceneStateCleaner](#).

```
public sealed class Seat
```

Inheritance

[object](#) ← Seat

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

AccessPosition

Access point location used by NPCs to approach the seat, if defined.

```
public Vector3 AccessPosition { get; }
```

Property Value

Vector3

AccessRotation

Access point rotation, or sitting rotation if not defined.

```
public Quaternion AccessRotation { get; }
```

Property Value

Quaternion

Count

Returns the number of registered seats.

```
public static int Count { get; }
```

Property Value

[int](#)

HierarchyPath

Returns the seat's hierarchy path (scene root to GameObject).

```
public string HierarchyPath { get; }
```

Property Value

[string](#)

IndexInSet

Index of the seat inside the parent `AvatarSeatSet.Seats` array, or `null` if not part of a set.

```
public int? IndexInSet { get; }
```

PropertyValue

[int](#)?

Label

Human-readable label combining seat set name (if available) and hierarchy path.

```
public string Label { get; }
```

PropertyValue

[string](#)

SeatSetName

Name of the parent [AvatarSeatSet](#) GameObject, if any.

```
public string SeatSetName { get; }
```

PropertyValue

[string](#)

SittingPosition

World position of the seat's sitting point, or the seat transform position if no sitting point exists.

```
public Vector3 SittingPosition { get; }
```

PropertyValue

SittingRotation

World rotation of the seat's sitting point, or the seat transform rotation if no sitting point exists.

```
public Quaternion SittingRotation { get; }
```

Property Value

Quaternion

Methods

FindByPathSuffix(string)

Finds the first seat whose hierarchy path ends with the provided suffix (case-insensitive). Useful when only a partial path is known (e.g. "Cafe/Booth01/SeatA").

```
public static Seat FindByPathSuffix(string pathSuffix)
```

Parameters

pathSuffix [string](#)

Returns

[Seat](#)

GetAll()

Returns all registered seats (snapshot).

```
public static Seat[] GetAll()
```

Returns

[Seat\[\]](#)

GetBySeatSet(string)

Returns all seats belonging to a named [AvatarSeatSet](#) (case-insensitive).

```
public static Seat[] GetBySeatSet(string setName)
```

Parameters

setName [string](#)

Returns

[Seat\[\]](#)

ResolveGameSeat()

Attempts to retrieve the live game [AvatarSeat](#) component, if it still exists.

```
public AvatarSeat ResolveGameSeat()
```

Returns

[AvatarSeat](#)

ResolveSeatGameObject()

Returns the seat GameObject, if the component still exists.

```
public GameObject ResolveSeatGameObject()
```

Returns

GameObject

ResolveSeatSet()

Attempts to retrieve the live parent [AvatarSeatSet](#), if any.

```
public AvatarSeatSet ResolveSeatSet()
```

Returns

AvatarSeatSet

Namespace S1API.Building

Classes

[BuildEventArgs](#)

Event arguments for building-related events. Provides access to the item being built and the resulting GameObject.

[BuildEvents](#)

Provides events for the building system, allowing mods to customize items as they're placed. Subscribe to these events instead of using Harmony patches on BuildManager.

Class BuildEventArgs

Namespace: [S1API.Building](#)

Assembly: S1API.dll

Event arguments for building-related events. Provides access to the item being built and the resulting GameObject.

```
public class BuildEventArgs
```

Inheritance

[object](#) ← BuildEventArgs

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

GameObject

The GameObject created in the world. Use this to customize appearance, add components, or modify materials.

```
public GameObject GameObject { get; }
```

Property Value

GameObject

ItemId

The item definition ID. Convenience property for filtering events by item type.

```
public string ItemId { get; }
```

Property Value

[string](#)

ItemInstance

The item instance that was built.

```
public ItemInstance ItemInstance { get; }
```

Property Value

[ItemInstance](#)

Storage

The storage entity if this item is a storage container. Returns null if the item is not a storage container.

```
public StorageEntity Storage { get; }
```

Property Value

[StorageEntity](#)

Remarks

Use this to customize storage properties when items are placed. Example: args.Storage?.AddSlots(5);

Class BuildEvents

Namespace: [S1API.Building](#)

Assembly: S1API.dll

Provides events for the building system, allowing mods to customize items as they're placed. Subscribe to these events instead of using Harmony patches on BuildManager.

```
public static class BuildEvents
```

Inheritance

[object](#) ← BuildEvents

Inherited Members

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

Examples

```
// Customize materials when items are placed
BuildEvents.OnGridItemCreated += (args) =>
{
    if (args.ItemId == "my_custom_rack")
    {
        MaterialHelper.ReplaceMaterials(args.GameObject, ...);
    }
};
```

Events

OnBuildableItemInitialized

Event raised after a buildable item component is initialized. This event fires for all buildable items and can be used for additional setup.

```
public static event Action<BuildEventArgs> OnBuildableItemInitialized
```

Event Type

[Action](#)<BuildEventArgs>

OnGridItemCreated

Event raised after a grid item (wall-mounted or floor-placed) is created in the world. This is the most common build event for furniture and storage items.

```
public static event Action<BuildEventArgs> OnGridItemCreated
```

Event Type

[Action](#)<BuildEventArgs>

Remarks

Subscribers receive a BuildEventArgs containing the item and GameObject. The GameObject can be modified to change appearance or behavior.

OnSurfaceItemCreated

Event raised after a surface item (table-top item) is created.

```
public static event Action<BuildEventArgs> OnSurfaceItemCreated
```

Event Type

[Action](#)<BuildEventArgs>

Namespace S1API.Cartel

Classes

[Cartel](#)

Provides access to Cartel status and state information. Wraps the game's Cartel singleton to provide a modder-friendly API.

[CartelGoon](#)

Represents a cartel goon (enemy NPC) that can be spawned and controlled.

[CartelInfluence](#)

Provides access to cartel influence per map region. Influence ranges from 0 (no cartel presence) to 1 (full cartel control).

[GoonManager](#)

Manages the spawning and tracking of cartel goons. Provides access to the game's goon pool for spawning hostile NPCs.

Enums

[CartelStatus](#)

Represents the current status/relationship state of the Cartel.

Class Cartel

Namespace: [S1API.Cartel](#)

Assembly: S1API.dll

Provides access to Cartel status and state information. Wraps the game's Cartel singleton to provide a modder-friendly API.

```
public sealed class Cartel
```

Inheritance

[object](#) ← Cartel

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

GoonPool

Gets the goon manager for spawning and controlling cartel goons. Returns null if the goon pool is not available.

```
public GoonManager? GoonPool { get; }
```

Property Value

[GoonManager](#)

HoursSinceStatusChange

The number of hours since the Cartel status last changed.

```
public int HoursSinceStatusChange { get; }
```

Property Value

[int](#)

Influence

Gets the cartel influence manager for regional influence tracking. Returns null if the influence system is not available.

```
public CartelInfluence? Influence { get; }
```

Property Value

[CartelInfluence](#)

Instance

Gets the current Cartel instance, or null if not available.

```
public static Cartel? Instance { get; }
```

Property Value

[Cartel](#)

Status

The current status of the Cartel.

```
public CartelStatus Status { get; }
```

Property Value

[CartelStatus](#)

Methods

SetStatus(CartelStatus, bool)

Sets the cartel status. This is a server RPC that will sync to all clients.

```
public void SetStatus(CartelStatus status, bool resetTimer = true)
```

Parameters

status [CartelStatus](#)

The new cartel status.

resetTimer [bool](#)

Whether to reset the hours since status change timer.

Events

OnStatusChange

Event fired when the Cartel status changes. Provides the old status and new status as parameters.

```
public event Action<CartelStatus, CartelStatus> OnStatusChange
```

Event Type

[Action](#) ↴ <CartelStatus, CartelStatus>

Class CartelGoon

Namespace: [S1API.Carte](#)

Assembly: S1API.dll

Represents a cartel goon (enemy NPC) that can be spawned and controlled.

```
public class CartelGoon
```

Inheritance

[object](#) ← CartelGoon

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

GameObject

The GameObject associated with this goon.

```
public GameObject GameObject { get; }
```

Property Value

GameObject

IsConscious

Whether this goon is still conscious (alive and not knocked out).

```
public bool IsConscious { get; }
```

Property Value

[bool](#) ↗

IsSpawned

Whether this goon is currently spawned in the world.

```
public bool IsSpawned { get; }
```

Property Value

[bool](#) ↗

Position

The current world position of this goon.

```
public Vector3 Position { get; }
```

Property Value

Vector3

Methods

Attack(IEntity)

Makes this goon attack a specific entity.

```
public void Attack(IEntity target)
```

Parameters

target [IEntity](#)

The target entity to attack.

AttackPlayer()

Makes this goon attack the local player.

```
public void AttackPlayer()
```

Despawn()

Despawns this goon, removing them from the world.

```
public void Despawn()
```

SetDefaultWeapon(string?)

Sets or clears the default weapon for this goon. Pass null to make them use fists.

```
public void SetDefaultWeapon(string? weaponAssetPath)
```

Parameters

weaponAssetPath [string](#)

The weapon asset path, or null for fists.

WarpTo(Vector3)

Teleports this goon to a specific world position.

```
public void WarpTo(Vector3 position)
```

Parameters

position Vector3

The target position.

Class CartelInfluence

Namespace: [S1API.CarteI](#)

Assembly: S1API.dll

Provides access to cartel influence per map region. Influence ranges from 0 (no cartel presence) to 1 (full cartel control).

```
public class CartelInfluence
```

Inheritance

[object](#) ← CartelInfluence

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

ChangeInfluence(Region, float)

Changes the cartel influence in a region by a specified amount. Positive values increase influence, negative values decrease it.

```
public void ChangeInfluence(Region region, float amount)
```

Parameters

region [Region](#)

The map region to modify.

amount [float](#)

The amount to change (-1.0 to 1.0).

GetInfluence(Region)

Gets the cartel influence level for a specific region.

```
public float GetInfluence(Region region)
```

Parameters

region [Region](#)

The map region to check.

Returns

[float](#)

Influence level from 0.0 to 1.0

Events

OnInfluenceChanged

Event fired when cartel influence changes in any region. Parameters: region, old influence, new influence.

Note: This event is only available in Mono builds.

```
public event Action<Region, float, float> OnInfluenceChanged
```

Event Type

[Action](#)<[Region](#), [float](#), [float](#)>

Enum CartelStatus

Namespace: [S1API.Cartel](#)

Assembly: S1API.dll

Represents the current status/relationship state of the Cartel.

```
public enum CartelStatus
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Defeated = 3

Defeated status - cartel has been defeated.

Hostile = 2

Hostile status - cartel is actively hostile toward the player.

Truced = 1

Truced status - cartel is friendly/neutral with the player.

Unknown = 0

Unknown status - initial state before cartel relationship is established.

Class GoonManager

Namespace: [S1API.Carte](#)

Assembly: S1API.dll

Manages the spawning and tracking of cartel goons. Provides access to the game's goon pool for spawning hostile NPCs.

```
public class GoonManager
```

Inheritance

[object](#) ← GoonManager

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

AvailableGoonCount

The number of goons available to spawn from the pool.

```
public int AvailableGoonCount { get; }
```

Property Value

[int](#)

Methods

SpawnGoon(Vector3?)

Spawns a single goon at the specified position.

```
public CartelGoon? SpawnGoon(Vector3? position)
```

Parameters

position Vector3

The world position to spawn the goon at.

Returns

[CartelGoon](#)

The spawned goon, or null if no goons are available.

SpawnGoons(Vector3, int)

Spawns multiple goons at the specified position.

```
public List<CartelGoon> SpawnGoons(Vector3 position, int count)
```

Parameters

position Vector3

The world position to spawn the goons at.

count [int](#)

The number of goons to spawn.

Returns

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

A list of spawned goons. May contain fewer than requested if pool is depleted.

SpawnGoonsAtPositions(Vector3[])

Spawns goons at multiple positions, one goon per position.

```
public List<CartelGoon> SpawnGoonsAtPositions(Vector3[] positions)
```

Parameters

positions Vector3[]

The positions to spawn goons at.

Returns

[List](#)<CartelGoon>

A list of spawned goons with their positions set.

Namespace S1API.Casino

Classes

[SlotMachineHelper](#)

Provides a modder-facing API for interacting with slot machines without exposing game types.

Handles finding slot machines, managing NPC cash, and triggering spins.

Enums

[GamblingSessionMode](#)

Defines how long an NPC should continue using a slot machine.

[SlotMachineHelper.Outcome](#)

Outcome categories for slot machine spins.

Enum GamblingSessionMode

Namespace: [S1API.Casino](#)

Assembly: S1API.dll

Defines how long an NPC should continue using a slot machine.

```
public enum GamblingSessionMode
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`SingleSpin = 0`

Play the slot machine exactly once.

`SpinCount = 1`

Play until a specific number of spins is reached.

`UntilBroke = 3`

Play until the NPC runs out of cash for the bet amount.

`UntilTime = 2`

Play until a specific time is reached.

`UntilTimeOrBroke = 4`

Play until time is reached OR the NPC runs out of cash, whichever comes first.

Class SlotMachineHelper

Namespace: [S1API.Casino](#)

Assembly: S1API.dll

Provides a modder-facing API for interacting with slot machines without exposing game types. Handles finding slot machines, managing NPC cash, and triggering spins.

```
public static class SlotMachineHelper
```

Inheritance

[object](#) ← SlotMachineHelper

Inherited Members

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

Methods

AddNPCCash(NPC, int)

Adds cash to an NPC's inventory. Tries to reuse existing cash stacks, otherwise ensures there's a spare slot (expanding the slot count if needed) and inserts a single cash item.

```
public static void AddNPCCash(NPC npc, int amount)
```

Parameters

npc [NPC](#)

The NPC to add cash to.

amount [int](#)

The amount of cash to add in dollars.

FindNearestSlotMachine(Vector3, float)

Finds the nearest slot machine to a given position.

```
public static SlotMachine FindNearestSlotMachine(Vector3 position,  
float maxDistance)
```

Parameters

position Vector3

The position to search from.

maxDistance [float](#)

Maximum distance to search.

Returns

SlotMachine

The nearest slot machine, or null if none found.

GetNPCCash(NPC)

Gets the total cash value from an NPC's inventory (cash items).

```
public static float GetNPCCash(NPC npc)
```

Parameters

npc [NPC](#)

The NPC whose cash to count.

Returns

[float](#)

Total cash amount in dollars.

RemoveNPCCash(NPC, int)

Removes a specified amount of cash from an NPC's inventory.

```
public static bool RemoveNPCCash(NPC npc, int amount)
```

Parameters

[npc](#) [NPC](#)

The NPC to remove cash from.

[amount](#) [int](#)

The amount of cash to remove in dollars.

Returns

[bool](#)

True if the full amount was successfully removed; false otherwise.

UseSlotMachine(NPC, Vector3, int, float)

Makes an NPC use a slot machine with the specified bet amount. This handles all cash transactions, animations, and outcome determination automatically.

```
public static bool UseSlotMachine(NPC npc, Vector3 machinePosition, int betAmount,  
float maxSearchDistance = 5)
```

Parameters

npc [NPC](#)

The NPC that will use the slot machine.

machinePosition [Vector3](#)

The world position of the slot machine.

betAmount [int](#)

The amount to bet in dollars.

maxSearchDistance [float](#)

Maximum distance to search for a slot machine from the specified position.

Returns

[bool](#)

True if the NPC successfully started using a slot machine; false otherwise.

Enum SlotMachineHelper.Outcome

Namespace: [S1API.Casino](#)

Assembly: S1API.dll

Outcome categories for slot machine spins.

```
public enum SlotMachineHelper.Outcome
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`BigWin = 1`

Big win - three bells.

`Jackpot = 0`

Jackpot - three sevens.

`MiniWin = 3`

Mini win - any three fruits.

`NoWin = 4`

No win.

`SmallWin = 2`

Small win - three matching fruits.

Namespace S1API.Conditions

Classes

[SystemTriggerEntry](#)

Wraps an in-game system trigger that evaluates configured conditions and applies variable or quest state setters. Exposes events for evaluation pass/fail.

Class SystemTriggerEntry

Namespace: [S1API.Conditions](#)

Assembly: S1API.dll

Wraps an in-game system trigger that evaluates configured conditions and applies variable or quest state setters. Exposes events for evaluation pass/fail.

```
public class SystemTriggerEntry
```

Inheritance

[object](#) ← SystemTriggerEntry

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Methods

AddQuestSetter(EvaluationType, Quest, QuestAction?, Tuple<int, QuestState>?)

Creates a ScheduleOne.Quests.QuestStateSetter instance

```
public void AddQuestSetter(EvaluationType evaluation, Quest questData, QuestAction?  
questAction = null, Tuple<int, QuestState>? questEntryState = null)
```

Parameters

evaluation [EvaluationType](#)

The condition to use

questData [Quest](#)

The [Quest](#) instance to use for this condition

questAction [QuestAction?](#)

(Optional) The state of the quest

questEntryState [Tuple<int, QuestState>](#)

(Optional) The state of the quest entry

AddVariableSetter(EvaluationType, string, string)

Creates a ScheduleOne.Variables.VariableSetter instance

```
public void AddVariableSetter(EvaluationType evaluation, string variableName,  
string newValue)
```

Parameters

evaluation [EvaluationType](#)

The condition to use

variableName [string](#)

The variable to use

newValue [string](#)

The new value for this variable

Trigger()

Trigger the conditions for evaluation

```
public void Trigger()
```

Events

OnEvaluateFalse

An action called when the ScheduleOne.Quests.SystemTrigger.Conditions is false

```
public event Action OnEvaluateFalse
```

Event Type

[Action ↗](#)

OnEvaluateTrue

An action called when the ScheduleOne.Quests.SystemTrigger.Conditions is true

```
public event Action OnEvaluateTrue
```

Event Type

[Action ↗](#)

Namespace S1API.Console

Classes

[BaseConsoleCommand](#)

Abstract base class for creating console commands. Inherit from this base class to auto-register a custom command in the API.

[ConsoleHelper](#)

Provides a stable, modder-friendly abstraction over the in-game console system. Use these helpers from mods instead of referencing the game's console types directly.

Class BaseConsoleCommand

Namespace: [S1API.Console](#)

Assembly: S1API.dll

Abstract base class for creating console commands. Inherit from this base class to auto-register a custom command in the API.

```
public abstract class BaseConsoleCommand
```

Inheritance

[object](#) ← BaseConsoleCommand

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BaseConsoleCommand()

```
protected BaseConsoleCommand()
```

Properties

CommandDescription

A brief description of what the command does.

```
public abstract string CommandDescription { get; }
```

PropertyValue

[string](#)

CommandWord

The command word that triggers this console command.

```
public abstract string CommandWord { get; }
```

PropertyValue

[string](#)

ExampleUsage

An example of how to use the command.

```
public abstract string ExampleUsage { get; }
```

PropertyValue

[string](#)

Methods

ExecuteCommand(List<string>)

Executes the command with the provided arguments.

```
public abstract void ExecuteCommand(List<string> args)
```

Parameters

args [List<string>](#)

The list of arguments passed to the command.

Class ConsoleHelper

Namespace: [S1API.Console](#)

Assembly: S1API.dll

Provides a stable, modder-friendly abstraction over the in-game console system. Use these helpers from mods instead of referencing the game's console types directly.

```
public static class ConsoleHelper
```

Inheritance

[object](#) ← ConsoleHelper

Inherited Members

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

Methods

AddItemToInventory(string, int?)

Gives the player an item by code. Optionally specify a quantity.

```
public static void AddItemToInventory(string itemCode, int? quantity = null)
```

Parameters

itemCode [string](#)

quantity [int](#)?

ClearInventory()

Clears the player's inventory.

```
public static void ClearInventory()
```

ClearTrash()

Instantly removes all trash from the world.

```
public static void ClearTrash()
```

ClearWanted()

Clears the player's wanted level and crimes.

```
public static void ClearWanted()
```

DiscoverProduct(string)

Marks a product as discovered by its item code.

```
public static void DiscoverProduct(string productCode)
```

Parameters

productCode [string](#)

GiveXp(int)

Gives the player experience points.

```
public static void GiveXp(int amount)
```

Parameters

amount int ↗

GrowPlants()

Instantly sets all plants in the world to fully grown.

```
public static void GrowPlants()
```

LowerWanted()

Lowers the player's wanted level.

```
public static void LowerWanted()
```

RaiseWanted()

Raises the player's wanted level.

```
public static void RaiseWanted()
```

RunCashCommand(int)

Executes the ChangeCash command with the given amount. Positive values add cash; negative values remove cash.

```
public static void RunCashCommand(int amount)
```

Parameters

amount [int](#)

RunOnlineBalanceCommand(int)

Changes the player's online bank balance by the specified amount.

```
public static void RunOnlineBalanceCommand(int amount)
```

Parameters

amount [int](#)

SaveGame()

Forces a save of the current game state.

```
public static void SaveGame()
```

SetLawIntensity(float)

Sets the intensity of law enforcement activity (0-10).

```
public static void SetLawIntensity(float intensity)
```

Parameters

intensity [float](#)

SetNpcRelationship(NPC, float)

Sets the relationship scalar for an NPC (0-5).

```
public static void SetNpcRelationship(NPC npc, float level)
```

Parameters

npc [NPC](#)

level [float](#)

SetNpcRelationship(string, float)

Sets the relationship scalar for an NPC by id (0-5).

```
public static void SetNpcRelationship(string npcId, float level)
```

Parameters

npcId [string](#)

level [float](#)

SetPlayerEnergyLevel(float)

Sets the player's energy to a value between 0 and 100.

```
public static void SetPlayerEnergyLevel(float amount)
```

Parameters

amount [float](#)

SetPlayerHealth(float)

Sets the player's health to the specified value.

```
public static void SetPlayerHealth(float amount)
```

Parameters

amount [float](#)

SetPlayerJumpMultiplier(float)

Sets the player's jump force multiplier. Must be non-negative.

```
public static void SetPlayerJumpMultiplier(float multiplier)
```

Parameters

multiplier [float](#)

SetPlayerMoveSpeedMultiplier(float)

Sets the player's movement speed multiplier. Must be non-negative.

```
public static void SetPlayerMoveSpeedMultiplier(float multiplier)
```

Parameters

multiplier [float](#)

SetQuality(Quality)

Sets the equipped item's quality.

```
public static void SetQuality(Quality quality)
```

Parameters

quality [Quality](#)

API quality value to set.

SetQuestState(string, QuestState)

Sets the state of a quest by name.

```
public static void SetQuestState(string questName, QuestState state)
```

Parameters

questName [string](#)

state [QuestState](#)

SetTime(string)

Sets the time of day using a 24h HHmm string (e.g., "1530").

```
public static void SetTime(string hhmm)
```

Parameters

hhmm [string](#)

SpawnVehicle(string)

Spawns a vehicle by code at the player's location.

```
public static void SpawnVehicle(string vehicleCode)
```

Parameters

vehicleCode [string](#)

Submit(IEnumerable<string>)

Submits a console command and arguments (e.g. Submit(["settime","1530"])). Works across both IL2CPP and Mono builds.

```
public static void Submit(IEnumerable<string> arguments)
```

Parameters

arguments [IEnumerable](#)<[string](#)>

Command word followed by its arguments.

Submit(string)

Submits a raw console command string (e.g. "settime 1530"). Works across both IL2CPP and Mono builds.

```
public static void Submit(string command)
```

Parameters

command [string](#)

The full command line to execute.

UnlockNpc(NPC)

Unlocks the given NPC's connection.

```
public static void UnlockNpc(NPC npc)
```

Parameters

npc [NPC](#)

Namespace S1API.DeadDrops

Classes

[DeadDropInstance](#)

Represents a dead drop in the scene.

[DeadDropManager](#)

Provides access to dead drops present in the scene.

Class DeadDropInstance

Namespace: [S1API.DeadDrops](#)

Assembly: S1API.dll

Represents a dead drop in the scene.

```
public class DeadDropInstance
```

Inheritance

[object](#) ← DeadDropInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Description

The descriptive label or location hint for the dead drop.

```
public string Description { get; }
```

Property Value

[string](#)

GUID

The unique identifier assigned for this dead drop.

```
public string GUID { get; }
```

Property Value

[string](#)

IsEmpty

Whether the dead drop contains no items.

```
public bool IsEmpty { get; }
```

Property Value

[bool](#)

ItemCount

The current number of items stored in the dead drop.

```
public int ItemCount { get; }
```

Property Value

[int](#)

Name

The display name of the dead drop.

```
public string Name { get; }
```

Property Value

[string](#)

Position

The world position of the dead drop.

```
public Vector3 Position { get; }
```

Property Value

Vector3

Region

The region this dead drop belongs to.

```
public Region Region { get; }
```

Property Value

[Region](#)

Storage

The storage container associated with this dead drop.

```
public StorageInstance Storage { get; }
```

Property Value

StorageInstance

Class DeadDropManager

Namespace: [S1API.DeadDrops](#)

Assembly: S1API.dll

Provides access to dead drops present in the scene.

```
public static class DeadDropManager
```

Inheritance

[object](#) ← DeadDropManager

Inherited Members

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

Properties

All

All dead drops currently registered in the scene.

```
public static DeadDropInstance[] All { get; }
```

Property Value

[DeadDropInstance\[\]](#)

Empty

All dead drops that contain no items.

```
public static DeadDropInstance[] Empty { get; }
```

Property Value

[DeadDropInstance\[\]](#)

Methods

GetByGUID(string)

Gets a dead drop by its GUID.

```
public static DeadDropInstance? GetByGUID(string guid)
```

Parameters

`guid` [string](#)

The GUID string to look up.

Returns

[DeadDropInstance](#)

The dead drop instance if found; otherwise null.

GetClosest(Vector3?, bool)

Gets the closest dead drop to a world position.

```
public static DeadDropInstance? GetClosest(Vector3? origin, bool mustBeEmpty  
= false)
```

Parameters

`origin` [Vector3](#)

World position to measure from.

mustBeEmpty [bool](#)

If true, only considers empty dead drops.

Returns

[DeadDropInstance](#)

The closest matching dead drop, or null if none exist.

GetRandomEmptyNear(Vector3?)

Gets a random empty dead drop near a world position. Applies a light bias: avoids the absolute nearest, then chooses randomly among closer half.

```
public static DeadDropInstance? GetRandomEmptyNear(Vector3? origin)
```

Parameters

origin [Vector3](#)

World position to bias selection around.

Returns

[DeadDropInstance](#)

A random nearby empty dead drop, or null if none exist.

Namespace S1API.Dialogues

Classes

[DialogueChoiceListener](#)

A static utility class that listens for and responds to specific dialogue choices in the game's dialogue system.

[DialogueInjection](#)

Represents a dialogue injection configuration for adding custom dialogues into an NPC's conversation flow dynamically.

[DialogueInjector](#)

The DialogueInjector class is a static utility that facilitates the injection of custom dialogue entries into a game's dialogue system at runtime. It provides methods for registering custom dialogue injections and ensures that these injections are processed correctly within the update loop.

Class DialogueChoiceListener

Namespace: [S1API.Dialogues](#)

Assembly: S1API.dll

A static utility class that listens for and responds to specific dialogue choices in the game's dialogue system.

```
public static class DialogueChoiceListener
```

Inheritance

[object](#) ← DialogueChoiceListener

Inherited Members

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

Methods

Register(DialogueHandler, string, Action)

```
public static void Register(DialogueHandler handlerRef, string label, Action action)
```

Parameters

handlerRef DialogueHandler

The reference to the DialogueHandler that manages dialogue choices.

label [string](#)

The label identifying the specific dialogue choice to be registered.

action [Action](#)

The callback action to execute when the dialogue choice is selected.

Class DialogueInjection

Namespace: [S1API.Dialogues](#)

Assembly: S1API.dll

Represents a dialogue injection configuration for adding custom dialogues into an NPC's conversation flow dynamically.

```
public class DialogueInjection
```

Inheritance

[object](#) ← DialogueInjection

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DialogueInjection(Predicate<NPC>, string, string, string, string, string, Action)

Represents an injectable dialogue configuration that can be used to add or modify dialogue interactions in a game.

```
public DialogueInjection(Predicate<NPC> appliesToNpc, string container, string from,  
string to, string label, string text, Action onConfirmed)
```

Parameters

appliesToNpc [Predicate](#)<NPC>

container [string](#)

from [string](#)

`to` [string](#)

`label` [string](#)

`text` [string](#)

`onConfirmed` [Action](#)

DialogueInjection(string, string, string, string, string, string, Action)

```
public DialogueInjection(string npc, string container, string from, string to,  
string label, string text, Action onConfirmed)
```

Parameters

`npc` [string](#)

`container` [string](#)

`from` [string](#)

`to` [string](#)

`label` [string](#)

`text` [string](#)

`onConfirmed` [Action](#)

Fields

AppliesTo

```
public Predicate<NPC> AppliesTo
```

Field Value

ChoiceLabel

Represents a descriptive label for a dialogue choice used in the dialogue system.

```
public string ChoiceLabel
```

Field Value

[string](#)

Remarks

This label is utilized for identifying a specific dialogue choice during execution and for associating a callback or specific functionality when that choice is selected.

ChoiceText

Represents the text displayed for a dialogue choice in the game's dialogue system.

```
public string ChoiceText
```

Field Value

[string](#)

Remarks

The property is utilized to define the text that appears visually for a specific dialogue choice in conjunction with the dialogue system. The text is injected dynamically during runtime for scenarios requiring additional or modified dialogue options.

ContainerName

Represents the name of the dialogue container being referenced for injections or modifications within the NPC's dialogue system.

```
public string ContainerName
```

Field Value

[string ↗](#)

Remarks

This variable is used for identifying a specific dialogue container when attempting to inject new dialogue nodes, choices, or links into an NPC's dialogue setup.

FromNodeGuid

Represents the unique identifier (GUID) of the starting dialogue node within a dialogue container.

```
public string FromNodeGuid
```

Field Value

[string ↗](#)

Remarks

This variable is used to identify the specific dialogue node from which a new choice or interaction is injected.

OnConfirmed

Represents a callback action that is invoked when a dialogue choice is confirmed.

```
public Action OnConfirmed
```

Field Value

[Action ↗](#)

ToNodeGuid

Represents the unique identifier (GUID) for the target dialogue node to which a choice or link is pointing in a dialogue system.

```
public string ToNodeGuid
```

Field Value

[string ↗](#)

Class DialogueInjector

Namespace: [S1API.Dialogues](#)

Assembly: S1API.dll

The DialogueInjector class is a static utility that facilitates the injection of custom dialogue entries into a game's dialogue system at runtime. It provides methods for registering custom dialogue injections and ensures that these injections are processed correctly within the update loop.

```
public static class DialogueInjector
```

Inheritance

[object](#) ← DialogueInjector

Inherited Members

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

Methods

Register(DialogueInjection)

Registers a dialogue injection to be processed in the update loop.

```
public static void Register(DialogueInjection injection)
```

Parameters

`injection` [DialogueInjection](#)

An instance of [DialogueInjection](#) representing the dialogue to be injected into the game.

Namespace S1API.Economy

Classes

[Contract](#)

Read-only wrapper for a base game Contract quest instance.

[ContractInfo](#)

Builder/DTO for offering a contract to a customer.

[ContractInfo.OrderLine](#)

A single product order line in the contract.

[ContractInfoBuilder](#)

Builder for composing ContractInfo at runtime with a fluent API. Provides an easy way to create contracts for NPC customers.

[ContractReceipt](#)

Lightweight DTO mirroring base game's ContractReceipt for analytics/history.

Enums

[CustomerStandard](#)

API-safe customer standards. Mirrors base game values without exposing game types.

[DealerType](#)

Type of dealer behavior for NPCs.

Class Contract

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

Read-only wrapper for a base game Contract quest instance.

```
public sealed class Contract
```

Inheritance

[object](#) ← Contract

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Payment

Base payment amount for the contract.

```
public float Payment { get; }
```

Property Value

[float](#)

TotalQuantity

Total ordered quantity across all entries.

```
public int TotalQuantity { get; }
```

Property Value

[int](#)

WindowEndTime

Delivery window end time (hhmm) if enabled; otherwise 0.

```
public int WindowEndTime { get; }
```

Property Value

[int](#)

WindowStartTime

Delivery window start time (hhmm) if enabled; otherwise 0.

```
public int WindowStartTime { get; }
```

Property Value

[int](#)

Methods

GetOrders()

Enumerates order lines for this contract.

```
public IEnumerable<(string productId, int quantity, Quality minQuality)> GetOrders()
```

Returns

[IEnumerable](#)<(string [productId](#), int [quantity](#), Quality [minQuality](#))>

Class ContractInfo

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

Builder/DTO for offering a contract to a customer.

```
public sealed class ContractInfo
```

Inheritance

[object](#) ← ContractInfo

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ContractInfo()

```
public ContractInfo()
```

Properties

DeliveryLocationGuid

Delivery location GUID (optional). If null or invalid, a reasonable location will be chosen.

```
public string DeliveryLocationGuid { get; set; }
```

Property Value

[string](#)

DeliveryWindow

Optional delivery window. If not set, game defaults are used.

```
public (int startTime, int endTime)? DeliveryWindow { get; set; }
```

Property Value

([int](#) [startTime](#), [int](#) [endTime](#))?

Expires

Whether the offer expires automatically.

```
public bool Expires { get; set; }
```

Property Value

[bool](#)

ExpiresAfterMinutes

Minutes until expiry if [Expires](#) is true.

```
public int ExpiresAfterMinutes { get; set; }
```

Property Value

[int](#)

IsCounterOffer

Whether this is a counter offer.

```
public bool IsCounterOffer { get; set; }
```

Property Value

[bool](#)

Orders

Ordered products.

```
public List<ContractInfo.OrderLine> Orders { get; }
```

Property Value

[List](#)<[ContractInfo.OrderLine](#)>

Payment

Contract payment (base, excluding bonuses).

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

Property Value

[float](#)

PickupScheduleIndex

Optional pickup schedule index; 0 is typical for immediate offers.

```
public int PickupScheduleIndex { get; set; }
```

Property Value

[int](#)

Methods

AddProduct(ContractInfo, ProductDefinition, int, Quality)

Adds a product by API definition.

```
public ContractInfo AddProduct(ProductDefinition definition, int quantity,  
Quality minQuality)
```

Parameters

definition [ProductDefinition](#)

quantity [int](#)

minQuality [Quality](#)

Returns

[ContractInfo](#)

AddProductById(string, int, Quality)

Adds a product by registry ID.

```
public ContractInfo AddProductById(string productId, int quantity,  
Quality minQuality)
```

Parameters

productId [string](#)

quantity [int](#)

minQuality [Quality](#)

Returns

[ContractInfo](#)

WithWindow(int, int)

Sets the delivery window in 24h hhmm format.

```
public ContractInfo WithWindow(int startTime, int endTime)
```

Parameters

startTime [int](#)

endTime [int](#)

Returns

[ContractInfo](#)

Class ContractInfo.OrderLine

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

A single product order line in the contract.

```
public sealed class ContractInfo.OrderLine
```

Inheritance

[object](#) ← ContractInfo.OrderLine

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

OrderLine()

```
public OrderLine()
```

Properties

MinQuality

Minimum acceptable quality.

```
public Quality MinQuality { get; set; }
```

PropertyValue

Quality

ProductId

Product registry ID.

```
public string ProductId { get; set; }
```

PropertyValue

string ↗

Quantity

Quantity of this product to deliver.

```
public int Quantity { get; set; }
```

PropertyValue

int ↗

Class ContractInfoBuilder

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

Builder for composing ContractInfo at runtime with a fluent API. Provides an easy way to create contracts for NPC customers.

```
public sealed class ContractInfoBuilder
```

Inheritance

[object](#) ← ContractInfoBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ContractInfoBuilder()

Creates a new ContractInfoBuilder instance.

```
public ContractInfoBuilder()
```

Methods

AddProduct(ProductDefinition, int, Quality)

Adds a product to the contract by product definition.

```
public ContractInfoBuilder AddProduct(ProductDefinition definition, int quantity,  
Quality minQuality)
```

Parameters

definition [ProductDefinition](#)

The product definition

quantity [int](#)

Quantity to order (minimum 1)

minQuality [Quality](#)

Minimum acceptable quality

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

AddProducts(IEnumerable<(ProductDefinition definition, int quantity, Quality minQuality)>)

Adds multiple products to the contract.

```
public ContractInfoBuilder AddProducts(IEnumerable<(ProductDefinition definition,  
int quantity, Quality minQuality)> products)
```

Parameters

products [IEnumerable](#)<(ProductDefinition [definition](#), [int](#) [quantity](#), [Quality](#) [minQuality](#))>

Collection of (definition, quantity, minQuality) tuples

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

AsCounterOffer(bool)

Marks this contract as a counter offer.

```
public ContractInfoBuilder AsCounterOffer(bool isCounterOffer = true)
```

Parameters

[isCounterOffer](#) [bool](#)

Whether this is a counter offer

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

Build()

Builds the final ContractInfo instance.

```
public ContractInfo Build()
```

Returns

[ContractInfo](#)

A new ContractInfo instance with the configured settings

Exceptions

[InvalidOperationException](#)

Thrown if no products have been added

ExpiresAfter(int)

Sets the expiration time in minutes.

```
public ContractInfoBuilder ExpiresAfter(int minutes)
```

Parameters

[minutes](#) [int](#)

Minutes until expiry (minimum 0)

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

QuickContract(ProductDefinition, int, float, Quality)

Creates a quick contract with a single product.

```
public static ContractInfo QuickContract(ProductDefinition definition, int quantity,
                                         float payment, Quality minQuality = Quality.Standard)
```

Parameters

[definition](#) [ProductDefinition](#)

Product definition

quantity [int](#)

Quantity

payment [float](#)

Payment amount

minQuality [Quality](#)

Minimum quality (defaults to Standard)

Returns

[ContractInfo](#)

A new ContractInfo instance

WithDeliveryLocation(DeliveryLocation)

Sets the delivery location using a DeliveryLocation wrapper.

```
public ContractInfoBuilder WithDeliveryLocation(DeliveryLocation location)
```

Parameters

location [DeliveryLocation](#)

The delivery location

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithDeliveryLocationByGuid(string)

Sets the delivery location by GUID using DeliveryLocations.GetByGuid().

```
public ContractInfoBuilder WithDeliveryLocationByGuid(string locationGuid)
```

Parameters

locationGuid [string](#)

The delivery location GUID

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithDeliveryLocationByName(string)

Sets the delivery location by name using DeliveryLocations.GetByName().

```
public ContractInfoBuilder WithDeliveryLocationByName(string locationName)
```

Parameters

locationName [string](#)

The delivery location name

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithDeliveryWindow(int, int)

Sets the delivery window in 24h hhmm format.

```
public ContractInfoBuilder WithDeliveryWindow(int startTime, int endTime)
```

Parameters

startTime [int](#)

Start time (e.g., 900 for 9:00 AM)

endTime [int](#)

End time (e.g., 1700 for 5:00 PM)

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithDeliveryWindow(TimeSpan, TimeSpan)

Sets the delivery window using TimeSpan for convenience.

```
public ContractInfoBuilder WithDeliveryWindow(TimeSpan startTime, TimeSpan endTime)
```

Parameters

startTime [TimeSpan](#)

Start time

endTime [TimeSpan](#)

End time

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithExpiration(bool)

Sets whether the contract expires automatically.

```
public ContractInfoBuilder WithExpiration(bool expires)
```

Parameters

`expires` [bool](#)

Whether the contract expires

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithPayment(float)

Sets the contract payment amount.

```
public ContractInfoBuilder WithPayment(float payment)
```

Parameters

`payment` [float](#)

The payment amount (must be ≥ 0)

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithPickupScheduleIndex(int)

Sets the pickup schedule index.

```
public ContractInfoBuilder WithPickupScheduleIndex(int index)
```

Parameters

`index` [int](#)

Pickup schedule index (minimum 0)

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

WithoutDeliveryWindow()

Removes the delivery window constraint.

```
public ContractInfoBuilder WithoutDeliveryWindow()
```

Returns

[ContractInfoBuilder](#)

This builder instance for method chaining

Class ContractReceipt

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

Lightweight DTO mirroring base game's ContractReceipt for analytics/history.

```
public sealed class ContractReceipt
```

Inheritance

[object](#) ← ContractReceipt

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ContractReceipt()

```
public ContractReceipt()
```

Properties

AmountPaid

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

Property Value

[float](#)

CompletionTime

```
public (int days, int time) CompletionTime { get; set; }
```

Property Value

([int](#) [days](#), [int](#) [time](#))

CustomerId

```
public string CustomerId { get; set; }
```

Property Value

[string](#)

Items

```
public (string id, int quantity)[] Items { get; set; }
```

Property Value

([string](#) [id](#), [int](#) [quantity](#))[]

ReceiptId

```
public int ReceiptId { get; set; }
```

Property Value

[int](#) ↗

Enum CustomerStandard

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

API-safe customer standards. Mirrors base game values without exposing game types.

```
public enum CustomerStandard
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

High = 3

Low = 1

Moderate = 2

VeryHigh = 4

VeryLow = 0

Enum DealerType

Namespace: [S1API.Economy](#)

Assembly: S1API.dll

Type of dealer behavior for NPCs.

```
public enum DealerType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`CartelDealer = 1`

Dealer that works for the cartel, independent of player control.

`PlayerDealer = 0`

Dealer that works for the player, selling products to assigned customers.

Namespace S1API.Entities

Namespaces

[S1API.Entities.Appearances](#)

[S1API.Entities.Behaviour](#)

[S1API.Entities.Customer](#)

[S1API.Entities.Dealer](#)

[S1API.Entities.Dialogue](#)

[S1API.Entities.Equipables](#)

[S1API.Entities.Interfaces](#)

[S1API.Entities.NPCs](#)

[S1API.Entities.Relation](#)

[S1API.Entities.Schedule](#)

Classes

[NPC](#)

Abstract base class for creating custom NPCs with modular architecture supporting both physical and non-physical NPCs. Physical NPCs are visible in the game world with 3D models, movement, and direct interaction. Non-physical NPCs are invisible contacts primarily used for messaging and phone interactions.

[NPCCAppearance](#)

Modder-facing appearance customization system for NPCs. Provides builders to configure visual appearance including physical features, clothing, and accessories. Appearance configuration is done in [OnCreated\(\)](#).

[NPCCustomer](#)

Modder-facing customer wrapper for an NPC. Provides helpers to configure and interact with customer behavior, including deal offers, contracts, and customer events. Customer configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

[NPCDealer](#)

Modder-facing dealer wrapper for an NPC. Provides helpers to configure and interact with dealer behavior, including customer assignment, cash management, and contract handling. Dealer configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

[NPCDialogue](#)

Modder-facing dialogue wrapper for an NPC. Provides helpers to create interactive conversations with branching dialogue trees, choice-based interactions, and dynamic responses. Use [BuildAndRegisterContainer\(string, Action<DialogueContainerBuilder>\)](#) to define custom conversations.

[NPCInventory](#)

Modder-facing inventory wrapper for an S1API.Entities.NPCInventory.NPC. Provides helpers to query capacity and insert items safely.

[NPCMovement](#)

Modder-facing movement wrapper for an S1API.Entities.NPCMovement.NPC. Provides navigation, warping, facing, and reachability helpers. Keep in mind that as of API version 1.8.1, custom NPCs do not have an Avatar or an NPCMovement component.

[NPCPrefabBuilder](#)

Builder for composing NPC prefab configuration before network spawn. Use to declare networked components, spawn position, customer behavior, relationships, schedules, and appearance defaults.

[NPCPrefabBuilder.AvatarDefaultsBuilder](#)

Wrapper for authoring appearance defaults without exposing game types to modders.

[NPCRelationship](#)

Modder-facing wrapper for an NPC's relationship data. Provides safe access to relationship values, unlock state, connections, and convenience helpers which bridge to the base game's NPCRelationData. Relationship configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

[NPCSchedule](#)

Modder-facing scheduling wrapper for an NPC. Exposes the underlying schedule manager to enable, disable, and manage scheduled actions and curfew modes. Schedule configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

[Player](#)

Represents a player within the game.

[RandomInventoryItemsBuilder](#)

Builder for configuring startup items and random cash for NPCs. Public surface uses strings/primitives only. All configurations are optional.

Enums

[NPCRelationship.UnlockType](#)

Unlock types mirrored from the base game.

Namespace S1API.Entities.Appearances

Namespaces

[S1API.Entities.Appearances.AccessoryFields](#)

[S1API.Entities.Appearances.Base](#)

[S1API.Entities.Appearances.BodyLayerFields](#)

[S1API.Entities.Appearances.CustomizationFields](#)

[S1API.Entities.Appearances.FaceLayerFields](#)

Namespace S1API.Entities.Appearances. AccessoryFields

Classes

Bottom

The Bottom index in AvatarSettings::AccessorySettings

Chest

The Chest index in AvatarSettings::AccessorySettings

FacialHairAccessory

The FacialHair Accessory index in AvatarSettings::AccessorySettings

Feet

The Feet index in AvatarSettings::AccessorySettings

Hands

The Hands index in AvatarSettings::AccessorySettings

Head

The Head index in AvatarSettings::AccessorySettings

Neck

The Neck index in AvatarSettings::AccessorySettings

Waist

The Waist index in AvatarSettings::AccessorySettings

Class Bottom

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Bottom index in AvatarSettings::AccessorySettings

```
public class Bottom : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Bottom

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Bottom()

```
public Bottom()
```

Fields

LongSkirt

```
public const string LongSkirt = "Avatar/Accessories/Bottom/LongSkirt/LongSkirt"
```

Field Value

[string](#)

MediumSkirt

```
public const string MediumSkirt =  
"Avatar/Accessories/Bottom/MediumSkirt/MediumSkirt"
```

Field Value

[string](#)

Class Chest

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Chest index in AvatarSettings::AccessorySettings

```
public class Chest : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Chest

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Chest()

```
public Chest()
```

Fields

Blazer

```
public const string Blazer = "Avatar/Accessories/Chest/Blazer/Blazer"
```

Field Value

[string](#)

BulletProofVest

```
public const string BulletProofVest =  
"Avatar/Accessories/Chest/BulletProofVest/BulletProofVest"
```

Field Value

[string](#)

BulletProofVestPolice

```
public const string BulletProofVestPolice =  
"Avatar/Accessories/Chest/BulletProofVest/BulletProofVest_Police"
```

Field Value

[string](#)

CollarJacket

```
public const string CollarJacket =  
"Avatar/Accessories/Chest/CollarJacket/CollarJacket"
```

Field Value

[string](#)

OpenVest

```
public const string OpenVest = "Avatar/Accessories/Chest/OpenVest/OpenVest"
```

Field Value

[string ↗](#)

Class FacialHairAccessory

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The FacialHair Accessory index in AvatarSettings::AccessorySettings

```
public class FacialHairAccessory : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← FacialHairAccessory

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

FacialHairAccessory()

```
public FacialHairAccessory()
```

Fields

Chevron

```
public const string Chevron = "Avatar/Accessories/FacialHair/Chevron/Chevron"
```

Field Value

[string](#) ↗

Class Feet

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Feet index in AvatarSettings::AccessorySettings

```
public class Feet : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Feet

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Feet()

```
public Feet()
```

Fields

CombatBoots

```
public const string CombatBoots = "Avatar/Accessories/Feet/CombatBoots/CombatBoots"
```

Field Value

[string](#)

DressShoes

```
public const string DressShoes = "Avatar/Accessories/Feet/DressShoes/DressShoes"
```

Field Value

[string](#)

Flats

```
public const string Flats = "Avatar/Accessories/Feet/Flats/Flats"
```

Field Value

[string](#)

Sandals

```
public const string Sandals = "Avatar/Accessories/Feet/Sandals/Sandals"
```

Field Value

[string](#)

Sneakers

```
public const string Sneakers = "Avatar/Accessories/Feet/Sneakers/Sneakers"
```

Field Value

[string](#) 

Class Hands

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Hands index in AvatarSettings::AccessorySettings

```
public class Hands : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Hands

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Hands()

```
public Hands()
```

Fields

Polex

```
public const string Polex = "Avatar/Accessories/Hands/Polex/Polex"
```

Field Value

[string](#) ↗

Class Head

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Head index in AvatarSettings::AccessorySettings

```
public class Head : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Head

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Head()

```
public Head()
```

Fields

Beanie

```
public const string Beanie = "Avatar/Accessories/Head/Beanie/Beanie"
```

Field Value

[string](#)

BucketHat

```
public const string BucketHat = "Avatar/Accessories/Head/BucketHat/BucketHat"
```

Field Value

[string](#)

Cap

```
public const string Cap = "Avatar/Accessories/Head/Cap/Cap"
```

Field Value

[string](#)

CapFastFood

```
public const string CapFastFood = "Avatar/Accessories/Head/Cap/Cap_FastFood"
```

Field Value

[string](#)

ChefHat

```
public const string ChefHat = "Avatar/Accessories/Head/ChefHat/ChefHat"
```

Field Value

[string](#)

CowboyHat

```
public const string CowboyHat = "Avatar/Accessories/Head/Cowboy/CowboyHat"
```

Field Value

[string](#)

FlatCap

```
public const string FlatCap = "Avatar/Accessories/Head/FlatCap/FlatCap"
```

Field Value

[string](#)

LegendSunglasses

```
public const string LegendSunglasses =
"Avatar/Accessories/Head/LegendSunglasses/LegendSunglasses"
```

Field Value

[string](#)

MushroomHat

```
public const string MushroomHat = "Avatar/Accessories/Head/MushroomHat/MushroomHat"
```

Field Value

[string](#) ↗

Oakleys

```
public const string Oakleys = "Avatar/Accessories/Head/Oakleys/Oakleys"
```

Field Value

[string](#) ↗

PoliceCap

```
public const string PoliceCap = "Avatar/Accessories/Head/PoliceCap/PoliceCap"
```

Field Value

[string](#) ↗

PorkpieHat

```
public const string PorkpieHat = "Avatar/Accessories/Head/PorkpieHat/PorkpieHat"
```

Field Value

[string](#) ↗

RectangleFrameGlasses

```
public const string RectangleFrameGlasses =  
"Avatar/Accessories/Head/RectangleFrameGlasses/RectangleFrameGlasses"
```

Field Value

[string ↗](#)

Respirator

```
public const string Respirator = "Avatar/Accessories/Head/Respirator/Respirator"
```

Field Value

[string ↗](#)

SantaHat

```
public const string SantaHat = "Avatar/Accessories/Head/SantaHat/SantaHat"
```

Field Value

[string ↗](#)

SaucePan

```
public const string SaucePan = "Avatar/Accessories/Head/SaucePan/SaucePan"
```

Field Value

[string](#)

SmallRoundGlasses

```
public const string SmallRoundGlasses =  
"Avatar/Accessories/Head/SmallRoundGlasses/SmallRoundGlasses"
```

Field Value

[string](#)

TrashCrown

```
public const string TrashCrown = "Avatar/Accessories/Head/TrashCrown/TrashCrown"
```

Field Value

[string](#)

Class Neck

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Neck index in AvatarSettings::AccessorySettings

```
public class Neck : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Neck

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Neck()

```
public Neck()
```

Fields

GoldChain

```
public const string GoldChain = "Avatar/Accessories/Neck/GoldChain/GoldChain"
```

Field Value

[string](#) ↗

Class Waist

Namespace: [S1API.Entities.Appearances.AccessoryFields](#)

Assembly: S1API.dll

The Waist index in AvatarSettings::AccessorySettings

```
public class Waist : BaseAccessoryAppearance
```

Inheritance

[object](#) ← [BaseAccessoryAppearance](#) ← Waist

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Waist()

```
public Waist()
```

Fields

Apron

```
public const string Apron = "Avatar/Accessories/Waist/Apron/Apron"
```

Field Value

[string](#)

Belt

```
public const string Belt = "Avatar/Accessories/Waist/Belt/Belt"
```

Field Value

[string](#)

HazmatSuit

```
public const string HazmatSuit = "Avatar/Accessories/Waist/HazmatSuit/HazmatSuit"
```

Field Value

[string](#)

PoliceBelt

```
public const string PoliceBelt = "Avatar/Accessories/Waist/PoliceBelt/PoliceBelt"
```

Field Value

[string](#)

PriestGown

```
public const string PriestGown = "Avatar/Accessories/Waist/PriestGown/PriestGown"
```

Field Value

[string](#) 

Namespace S1API.Entities.Appearances.Base

Classes

[BaseAccessoryAppearance](#)

The base accessory appearance class

[BaseAppearance](#)

The base appearance class

[BaseBodyAppearance](#)

The base body appearance class

[BaseFaceAppearance](#)

The base face appearance class

Class BaseAccessoryAppearance

Namespace: [S1API.Entities.Appearances.Base](#)

Assembly: S1API.dll

The base accessory appearance class

```
public class BaseAccessoryAppearance
```

Inheritance

[object](#) ← BaseAccessoryAppearance

Derived

[Bottom](#), [Chest](#), [FacialHairAccessory](#), [Feet](#), [Hands](#), [Head](#), [Neck](#), [Waist](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This is used to track the properties within the AvatarSettings

Constructors

BaseAccessoryAppearance()

```
public BaseAccessoryAppearance()
```

Class BaseAppearance

Namespace: [S1API.Entities.Appearances.Base](#)

Assembly: S1API.dll

The base appearance class

```
public class BaseAppearance
```

Inheritance

[object](#) ← BaseAppearance

Derived

[EyeBallTint](#), [EyeLidRestingStateLeft](#), [EyeLidRestingStateRight](#), [EyebrowRestingAngle](#),
[EyebrowRestingHeight](#), [EyebrowScale](#), [EyebrowThickness](#), [Gender](#), [HairColor](#), [HairStyle](#), [Height](#),
[PupilDilation](#), [SkinColor](#), [Weight](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This is used to track the properties within the AvatarSettings

Constructors

BaseAppearance()

```
public BaseAppearance()
```

Class BaseBodyAppearance

Namespace: [S1API.Entities.Appearances.Base](#)

Assembly: S1API.dll

The base body appearance class

```
public class BaseBodyAppearance
```

Inheritance

[object](#) ← BaseBodyAppearance

Derived

[Accessories](#), [ChestTattoos](#), [LeftArmTattoos](#), [Pants](#), [RightArmTattoos](#), [Shirts](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This is used to track the properties within the AvatarSettings

Constructors

BaseBodyAppearance()

```
public BaseBodyAppearance()
```

Class BaseFaceAppearance

Namespace: [S1API.Entities.Appearances.Base](#)

Assembly: S1API.dll

The base face appearance class

```
public class BaseFaceAppearance
```

Inheritance

[object](#) ← BaseFaceAppearance

Derived

[Eyes](#), [Face](#), [FaceTattoos](#), [FacialHair](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This is used to track the properties within the AvatarSettings

Constructors

BaseFaceAppearance()

```
public BaseFaceAppearance()
```

Namespace S1API.Entities.Appearances.BodyLayerFields

Classes

[Accessories](#)

The Accessories (Gloves) index in AvatarSettings BodyLayerSettings

[ChestTattoos](#)

The Chest Tattoos index in AvatarSettings BodyLayerSettings

[LeftArmTattoos](#)

The Left Arm Tattoos index in AvatarSettings BodyLayerSettings

[Pants](#)

The Bottom index in AvatarSettings

[RightArmTattoos](#)

The Right Arm Tattoos index in AvatarSettings BodyLayerSettings

[Shirts](#)

The Top index in AvatarSettings

Class Accessories

Namespace: [S1API.Entities.Appearances.BodyLayerFields](#)

Assembly: S1API.dll

The Accessories (Gloves) index in AvatarSettings BodyLayerSettings

```
public class Accessories : BaseBodyAppearance
```

Inheritance

[object](#) ← [BaseBodyAppearance](#) ← Accessories

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Accessories()

```
public Accessories()
```

Fields

FingerlessGloves

```
public const string FingerlessGloves = "Avatar/Layers/Accessories/FingerlessGloves"
```

Field Value

[string](#)

Gloves

```
public const string Gloves = "Avatar/Layers/Accessories/Gloves"
```

Field Value

[string](#)

Class ChestTattoos

Namespace: [S1API.Entities.Appearances.BodyLayerFields](#)

Assembly: S1API.dll

The Chest Tattoos index in AvatarSettings BodyLayerSettings

```
public class ChestTattoos : BaseBodyAppearance
```

Inheritance

[object](#) ← [BaseBodyAppearance](#) ← ChestTattoos

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ChestTattoos()

```
public ChestTattoos()
```

Fields

Bird

```
public const string Bird = "Avatar/Layers/Tattoos/chest/Chest_Bird"
```

Field Value

[string](#)

DeadFace

```
public const string DeadFace = "Avatar/Layers/Tattoos/chest/Chest_DeadFace"
```

Field Value

[string](#)

Egg

```
public const string Egg = "Avatar/Layers/Tattoos/chest/Chest_Egg"
```

Field Value

[string](#)

LBC

```
public const string LBC = "Avatar/Layers/Tattoos/chest/Chest_LBC"
```

Field Value

[string](#)

Sword

```
public const string Sword = "Avatar/Layers/Tattoos/chest/Chest_Sword"
```

Field Value

[string](#) 

Class LeftArmTattoos

Namespace: [S1API.Entities.Appearances.BodyLayerFields](#)

Assembly: S1API.dll

The Left Arm Tattoos index in AvatarSettings BodyLayerSettings

```
public class LeftArmTattoos : BaseBodyAppearance
```

Inheritance

[object](#) ← [BaseBodyAppearance](#) ← LeftArmTattoos

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

LeftArmTattoos()

```
public LeftArmTattoos()
```

Fields

Alien

```
public const string Alien = "Avatar/Layers/Tattoos/leftarm/LeftArm_Alien"
```

Field Value

[string](#)

Heart

```
public const string Heart = "Avatar/Layers/Tattoos/leftarm/LeftArm_Heart"
```

Field Value

[string](#)

Peace

```
public const string Peace = "Avatar/Layers/Tattoos/leftarm/LeftArm_Peace"
```

Field Value

[string](#)

Web

```
public const string Web = "Avatar/Layers/Tattoos/leftarm/LeftArm_Web"
```

Field Value

[string](#)

Weed

```
public const string Weed = "Avatar/Layers/Tattoos/leftarm/LeftArm_Weed"
```

Field Value

[string](#) 

Class Pants

Namespace: [S1API.Entities.Appearances.BodyLayerFields](#)

Assembly: S1API.dll

The Bottom index in AvatarSettings

```
public class Pants : BaseBodyAppearance
```

Inheritance

[object](#) ← [BaseBodyAppearance](#) ← Pants

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Pants()

```
public Pants()
```

Fields

CargoPants

```
public const string CargoPants = "Avatar/Layers/Bottom/CargoPants"
```

Field Value

[string](#)

FemaleUnderwear

```
public const string FemaleUnderwear = "Avatar/Layers/Bottom/FemaleUnderwear"
```

Field Value

[string](#)

Jeans

```
public const string Jeans = "Avatar/Layers/Bottom/Jeans"
```

Field Value

[string](#)

Jorts

```
public const string Jorts = "Avatar/Layers/Bottom/Jorts"
```

Field Value

[string](#)

MaleUnderwear

```
public const string MaleUnderwear = "Avatar/Layers/Bottom/MaleUnderwear"
```

Field Value

[string](#) 

Class RightArmTattoos

Namespace: [S1API.Entities.Appearances.BodyLayerFields](#)

Assembly: S1API.dll

The Right Arm Tattoos index in AvatarSettings BodyLayerSettings

```
public class RightArmTattoos : BaseBodyAppearance
```

Inheritance

[object](#) ← [BaseBodyAppearance](#) ← RightArmTattoos

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

RightArmTattoos()

```
public RightArmTattoos()
```

Fields

Alien

```
public const string Alien = "Avatar/Layers/Tattoos/rightarm/RightArm_Alien"
```

Field Value

[string](#)

Heart

```
public const string Heart = "Avatar/Layers/Tattoos/rightarm/RightArm_Heart"
```

Field Value

[string](#)

Peace

```
public const string Peace = "Avatar/Layers/Tattoos/rightarm/RightArm_Peace"
```

Field Value

[string](#)

Web

```
public const string Web = "Avatar/Layers/Tattoos/rightarm/RightArm_Web"
```

Field Value

[string](#)

Weed

```
public const string Weed = "Avatar/Layers/Tattoos/rightarm/RightArm_Weed"
```

Field Value

[string](#) 

Class Shirts

Namespace: [S1API.Entities.Appearances.BodyLayerFields](#)

Assembly: S1API.dll

The Top index in AvatarSettings

```
public class Shirts : BaseBodyAppearance
```

Inheritance

[object](#) ← [BaseBodyAppearance](#) ← Shirts

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Shirts()

```
public Shirts()
```

Fields

Buttonup

```
public const string Buttonup = "Avatar/Layers/Top/Buttonup"
```

Field Value

[string](#)

ChestHair

```
public const string ChestHair = "Avatar/Layers/Top/ChestHair1"
```

Field Value

[string](#)

FastFoodTShirt

```
public const string FastFoodTShirt = "Avatar/Layers/Top/FastFood T-Shirt"
```

Field Value

[string](#)

FlannelButtonUp

```
public const string FlannelButtonUp = "Avatar/Layers/Top/FlannelButtonUp"
```

Field Value

[string](#)

GasStationTShirt

```
public const string GasStationTShirt = "Avatar/Layers/Top/GasStation T-Shirt"
```

Field Value

[string](#)

HazmatSuit

```
public const string HazmatSuit = "Avatar/Layers/Top/HazmatSuit"
```

Field Value

[string](#)

Nipples

```
public const string Nipples = "Avatar/Layers/Top/Nipples"
```

Field Value

[string](#)

Overalls

```
public const string Overalls = "Avatar/Layers/Top/Overalls"
```

Field Value

[string](#)

RolledButtonUp

```
public const string RolledButtonUp = "Avatar/Layers/Top/RolledButtonUp"
```

Field Value

[string](#) ↗

TShirt

```
public const string TShirt = "Avatar/Layers/Top/T-Shirt"
```

Field Value

[string](#) ↗

TuckedTShirt

```
public const string TuckedTShirt = "Avatar/Layers/Top/Tucked T-Shirt"
```

Field Value

[string](#) ↗

UpperBodyTattoos

```
public const string UpperBodyTattoos = "Avatar/Layers/Top/UpperBodyTattoos"
```

Field Value

[string](#) ↗

VNeck

```
public const string VNeck = "Avatar/Layers/Top/V-Neck"
```

Field Value

[string ↗](#)

Namespace S1API.Entities.Appearances. CustomizationFields

Classes

[EyeBallTint](#)

The EyeBallTint field in AvatarSettings

[EyeLidRestingStateLeft](#)

The LeftEyeRestingState field in AvatarSettings

[EyeLidRestingStateRight](#)

The RightEyeRestingState field in AvatarSettings

[EyebrowRestingAngle](#)

The EyebrowRestingAngle field in AvatarSettings

[EyebrowRestingHeight](#)

The EyebrowRestingHeight field in AvatarSettings

[EyebrowScale](#)

The EyebrowScale field in AvatarSettings

[EyebrowThickness](#)

The EyebrowThickness field in AvatarSettings

[Gender](#)

The Gender field in AvatarSettings

[HairColor](#)

The HairColor field in AvatarSettings

[HairStyle](#)

The HairPath field in AvatarSettings

[Height](#)

The Height field in AvatarSettings

[PupilDilation](#)

The PupilDilation field in AvatarSettings

[SkinColor](#)

The SkinColor field in AvatarSettings

Weight

The Weight field in AvatarSettings

Class EyeBallTint

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The EyeBallTint field in AvatarSettings

```
public class EyeBallTint : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyeBallTint

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a Color type

Constructors

EyeBallTint()

```
public EyeBallTint()
```

Class EyeLidRestingStateLeft

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The LeftEyeRestingState field in AvatarSettings

```
public class EyeLidRestingStateLeft : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyeLidRestingStateLeft

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a (float, float) type

Constructors

EyeLidRestingStateLeft()

```
public EyeLidRestingStateLeft()
```

Class EyeLidRestingStateRight

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The RightEyeRestingState field in AvatarSettings

```
public class EyeLidRestingStateRight : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyeLidRestingStateRight

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a (float, float) type

Constructors

[EyeLidRestingStateRight\(\)](#)

```
public EyeLidRestingStateRight()
```

Class EyebrowRestingAngle

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The EyebrowRestingAngle field in AvatarSettings

```
public class EyebrowRestingAngle : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyebrowRestingAngle

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

EyebrowRestingAngle()

```
public EyebrowRestingAngle()
```

Class EyebrowRestingHeight

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The EyebrowRestingHeight field in AvatarSettings

```
public class EyebrowRestingHeight : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyebrowRestingHeight

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

EyebrowRestingHeight()

```
public EyebrowRestingHeight()
```

Class EyebrowScale

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The EyebrowScale field in AvatarSettings

```
public class EyebrowScale : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyebrowScale

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

EyebrowScale()

```
public EyebrowScale()
```

Class EyebrowThickness

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The EyebrowThickness field in AvatarSettings

```
public class EyebrowThickness : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← EyebrowThickness

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

EyebrowThickness()

```
public EyebrowThickness()
```

Class Gender

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The Gender field in AvatarSettings

```
public class Gender : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← Gender

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

Gender()

```
public Gender()
```

Class HairColor

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The HairColor field in AvatarSettings

```
public class HairColor : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← HairColor

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a Color type

Constructors

HairColor()

```
public HairColor()
```

Class HairStyle

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The HairPath field in AvatarSettings

```
public class HairStyle : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← HairStyle

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This contains all available hairstyles.

Is a string type.

Constructors

HairStyle()

```
public HairStyle()
```

Fields

Afro

```
public const string Afro = "Avatar/Hair/afro/Afro"
```

Field Value

[string ↗](#)

Balding

```
public const string Balding = "Avatar/Hair/balding/Balding"
```

Field Value

[string ↗](#)

BowlCut

```
public const string BowlCut = "Avatar/Hair/bowlcut/BowlCut"
```

Field Value

[string ↗](#)

Bun

```
public const string Bun = "Avatar/Hair/bun/Bun"
```

Field Value

[string ↗](#)

BuzzCut

```
public const string BuzzCut = "Avatar/Hair/buzzcut/BuzzCut"
```

Field Value

[string ↗](#)

CloseBuzzCut

```
public const string CloseBuzzCut = "Avatar/Hair/closebuzzcut/CloseBuzzCut"
```

Field Value

[string ↗](#)

DoubleTopKnot

```
public const string DoubleTopKnot = "Avatar/Hair/doubletopknot/DoubleTopKnot"
```

Field Value

[string ↗](#)

Franklin

```
public const string Franklin = "Avatar/Hair/franklin/Franklin"
```

Field Value

[string](#)

FringePonyTail

```
public const string FringePonyTail = "Avatar/Hair/fringeponytail/FringePonyTail"
```

Field Value

[string](#)

HighBun

```
public const string HighBun = "Avatar/Hair/highbun/HighBun"
```

Field Value

[string](#)

Jesus

```
public const string Jesus = "Avatar/Hair/jesus/Jesus"
```

Field Value

[string](#)

LongCurly

```
public const string LongCurly = "Avatar/Hair/longcurly/LongCurly"
```

Field Value

[string](#)

LowBun

```
public const string LowBun = "Avatar/Hair/lowbun/LowBun"
```

Field Value

[string](#)

MessyBob

```
public const string MessyBob = "Avatar/Hair/messybob/MessyBob"
```

Field Value

[string](#)

MidFringe

```
public const string MidFringe = "Avatar/Hair/midfringe/MidFringe"
```

Field Value

[string](#)

Mohawk

```
public const string Mohawk = "Avatar/Hair/mohawk/Mohawk"
```

Field Value

[string](#) ↗

Monk

```
public const string Monk = "Avatar/Hair/monk/Monk"
```

Field Value

[string](#) ↗

Peaked

```
public const string Peaked = "Avatar/Hair/peaked/Peaked"
```

Field Value

[string](#) ↗

Receding

```
public const string Receding = "Avatar/Hair/receding/Receding"
```

Field Value

[string](#) ↗

ShoulderLength

```
public const string ShoulderLength = "Avatar/Hair/shoulderlength/ShoulderLength"
```

Field Value

[string](#) ↗

SidePartBob

```
public const string SidePartBob = "Avatar/Hair/sidepartbob/SidePartBob"
```

Field Value

[string](#) ↗

Spiky

```
public const string Spiky = "Avatar/Hair/spiky/Spiky"
```

Field Value

[string](#) ↗

Tony

```
public const string Tony = "Avatar/Hair/tony/Tony"
```

Field Value

[string](#) ↗

Class Height

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The Height field in AvatarSettings

```
public class Height : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← Height

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

Height()

```
public Height()
```

Class PupilDilation

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The PupilDilation field in AvatarSettings

```
public class PupilDilation : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← PupilDilation

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

PupilDilation()

```
public PupilDilation()
```

Class SkinColor

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The SkinColor field in AvatarSettings

```
public class SkinColor : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← SkinColor

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a Color type

Constructors

SkinColor()

```
public SkinColor()
```

Class Weight

Namespace: [S1API.Entities.Appearances.CustomizationFields](#)

Assembly: S1API.dll

The Weight field in AvatarSettings

```
public class Weight : BaseAppearance
```

Inheritance

[object](#) ← [BaseAppearance](#) ← Weight

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Is a float type

Constructors

Weight()

```
public Weight()
```

Namespace S1API.Entities.Appearances.FaceLayerFields

Classes

[Eyes](#)

The EyeShadow index in AvatarSettings

[Face](#)

The Face index in FaceLayerSettings

[FaceTattoos](#)

The Face Tattoos index in AvatarSettings FaceLayerSettings

[FacialHair](#)

The FacialHair index in FaceLayerSettings

Class Eyes

Namespace: [S1API.Entities.Appearances.FaceLayerFields](#)

Assembly: S1API.dll

The EyeShadow index in AvatarSettings

```
public class Eyes : BaseFaceAppearance
```

Inheritance

[object](#) ← [BaseFaceAppearance](#) ← Eyes

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Eyes()

```
public Eyes()
```

Fields

EyeShadow

```
public const string EyeShadow = "Avatar/Layers/Face/EyeShadow"
```

Field Value

[string](#)

Freckles

```
public const string Freckles = "Avatar/Layers/Face/Freckles"
```

Field Value

[string](#)

OldPersonWrinkles

```
public const string OldPersonWrinkles = "Avatar/Layers/Face/OldPersonWrinkles"
```

Field Value

[string](#)

TiredEyes

```
public const string TiredEyes = "Avatar/Layers/Face/TiredEyes"
```

Field Value

[string](#)

Class Face

Namespace: [S1API.Entities.Appearances.FaceLayerFields](#)

Assembly: S1API.dll

The Face index in FaceLayerSettings

```
public class Face : BaseFaceAppearance
```

Inheritance

[object](#) ← [BaseFaceAppearance](#) ← Face

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Face()

```
public Face()
```

Fields

Agape

```
public const string Agape = "Avatar/Layers/Face/Face_Agape"
```

Field Value

[string](#)

Agitated

```
public const string Agitated = "Avatar/Layers/Face/Face_Agitated"
```

Field Value

[string](#)

FaceTattoos1

```
public const string FaceTattoos1 = "Avatar/Layers/Face/FaceTattoos1"
```

Field Value

[string](#)

FrownPout

```
public const string FrownPout = "Avatar/Layers/Face/Face_FrownPout"
```

Field Value

[string](#)

Neutral

```
public const string Neutral = "Avatar/Layers/Face/Face_Neutral"
```

Field Value

[string](#)

NeutralPout

```
public const string NeutralPout = "Avatar/Layers/Face/Face_NeutralPout"
```

Field Value

[string](#)

OpenMouthSmile

```
public const string OpenMouthSmile = "Avatar/Layers/Face/Face_OpenMouthSmile"
```

Field Value

[string](#)

Scared

```
public const string Scared = "Avatar/Layers/Face/Face_Scared"
```

Field Value

[string](#)

SlightFrown

```
public const string SlightFrown = "Avatar/Layers/Face/Face_SlightFrown"
```

Field Value

[string](#) ↗

SlightSmile

```
public const string SlightSmile = "Avatar/Layers/Face/Face_SlightSmile"
```

Field Value

[string](#) ↗

Smile

```
public const string Smile = "Avatar/Layers/Face/Face_Smile"
```

Field Value

[string](#) ↗

SmugPout

```
public const string SmugPout = "Avatar/Layers/Face/Face_SmugPout"
```

Field Value

[string](#) ↗

Surprised

```
public const string Surprised = "Avatar/Layers/Face/Face_Surprised"
```

Field Value

[string ↗](#)

Class FaceTattoos

Namespace: [S1API.Entities.Appearances.FaceLayerFields](#)

Assembly: S1API.dll

The Face Tattoos index in AvatarSettings FaceLayerSettings

```
public class FaceTattoos : BaseFaceAppearance
```

Inheritance

[object](#) ← [BaseFaceAppearance](#) ← FaceTattoos

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

FaceTattoos()

```
public FaceTattoos()
```

Fields

ForeheadCross

```
public const string ForeheadCross = "Avatar/Layers/Tattoos/face/Face_ForeheadCross"
```

Field Value

[string](#)

Sword

```
public const string Sword = "Avatar/Layers/Tattoos/face/Face_Sword"
```

Field Value

[string](#)

Teardrop

```
public const string Teardrop = "Avatar/Layers/Tattoos/face/Face_Teardrop"
```

Field Value

[string](#)

Tribal

```
public const string Tribal = "Avatar/Layers/Tattoos/face/Face_Tribal"
```

Field Value

[string](#)

Class FacialHair

Namespace: [S1API.Entities.Appearances.FaceLayerFields](#)

Assembly: S1API.dll

The FacialHair index in FaceLayerSettings

```
public class FacialHair : BaseFaceAppearance
```

Inheritance

[object](#) ← [BaseFaceAppearance](#) ← FacialHair

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

FacialHair()

```
public FacialHair()
```

Fields

Goatee

```
public const string Goatee = "Avatar/Layers/Face/FacialHair_Goatee"
```

Field Value

[string](#)

Stubble

```
public const string Stubble = "Avatar/Layers/Face/FacialHair_Stubble"
```

Field Value

[string](#)

Swirl

```
public const string Swirl = "Avatar/Layers/Face/FacialHair_Swirl"
```

Field Value

[string](#)

Namespace S1API.Entities.Behaviour

Classes

[CombatBehaviour](#)

Represents the combat behaviour of an NPC, allowing configuration of combat-related settings and actions.

Class CombatBehaviour

Namespace: [S1API.Entities.Behaviour](#)

Assembly: S1API.dll

Represents the combat behaviour of an NPC, allowing configuration of combat-related settings and actions.

```
public class CombatBehaviour
```

Inheritance

[object](#) ← CombatBehaviour

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

DefaultWeaponAssetPath

Gets or sets the default weapon asset path for the NPC's combat behaviour. This property allows you to specify the weapon that the NPC will use by default. [Weapon](#) for convenience when setting this property.

```
public string DefaultWeaponAssetPath { get; set; }
```

Property Value

[string](#)

GiveUpRange

Gets or sets the range in units at which the NPC will give up pursuing a target.

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

Property Value

[float](#)

GiveUpTime

Gets or sets the time in seconds the NPC will continue to pursue a target before giving up.

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

Property Value

[float](#)

Methods

SetAndAttackTarget(IEntity)

Sets the specified target as the NPC's combat target and enables attacking behavior.

```
public void SetAndAttackTarget(IEntity target)
```

Parameters

target [IEntity](#)

The target entity to be attacked by the NPC.

SetCurrentWeapon(Equippable)

Sets the current weapon of the NPC using an S1API Equippable wrapper. This method extracts the asset path from the equippable's associated AvatarEquippable component.

```
public void SetCurrentWeapon(Equippable equippable)
```

Parameters

equippable [Equippable](#)

The S1API Equippable wrapper containing the weapon to equip.

Remarks

This method attempts to find an AvatarEquippable component on the equippable's GameObject and uses its AssetPath. If no AvatarEquippable is found, this method will log an error.

SetCurrentWeapon(EquippableBuilder)

Sets the current weapon of the NPC using an EquippableBuilder-created equippable. This method extracts the asset path from the builder's AvatarEquippable configuration.

```
public void SetCurrentWeapon(EquippableBuilder equippableBuilder)
```

Parameters

equippableBuilder [EquippableBuilder](#)

The EquippableBuilder instance containing the weapon configuration.

Remarks

This method requires the EquippableBuilder to have been configured with WithAvatarEquippable() to provide an asset path. The builder will be built automatically if not already built.

SetCurrentWeapon(string)

Sets the current weapon of the NPC by specifying the weapon's resource path.

```
public void SetCurrentWeapon(string weaponPath)
```

Parameters

weaponPath [string](#)

The resource path of the weapon to equip (e.g., "Avatar/Equippables/M1911"). For convenience, [Weapon](#) can be used.

SetDefaultWeapon(Equippable)

Sets the default weapon for the NPC's combat behaviour using an S1API Equippable wrapper. This method extracts the asset path from the equippable's associated AvatarEquippable component.

```
public void SetDefaultWeapon(Equippable equippable)
```

Parameters

equippable [Equippable](#)

The S1API Equippable wrapper containing the weapon to set as default.

Remarks

This method attempts to find an AvatarEquippable component on the equippable's GameObject and uses its AssetPath. If no AvatarEquippable is found, this method will log an error.

SetDefaultWeapon(EquippableBuilder)

Sets the default weapon for the NPC's combat behaviour using an EquippableBuilder-created equippable. This method extracts the asset path from the builder's AvatarEquippable configuration.

```
public void SetDefaultWeapon(EquippableBuilder equippableBuilder)
```

Parameters

equippableBuilder [EquippableBuilder](#)

The EquippableBuilder instance containing the weapon configuration.

Remarks

This method requires the EquippableBuilder to have been configured with WithAvatarEquippable() to provide an asset path. The builder will be built automatically if not already built.

Namespace S1API.Entities.Customer

Classes

[CustomerDataBuilder](#)

Builder for composing CustomerData at runtime without asset bundles. Public surface uses strings/primitives only.

Class CustomerDataBuilder

Namespace: [S1API.Entities.Customer](#)

Assembly: S1API.dll

Builder for composing CustomerData at runtime without asset bundles. Public surface uses strings/primitives only.

```
public sealed class CustomerDataBuilder
```

Inheritance

[object](#) ← CustomerDataBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

AllowDirectApproach(bool)

```
public CustomerDataBuilder AllowDirectApproach(bool allow)
```

Parameters

allow [bool](#)

Returns

[CustomerDataBuilder](#)

GuaranteeFirstSample(bool)

```
public CustomerDataBuilder GuaranteeFirstSample(bool guarantee)
```

Parameters

guarantee [bool](#)

Returns

[CustomerDataBuilder](#)

WithAffinities(IEnumerable<(DrugType drugType, float affinity)>)

Sets product type affinities using the enum. Replaces any existing default affinity data. Ensures all drug types are initialized (unspecified ones default to neutral affinity).

```
public CustomerDataBuilder WithAffinities(IEnumerable<(DrugType drugType, float affinity)> entries)
```

Parameters

entries [IEnumerable](#)<[DrugType](#) [drugType](#), [float](#) [affinity](#)>

Returns

[CustomerDataBuilder](#)

WithAffinities(IEnumerable<(string drugType, float affinity)>)

Sets product type affinities by drug-type name. e.g., ("Weed", 0.3f), ("Cocaine", -0.5f). Ensures all drug types are initialized (unspecified ones default to neutral affinity).

```
public CustomerDataBuilder WithAffinities(IEnumerable<string drugType, float affinity> entries)
```

Parameters

entries [IEnumerable<string, float>](#)

Returns

[CustomerDataBuilder](#)

WithAffinity(DrugType, float)

Adds or overrides a single product type affinity entry.

```
public CustomerDataBuilder WithAffinity(DrugType drugType, float affinity)
```

Parameters

drugType [DrugType](#)

affinity [float](#)

Returns

[CustomerDataBuilder](#)

WithCallPoliceChance(float)

```
public CustomerDataBuilder WithCallPoliceChance(float chance)
```

Parameters

chance [float](#)

Returns

[CustomerDataBuilder](#)

WithDependence(float, float)

```
public CustomerDataBuilder WithDependence(float baseAddiction, float  
dependenceMultiplier = 1)
```

Parameters

baseAddiction [float](#)

dependenceMultiplier [float](#)

Returns

[CustomerDataBuilder](#)

WithMutualRelationRequirement(float, float)

```
public CustomerDataBuilder WithMutualRelationRequirement(float minAt50,  
float maxAt100)
```

Parameters

minAt50 [float](#)

maxAt100 [float](#)

Returns

[CustomerDataBuilder](#)

WithOrderTime(int)

Sets the order time in 24h integer format (e.g., 930 for 9:30AM, 1745 for 5:45PM).

```
public CustomerDataBuilder WithOrderTime(int hhmm)
```

Parameters

hhmm [int](#)

Returns

[CustomerDataBuilder](#)

WithOrdersPerWeek(int, int)

```
public CustomerDataBuilder WithOrdersPerWeek(int min, int max)
```

Parameters

min [int](#)

max [int](#)

Returns

[CustomerDataBuilder](#)

WithPreferredOrderDay(Day)

Sets the preferred order day using the API Day enum.

```
public CustomerDataBuilder WithPreferredOrderDay(Day day)
```

Parameters

day [Day](#)

Returns

[CustomerDataBuilder](#)

WithPreferredOrderDay(string)

Sets the preferred order day using names like "Monday", "Tuesday", ... or numeric 0..6.

```
public CustomerDataBuilder WithPreferredOrderDay(string day)
```

Parameters

day [string](#)

Returns

[CustomerDataBuilder](#)

WithPreferredProperties(params PropertyBase[])

Assigns preferred properties from API property abstractions (tokens or wrappers). Does not expose game types to mod developers.

```
public CustomerDataBuilder WithPreferredProperties(params PropertyBase[] properties)
```

Parameters

properties [PropertyBase\[\]](#)

Returns

[CustomerDataBuilder](#)

WithPreferredProperties(params ProductPropertyWrapper[])

Assigns preferred properties from wrappers.

```
public CustomerDataBuilder WithPreferredProperties(params ProductPropertyWrapper[]  
wrappers)
```

Parameters

wrappers [ProductPropertyWrapper\[\]](#)

Returns

[CustomerDataBuilder](#)

WithPreferredPropertiesById(params string[])

Tries to assign preferred properties by ID using in-game Resources. IDs must match existing property assets.

```
public CustomerDataBuilder WithPreferredPropertiesById(params string[] propertyIds)
```

Parameters

propertyIds [string\[\]](#)

Returns

[CustomerDataBuilder](#)

WithPreferredPropertiesByName(params string[])

Tries to assign preferred properties by asset name using in-game Resources. Names must match existing property assets.

```
public CustomerDataBuilder WithPreferredPropertiesByName(params  
string[] propertyNames)
```

Parameters

propertyNames [string](#)[]

Returns

[CustomerDataBuilder](#)

WithSpending(float, float)

```
public CustomerDataBuilder WithSpending(float minWeekly, float maxWeekly)
```

Parameters

minWeekly [float](#)

maxWeekly [float](#)

Returns

[CustomerDataBuilder](#)

WithStandards(CustomerStandard)

Sets standards using the strongly-typed enum.

```
public CustomerDataBuilder WithStandards(CustomerStandard standards)
```

Parameters

standards [CustomerStandard](#)

Returns

[CustomerDataBuilder](#)

WithStandards(string)

Sets standards from string (e.g., "VeryLow", "Low", "Moderate", "High", "VeryHigh").

```
public CustomerDataBuilder WithStandards(string standards)
```

Parameters

standards [string](#) ↗

Returns

[CustomerDataBuilder](#)

Namespace S1API.Entities.Dealer

Classes

[DealerDataBuilder](#)

Builder for composing dealer configuration at runtime without asset bundles. Public surface uses strings/primitives only.

Class DealerDataBuilder

Namespace: [S1API.Entities.Dealer](#)

Assembly: S1API.dll

Builder for composing dealer configuration at runtime without asset bundles. Public surface uses strings/primitives only.

```
public sealed class DealerDataBuilder
```

Inheritance

[object](#) ← DealerDataBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

AllowExcessQuality(bool)

Allows the dealer to sell items above the customer's quality standards.

```
public DealerDataBuilder AllowExcessQuality(bool allow)
```

Parameters

allow [bool](#)

Returns

[DealerDataBuilder](#)

AllowInsufficientQuality(bool)

Allows the dealer to sell items below the customer's quality standards.

```
public DealerDataBuilder AllowInsufficientQuality(bool allow)
```

Parameters

allow [bool](#)

Returns

[DealerDataBuilder](#)

WithCompletedDealsVariable(string)

Sets the variable name to track completed deals for this dealer.

```
public DealerDataBuilder WithCompletedDealsVariable(string varName)
```

Parameters

varName [string](#)

Returns

[DealerDataBuilder](#)

WithCut(float)

Sets the dealer's commission cut (percentage of earnings they keep). Range: 0.0 to 1.0.

```
public DealerDataBuilder WithCut(float percentage)
```

Parameters

percentage [float](#)

Returns

[DealerDataBuilder](#)

WithDealerType(DealerType)

Sets the dealer type (PlayerDealer or CartelDealer).

```
public DealerDataBuilder WithDealerType(DealerType type)
```

Parameters

type [DealerType](#)

Returns

[DealerDataBuilder](#)

WithHome(Building)

Sets the home building for this dealer using a Building wrapper.

```
public DealerDataBuilder WithHome(Building building)
```

Parameters

building [Building](#)

The Building wrapper for the dealer's home.

Returns

[DealerDataBuilder](#)

WithHomeName(string)

Sets the home building name for this dealer.

```
public DealerDataBuilder WithHomeName(string name)
```

Parameters

name [string](#)

Returns

[DealerDataBuilder](#)

WithSigningFee(float)

Sets the signing fee required to recruit this dealer.

```
public DealerDataBuilder WithSigningFee(float fee)
```

Parameters

fee [float](#)

Returns

[DealerDataBuilder](#)

Namespace S1API.Entities.Dialogue

Classes

[DialogueContainerBuilder](#)

Builder to compose a choice-based DialogueContainer entirely from code. Public surface uses only strings and primitives; no game types are exposed.

[DialogueContainerBuilder.ChoiceList](#)

Helper class used during choice configuration.

[DialogueDatabaseBuilder](#)

Builder for composing an NPC-specific dialogue database without requiring asset bundles. Public surface only accepts strings and arrays; no game types exposed.

Class DialogueContainerBuilder

Namespace: [S1API.Entities.Dialogue](#)

Assembly: S1API.dll

Builder to compose a choice-based DialogueContainer entirely from code. Public surface uses only strings and primitives; no game types are exposed.

```
public sealed class DialogueContainerBuilder
```

Inheritance

[object](#) ← DialogueContainerBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DialogueContainerBuilder()

```
public DialogueContainerBuilder()
```

Methods

AddNode(string, string, Action<ChoiceList>)

Adds a dialogue node by label with the text shown in the bubble/UI.

```
public DialogueContainerBuilder AddNode(string nodeLabel, string text,
```

```
Action<DialogueContainerBuilder.ChoiceList> choices = null)
```

Parameters

nodeLabel [string](#)

text [string](#)

choices [Action](#)<[DialogueContainerBuilder.ChoiceList](#)>

Returns

[DialogueContainerBuilder](#)

SetAllowExit(bool)

Sets whether the player can exit while this container is active.

```
public DialogueContainerBuilder SetAllowExit(bool allow)
```

Parameters

allow [bool](#)

Returns

[DialogueContainerBuilder](#)

Class DialogueContainerBuilder.ChoiceList

Namespace: [S1API.Entities.Dialogue](#)

Assembly: S1API.dll

Helper class used during choice configuration.

```
public sealed class DialogueContainerBuilder.ChoiceList
```

Inheritance

[object](#) ← DialogueContainerBuilder.ChoiceList

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

Add(string, string, string)

Adds a choice with a label and shown text and links it to a target node label.

```
public DialogueContainerBuilder.ChoiceList Add(string choiceLabel, string shownText,  
string targetNodeLabel = null)
```

Parameters

choiceLabel [string](#)

shownText [string](#)

targetNodeLabel [string](#)

Returns

[DialogueContainerBuilder.ChoiceList](#)

Class DialogueDatabaseBuilder

Namespace: [S1API.Entities.Dialogue](#)

Assembly: S1API.dll

Builder for composing an NPC-specific dialogue database without requiring asset bundles. Public surface only accepts strings and arrays; no game types exposed.

```
public sealed class DialogueDatabaseBuilder
```

Inheritance

[object](#) ← DialogueDatabaseBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DialogueDatabaseBuilder()

```
public DialogueDatabaseBuilder()
```

Methods

WithGeneric(string, params string[])

Adds a generic entry (available under the Generic module).

```
public DialogueDatabaseBuilder WithGeneric(string key, params string[] lines)
```

Parameters

key [string](#)

lines [string](#)[]

Returns

[DialogueDatabaseBuilder](#)

WithModuleEntry(string, string, params string[])

Adds an entry to a named module (e.g., "Reactions", "SmallTalk").

```
public DialogueDatabaseBuilder WithModuleEntry(string moduleName, string key, params  
string[] lines)
```

Parameters

moduleName [string](#)

key [string](#)

lines [string](#)[]

Returns

[DialogueDatabaseBuilder](#)

Namespace S1API.Entities.Equipables

Classes

[Misc](#)

Defines constants for miscellaneous equippable asset paths.

[Weapon](#)

Defines constants for equippable weapon asset paths.

Class Misc

Namespace: [S1API.Entities.Equippables](#)

Assembly: S1API.dll

Defines constants for miscellaneous equippable asset paths.

```
public class Misc
```

Inheritance

[object](#) ← Misc

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Misc()

```
public Misc()
```

Fields

Baton

The asset path for the Baton equippable item.

```
public const string Baton = "Avatar/Equippables/Baton"
```

Field Value

[string](#)

Beer

The asset path for the Beer equippable item.

```
public const string Beer = "Avatar/Equippables/Beer"
```

Field Value

[string](#)

Coffee

The asset path for the Coffee equippable item.

```
public const string Coffee = "Avatar/Equippables/Coffee"
```

Field Value

[string](#)

Cuke

The asset path for the Cuke equippable item.

```
public const string Cuke = "Avatar/Equippables/Cuke"
```

Field Value

[string](#)

Hammer

The asset path for the Hammer equippable item.

```
public const string Hammer = "Avatar/Equippables/Hammer"
```

Field Value

[string](#)

Joint

The asset path for the Joint equippable item.

```
public const string Joint = "Avatar/Equippables/Joint"
```

Field Value

[string](#)

Phone_Lowered

The asset path for the Phone_Lowered equippable item.

```
public const string Phone_Lowered = "Avatar/Equippables/Phone_Lowered"
```

Field Value

[string](#)

Phone_Raised

The asset path for the Phone_Raised equippable item.

```
public const string Phone_Raised = "Avatar/Equippables/Phone_Raised"
```

Field Value

[string](#)

Pipe

The asset path for the Pipe equippable item.

```
public const string Pipe = "Avatar/Equippables/Pipe"
```

Field Value

[string](#)

TrashBag

The asset path for the TrashBag equippable item.

```
public const string TrashBag = "Avatar/Equippables/TrashBag"
```

Field Value

[string](#)

Class Weapon

Namespace: [S1API.Entities.Equippables](#)

Assembly: S1API.dll

Defines constants for equippable weapon asset paths.

```
public class Weapon
```

Inheritance

[object](#) ← Weapon

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Weapon()

```
public Weapon()
```

Fields

BrokenBottle

The asset path for the Broken Bottle equippable melee weapon.

```
public const string BrokenBottle = "Avatar/Equippables/BrokenBottle"
```

Field Value

[string](#)

Knife

The asset path for the Crowbar equippable melee weapon.

```
public const string Knife = "Avatar/Equippables/Knife"
```

Field Value

[string](#)

M1911

The asset path for the M1911 equippable ranged weapon.

```
public const string M1911 = "Avatar/Equippables/M1911"
```

Field Value

[string](#)

PumpShotgun

The asset path for the Pump Shotgun equippable ranged weapon.

```
public const string PumpShotgun = "Avatar/Equippables/PumpShotgun"
```

Field Value

[string](#)

Revolver

The asset path for the Revolver equippable ranged weapon.

```
public const string Revolver = "Avatar/Equippables/Revolver"
```

Field Value

[string](#)

Taser

The asset path for the Taser equippable ranged weapon.

```
public const string Taser = "Avatar/Equippables/Taser"
```

Field Value

[string](#)

Namespace S1API.Entities.Interfaces

Interfaces

[IEntity](#)

Represents an entity within the game world.

[IHealth](#)

Represents an entity that has health associated.

Interface IEntity

Namespace: [S1API.Entities.Interfaces](#)

Assembly: S1API.dll

Represents an entity within the game world.

```
public interface IEntity
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Position

The world position of the entity.

```
Vector3 Position { get; set; }
```

Property Value

Vector3

Scale

The scale of the entity.

```
float Scale { get; set; }
```

Property Value

[float](#)

gameObject

INTERNAL: Tracking of the GameObject associated with this entity.

```
GameObject gameObject { get; }
```

Property Value

GameObject

Interface IHealth

Namespace: [S1API.Entities.Interfaces](#)

Assembly: S1API.dll

Represents an entity that has health associated.

```
public interface IHealth
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

CurrentHealth

The current health of the entity.

```
float CurrentHealth { get; }
```

Property Value

[float](#)

IsDead

Whether the entity is dead or not.

```
bool IsDead { get; }
```

Property Value

[bool](#)

IsInvincible

Whether the entity is invincible.

```
bool IsInvincible { get; set; }
```

Property Value

[bool](#)

MaxHealth

The max health of the entity.

```
float MaxHealth { get; set; }
```

Property Value

[float](#)

Methods

Damage(int)

Deals damage to the entity.

```
void Damage(int amount)
```

Parameters

[amount int](#)

Amount of damage to deal.

Heal(int)

Heals the entity.

```
void Heal(int amount)
```

Parameters

amount int ↗

Amount of health to heal.

Kill()

Kills the entity.

```
void Kill()
```

Revive()

Revives the entity.

```
void Revive()
```

Events

OnDeath

Called when entity's health is less than or equal to 0.

event Action OnDeath

Event Type

[Action](#) ↗

Namespace S1API.Entities.NPCs

Namespaces

[S1API.Entities.NPCs.Docks](#)

[S1API.Entities.NPCs.Downtown](#)

[S1API.Entities.NPCs.Northtown](#)

[S1API.Entities.NPCs.PoliceOfficers](#)

[S1API.Entities.NPCs.Suburbia](#)

[S1API.Entities.NPCs.Uptown](#)

[S1API.Entities.NPCs.Westville](#)

Classes

[DanSamwell](#)

UNCONFIRMED: Dan Samwell is a customer. He is the NPC that owns Dan's Hardware! If you confirm this, please let us know so we can update the documentation!

[IgorRomanovich](#)

Igor Romanovich is a npc. He is Manny's bodyguard. Igor can be found inside the Warehouse!

[MannyOakfield](#)

Manny is a NPC. He provides workers to the player. Manny can be found in the Warehouse!

[OscarHolland](#)

Oscar Holland is a NPC. He is a supplier located in the Warehouse!

[StanCarney](#)

Stan Carney is a NPC. He is the NPC that sells weapons. Stan can be found in the Warehouse!

[UncleNelson](#)

Uncle Nelson is a NPC. He is the uncle of the main character!

Namespace S1API.Entities.NPCs.Docks

Classes

[AnnaChesterfield](#)

Anna Chesterfield is a customer. She lives in the Docks region. Anna also works at the Barbershop.

[BillyKramer](#)

Billy Kramer is a customer. He lives in the Docks region.

[CrankyFrank](#)

Cranky Frank is a customer. He lives in the Docks region. Frank is the NPC with a pot on his head!

[GenghisBarn](#)

Genghis Barn is a customer. He lives in the Docks region. Genghis is the NPC with a mohawk!

[JaneLucero](#)

Jane Lucero is a dealer. She lives in the Docks region. Jane is the dealer with a tear tattoo!

[JavierPerez](#)

Javier Perez is a customer. He lives in the Docks region. Javier works night shift at the Gas-Mart!

[LisaGardener](#)

Lisa Gardener is a customer. She lives in the Docks region. Lisa is the NPC wearing blue scrubs!

[MacCooper](#)

Mac Cooper is a customer. He lives in the Docks region. Mac is the NPC with a blonde mohawk and gold shades!

[MarcoBaron](#)

Marco Baron is a customer. He lives in the Docks region. Marco is the NPC that runs the Auto Shop!

[MelissaWood](#)

Melissa Wood is a customer. She lives in the Docks region. Melissa is the Blackjack dealer at the casino!

[SalvadorMoreno](#)

Salvador Moreno is a supplier. He lives in the Docks region. Salvador is the NPC that supplies coca seeds to the player!

Class AnnaChesterfield

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Anna Chesterfield is a customer. She lives in the Docks region. Anna also works at the Barbershop.

```
public class AnnaChesterfield : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [AnnaChesterfield](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Anna Chesterfield. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class BillyKramer

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Billy Kramer is a customer. He lives in the Docks region.

```
public class BillyKramer : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [BillyKramer](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Billy Kramer. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class CrankyFrank

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Cranky Frank is a customer. He lives in the Docks region. Frank is the NPC with a pot on his head!

```
public class CrankyFrank : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ CrankyFrank

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Cranky Frank. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class GenghisBarn

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Genghis Barn is a customer. He lives in the Docks region. Genghis is the NPC with a mohawk!

```
public class GenghisBarn : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ GenghisBarn

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Genghis Barn. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JaneLucero

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Jane Lucero is a dealer. She lives in the Docks region. Jane is the dealer with a tear tattoo!

```
public class JaneLucero : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JaneLucero](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jane Lucero. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JavierPerez

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Javier Perez is a customer. He lives in the Docks region. Javier works night shift at the Gas-Mart!

```
public class JavierPerez : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← JavierPerez

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Javier Perez. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class LisaGardener

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Lisa Gardener is a customer. She lives in the Docks region. Lisa is the NPC wearing blue scrubs!

```
public class LisaGardener : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [LisaGardener](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Lisa Gardener. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MacCooper

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Mac Cooper is a customer. He lives in the Docks region. Mac is the NPC with a blonde mohawk and gold shades!

```
public class MacCooper : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← MacCooper

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Mac Cooper. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MarcoBaron

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Marco Baron is a customer. He lives in the Docks region. Marco is the NPC that runs the Auto Shop!

```
public class MarcoBaron : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [MarcoBaron](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Marco Baron. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MelissaWood

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Melissa Wood is a customer. She lives in the Docks region. Melissa is the Blackjack dealer at the casino!

```
public class MelissaWood : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [MelissaWood](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Melissa Wood. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class SalvadorMoreno

Namespace: [S1API.Entities.NPCs.Docks](#)

Assembly: S1API.dll

Salvador Moreno is a supplier. He lives in the Docks region. Salvador is the NPC that supplies coca seeds to the player!

```
public class SalvadorMoreno : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [SalvadorMoreno](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Salvador Moreno. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.NPCs.Downtown

Classes

[BradCrosby](#)

Brad Crosby is a dealer. He lives in the Downtown region. Brad lives in a tent at the parking garage next to the casino!

[ElizabethHomley](#)

Elizabeth Homley is a customer. She lives in the Downtown region. Elizabeth is the NPC with light blue hair!

[EugeneBuckley](#)

Eugene Buckley is a customer. He lives in the Downtown region. Eugene is the NPC with light brown hair, freckles, and black glasses!

[GregFliggle](#)

Greg Fliggle is a customer. He lives in the Downtown region. Greg is the NPC with a teardrop tattoo and wrinkles!

[JeffGilmore](#)

Jeff Gilmore is a customer. He lives in the Downtown region. Jeff is the NPC that runs the skateboard shop!

[JenniferRivera](#)

Jennifer Rivera is a customer. She lives in the Downtown region. Jennifer is the NPC with blonde haired buns!

[KevinOakley](#)

Kevin Oakley is a customer. He lives in the Downtown region. Kevin is the NPC wearing a green apron!

[LouisFourier](#)

Louis Fourier is a customer. He lives in the Downtown region. Louis is the NPC with a chef's hat!

[LucyPennington](#)

Lucy Pennington is a customer. She lives in the Downtown region. Lucy is the NPC with blonde haired buns up high!

[PhilipWentworth](#)

Philip Wentworth is a customer. He lives in the Downtown region. Philip is the bald NPC with a goatee!

[RandyCaulfield](#)

Randy Caulfield is a customer. He lives in the Downtown region. Randy is the NPC wearing a green hat!

Class BradCrosby

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Brad Crosby is a dealer. He lives in the Downtown region. Brad lives in a tent at the parking garage next to the casino!

```
public class BradCrosby : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← BradCrosby

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Brad Crosby. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class ElizabethHomley

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Elizabeth Homley is a customer. She lives in the Downtown region. Elizabeth is the NPC is lightning blue hair!

```
public class ElizabethHomley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← ElizabethHomley

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Elizabeth Homley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class EugeneBuckley

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Eugene Buckley is a customer. He lives in the Downtown region. Eugene is the NPC with light brown hair, freckles, and black glasses!

```
public class EugeneBuckley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [EugeneBuckley](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Eugene Buckley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class GregFliggle

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Greg Fliggle is a customer. He lives in the Downtown region. Greg is the NPC with a teardrop tattoo and wrinkles!

```
public class GregFliggle : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← GregFliggle

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Greg Fliggle. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JeffGilmore

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Jeff Gilmore is a customer. He lives in the Downtown region. Jeff is the NPC that runs the skateboard shop!

```
public class JeffGilmore : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JeffGilmore](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jeff Gilmore. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JenniferRivera

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Jennifer Rivera is a customer. She lives in the Downtown region. Jennifer is the NPC with blonde haired buns!

```
public class JenniferRivera : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JenniferRivera](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jennifer Rivera. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class KevinOakley

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Kevin Oakley is a customer. He lives in the Downtown region. Kevin is the NPC wearing a green apron!

```
public class KevinOakley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← KevinOakley

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Kevin Oakley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class LouisFourier

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Louis Fourier is a customer. He lives in the Downtown region. Louis is the NPC with a chef's hat!

```
public class LouisFourier : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← LouisFourier

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Louis Fourier. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class LucyPennington

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Lucy Pennington is a customer. She lives in the Downtown region. Lucy is the NPC with blonde haired buns up high!

```
public class LucyPennington : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← LucyPennington

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Lucy Pennington. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class PhilipWentworth

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Philip Wentworth is a customer. He lives in the Downtown region. Philip is the bald NPC with a goatee!

```
public class PhilipWentworth : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← PhilipWentworth

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Philip Wentworth. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class RandyCaulfield

Namespace: [S1API.Entities.NPCs.Downtown](#)

Assembly: S1API.dll

Randy Caulfield is a customer. He lives in the Downtown region. Randy is the NPC wearing a green hat!

```
public class RandyCaulfield : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [RandyCaulfield](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Randy Caulfield. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.NPCs.Northtown

Classes

[AlbertHoover](#)

Albert Hoover is a supplier. He lives in the Northtown region. Albert is the supplier for weed seeds!

[AustinSteiner](#)

Austin Steiner is a customer. He lives in the Northtown region. Austin is the NPC with a red/orange afro and black glasses!

[BenjiColeman](#)

Benji Coleman is a dealer. He lives in the Northtown region. Benji lives at the motel in room #2. He is the first dealer the player unlocks!

[BethPenn](#)

Beth Penn is a customer. She lives in the Northtown region. Beth is the NPC with a blonde bowl cut and wears green glasses!

[ChloeBowers](#)

Chloe Bowers is a customer. She lives in the Northtown region. Chloe is the NPC with long, straight, red hair!

[DonnaMartin](#)

Donna Martin is a customer. She lives in the Northtown region. Donna is the attendant of the Motel!

[GeraldinePoon](#)

Geraldine Poon is a customer. He lives in the Northtown region. Geraldine is the balding NPC with small gold glasses!

[JessiWaters](#)

Jessi Waters is a customer. She lives in the Northtown region. Jessi is the purple haired NPC with face tattoos!

[KathyHenderson](#)

Kathy Henderson is a customer. She lives in the Northtown region. Kathy is the NPC with long blonde hair with bangs!

[KyleCooley](#)

Kyle Cooley is a customer. He lives in the Northtown region. Kyle is the NPC that works at Taco Ticklers!

[LudwigMeyer](#)

Ludwig Meyer is a customer. He lives in the Northtown region. Ludwig is the NPC with spiky hair and gold glasses!

MickLubbin

Mick Lubbin is a customer. He lives in the Northtown region. Mick is the owner of the pawn shop!

Ming

Mrs. Ming is a customer. She lives in the Northtown region. Ming is the NPC that owns the chinese restaurant!

PeggyMyers

Peggy Myers is a customer. She lives in the Northtown region. Peggy is the NPC with freckles and brown hair pulled back!

PeterFile

Peter File is a customer. He lives in the Northtown region. Peter is the NPC with a black bowl cut and black glasses!

SamThompson

Sam Thompson is a customer. He lives in the Northtown region. Sam is the NPC with a green hair and wrinkles!

Class AlbertHoover

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Albert Hoover is a supplier. He lives in the Northtown region. Albert is the supplier for weed seeds!

```
public class AlbertHoover : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [AlbertHoover](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Albert Hoover. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class AustinSteiner

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Austin Steiner is a customer. He lives in the Northtown region. Austin is the NPC with a red/orange afro and black glasses!

```
public class AustinSteiner : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← AustinSteiner

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Austin Steiner. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class BenjiColeman

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Benji Coleman is a dealer. He lives in the Northtown region. Benji lives at the motel in room #2. He is the first dealer the player unlocks!

```
public class BenjiColeman : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← BenjiColeman

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Benji Coleman. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class BethPenn

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Beth Penn is a customer. She lives in the Northtown region. Beth is the NPC with a blonde bowl cut and wears green glasses!

```
public class BethPenn : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [BethPenn](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Beth Penn. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class ChloeBowers

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Chloe Bowers is a customer. She lives in the Northtown region. Chloe is the NPC with long, straight, red hair!

```
public class ChloeBowers : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [ChloeBowers](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Chloe Bowers. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class DonnaMartin

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Donna Martin is a customer. She lives in the Northtown region. Donna is the attendant of the Motel!

```
public class DonnaMartin : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [DonnaMartin](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Donna Martin. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class GeraldinePoon

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Geraldine Poon is a customer. He lives in the Northtown region. Geraldine is the balding NPC with small gold glasses!

```
public class GeraldinePoon : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [GeraldinePoon](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Geraldine Poon. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JessiWaters

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Jessi Waters is a customer. She lives in the Northtown region. Jessi is the purple haired NPC with face tattoos!

```
public class JessiWaters : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JessiWaters](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jessi Waters. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class KathyHenderson

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Kathy Henderson is a customer. She lives in the Northtown region. Kathy is the NPC with long blonde hair with bangs!

```
public class KathyHenderson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [KathyHenderson](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Kathy Henderson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class KyleCooley

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Kyle Cooley is a customer. He lives in the Northtown region. Kyle is the NPC that works at Taco Ticklers!

```
public class KyleCooley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [KyleCooley](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Kyle Cooley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class LudwigMeyer

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Ludwig Meyer is a customer. He lives in the Northtown region. Ludwig is the NPC with spiky hair and gold glasses!

```
public class LudwigMeyer : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← LudwigMeyer

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Ludwig Meyer. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MickLubbin

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Mick Lubbin is a customer. He lives in the Northtown region. Mick is the owner of the pawn shop!

```
public class MickLubbin : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [MickLubbin](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Mick Lubbin. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class Ming

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Mrs. Ming is a customer. She lives in the Northtown region. Ming is the NPC that owns the chinese restaurant!

```
public class Ming : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← Ming

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Ming. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class PeggyMyers

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Peggy Myers is a customer. She lives in the Northtown region. Peggy is the NPC with freckles and brown hair pulled back!

```
public class PeggyMyers : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [PeggyMyers](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Peggy Myers. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class PeterFile

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Peter File is a customer. He lives in the Northtown region. Peter is the NPC with a black bowl cut and black glasses!

```
public class PeterFile : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← PeterFile

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Peter File. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class SamThompson

Namespace: [S1API.Entities.NPCs.Northtown](#)

Assembly: S1API.dll

Sam Thompson is a customer. He lives in the Northtown region. Sam is the NPC with a green hair and wrinkles!

```
public class SamThompson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← SamThompson

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Sam Thompson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.NPCs.PoliceOfficers

Classes

[OfficerBailey](#)

Officer Bailey is a police officer. He is the bald officer with a swirling mustache!

[OfficerCooper](#)

Officer Cooper is a police officer. She is the officer with two high black buns and black glasses!

[OfficerGreen](#)

Officer Green is a police officer. She is the officer with light brown hair in a bun!

[OfficerHoward](#)

Officer Howard is a police officer. He is the officer with a light brown afro and goatee!

[OfficerJackson](#)

Officer Jackson is a police officer. He is the officer with a light brown goatee and police hat!

[OfficerLee](#)

Officer Lee is a police officer. He is the officer with a button-up shirt and black hair!

[OfficerLopez](#)

Officer Lopez is a police officer. She is the officer with a blue button-up and long black hair!

[OfficerMurphy](#)

Officer Murphy is a police officer. He is the balding officer with grey hair and wrinkles!

[OfficerOakley](#)

Officer Oakley is a police officer. He is the officer with light brown spiky hair and a goatee!

Class OfficerBailey

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Bailey is a police officer. He is the bald officer with a swirling mustache!

```
public class OfficerBailey : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ OfficerBailey

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Bailey. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerCooper

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Cooper is a police officer. She is the officer with two high black buns and black glasses!

```
public class OfficerCooper : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [OfficerCooper](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Cooper. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerGreen

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Green is a police officer. She is the officer with light brown hair in a bun!

```
public class OfficerGreen : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [OfficerGreen](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Green. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerHoward

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Howard is a police officer. He is the officer with a light brown afro and goatee!

```
public class OfficerHoward : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [OfficerHoward](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Howard. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerJackson

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Jackson is a police officer. He is the officer with a light brown goatee and police hat!

```
public class OfficerJackson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [OfficerJackson](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Jackson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerLee

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Lee is a police officer. He is the officer with a button-up shirt and black hair!

```
public class OfficerLee : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [OfficerLee](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Lee. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerLopez

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Lopez is a police officer. She is the officer with a blue button-up and long black hair!

```
public class OfficerLopez : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ OfficerLopez

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Lopez. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerMurphy

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Murphy is a police officer. He is the balding officer with grey hair and wrinkles!

```
public class OfficerMurphy : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [OfficerMurphy](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Murphy. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OfficerOakley

Namespace: [S1API.Entities.NPCs.PoliceOfficers](#)

Assembly: S1API.dll

Officer Oakley is a police officer. He is the officer with light brown spiky hair and a goatee!

```
public class OfficerOakley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ OfficerOakley

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Officer Oakley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.NPCs.Suburbia

Classes

[AlisonKnight](#)

Alison Knight is a customer. She lives in the Suburbia region. Alison is the NPC with long light brown hair!

[CarlBundy](#)

Carl Bundy is a customer. He lives in the Suburbia region. Carl is the NPC with a brown apron!

[ChrisSullivan](#)

Chris Sullivan is a customer. He lives in the Suburbia region. Chris is the NPC with black spiky hair and black glasses!

[DennisKennedy](#)

Dennis Kennedy is a customer. He lives in the Suburbia region. Dennis is the NPC with light blonde spiky hair and a thick mustache!

[HankStevenson](#)

Hank Stevenson is a customer. He lives in the Suburbia region. Hank is the balding NPC with greying brown hair and a goatee!

[HaroldColt](#)

Harold Colt is a customer. He lives in the Suburbia region. Harold is the NPC with grey spiky hair and wrinkles!

[JackKnight](#)

Jack Knight is a customer. He lives in the Suburbia region. Jack is the balding NPC with small gold glasses!

[JackieStevenson](#)

Jackie Stevenson is a customer. He lives in the Suburbia region. Jackie is the NPC with short brown hair and light freckles!

[JeremyWilkinson](#)

Jeremy Wilkinson is a customer. He lives in the Suburbia region. Jeremy is the NPC that works at Hyland Auto!

[KarenKennedy](#)

Karen Kennedy is a customer. She lives in the Suburbia region. Karen is the NPC with wavy blonde hair and purple eyelids! She can be found at the casino upstairs when it's open.

WeiLong

Wei Long is a dealer. He lives in the Suburbia region. Wei is the dealer with a black bowl cut and gold glasses!

Class AlisonKnight

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Alison Knight is a customer. She lives in the Suburbia region. Alison is the NPC with long light brown hair!

```
public class AlisonKnight : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← AlisonKnight

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Alison Knight. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class CarlBundy

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Carl Bundy is a customer. He lives in the Suburbia region. Carl is the NPC with a brown apron!

```
public class CarlBundy : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [CarlBundy](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Carl Bundy. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class ChrisSullivan

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Chris Sullivan is a customer. He lives in the Suburbia region. Chris is the NPC with black spiky hair and black glasses!

```
public class ChrisSullivan : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [ChrisSullivan](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Chris Sullivan. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class DennisKennedy

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Dennis Kennedy is a customer. He lives in the Suburbia region. Dennis is the NPC with light blonde spiky hair and a thick mustache!

```
public class DennisKennedy : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← DennisKennedy

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Dennis Kennedy. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class HankStevenson

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Hank Stevenson is a customer. He lives in the Suburbia region. Hank is the balding NPC with greying brown hair and a goatee!

```
public class HankStevenson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [HankStevenson](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Hank Stevenson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class HaroldColt

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Harold Colt is a customer. He lives in the Suburbia region. Harold is the NPC with grey spiky hair and wrinkles!

```
public class HaroldColt : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [HaroldColt](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Harold Colt. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JackKnight

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Jack Knight is a customer. He lives in the Suburbia region. Jack is the balding NPC with small gold glasses!

```
public class JackKnight : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JackKnight](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jack Knight. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JackieStevenson

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Jackie Stevenson is a customer. He lives in the Suburbia region. Jackie is the NPC with short brown hair and light freckles!

```
public class JackieStevenson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← JackieStevenson

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jackie Stevenson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JeremyWilkinson

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Jeremy Wilkinson is a customer. He lives in the Suburbia region. Jeremy is the NPC that works at Hyland Auto!

```
public class JeremyWilkinson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JeremyWilkinson](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jeremy Wilkinson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class KarenKennedy

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Karen Kennedy is a customer. She lives in the Suburbia region. Karen is the NPC with wavy blonde hair and purple eyelids! She can be found at the casino upstairs when it's open.

```
public class KarenKennedy : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [KarenKennedy](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Karen Kennedy. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class WeiLong

Namespace: [S1API.Entities.NPCs.Suburbia](#)

Assembly: S1API.dll

Wei Long is a dealer. He lives in the Suburbia region. Wei is the dealer with a black bowl cut and gold glasses!

```
public class WeiLong : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← WeiLong

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Wei Long. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.NPCs.Uptown

Classes

[FionaHancock](#)

Fiona Hancock is a customer. She lives in the Uptown region. Fiona is the NPC with light brown buns and green glasses!

[HerbertBleuball](#)

Herbert Bleuball is a customer. He lives in the Uptown region. Herbert is the NPC that owns Bleuball's Boutique!

[JenHeard](#)

Jen Heard is a customer. She lives in the Uptown region. Jen is the NPC with low orange buns!

[LeoRivers](#)

Leo Rivers is a dealer. He lives in the Uptown region. Leo is the dealer wearing a black hat and gold shades!

[LilyTurner](#)

Lily Turner is a customer. She lives in the Uptown region. Lily is the NPC with long brown hair with bangs!

[MichaelBoog](#)

Michael Boog is a customer. He lives in the Uptown region. Michael is the NPC with a bright blue flat cap and black glasses!

[PearlMoore](#)

Pearl Moore is a customer. She lives in the Uptown region. Pearl is the NPC with long white hair with bangs!

[RayHoffman](#)

Ray Hoffman is a customer. He lives in the Uptown region. Ray is the NPC that owns Ray's Realty!

[TobiasWentworth](#)

Tobias Wentworth is a customer. He lives in the Uptown region. Tobias is the balding NPC with extremely light brown hair and small black glasses!

[WalterCussler](#)

Walter Cussler is a customer. He lives in the Uptown region. Walter is the NPC with white hair and dressed as a priest!

Class FionaHancock

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Fiona Hancock is a customer. She lives in the Uptown region. Fiona is the NPC with light brown buns and green glasses!

```
public class FionaHancock : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [FionaHancock](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Fiona Hancock. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class HerbertBleuball

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Herbert Bleuball is a customer. He lives in the Uptown region. Herbert is the NPC that owns Bleuball's Boutique!

```
public class HerbertBleuball : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [HerbertBleuball](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Herbert Bleuball. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JenHeard

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Jen Heard is a customer. She lives in the Uptown region. Jen is the NPC with low orange buns!

```
public class JenHeard : NPC, IEntity, IHealth
```

Inheritance

[object](#) ↳ [Registerable](#) ↳ [Saveable](#) ↳ [NPC](#) ↳ [JenHeard](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#) ↳, [object.MemberwiseClone\(\)](#) ↳, [object.ToString\(\)](#) ↳,
[object.Equals\(object\)](#) ↳, [object.Equals\(object, object\)](#) ↳, [object.ReferenceEquals\(object, object\)](#) ↳,
[object.GetHashCode\(\)](#) ↳

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jen Heard. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class LeoRivers

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Leo Rivers is a dealer. He lives in the Uptown region. Leo is the dealer wearing a black hat and gold shades!

```
public class LeoRivers : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [LeoRivers](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Leo Rivers. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class LilyTurner

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Lily Turner is a customer. She lives in the Uptown region. Lily is the NPC with long brown hair with bangs!

```
public class LilyTurner : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [LilyTurner](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Lily Turner. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MichaelBoog

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Michael Boog is a customer. He lives in the Uptown region. Michael is the NPC with a bright blue flat cap and black glasses!

```
public class MichaelBoog : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← MichaelBoog

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Michael Boog. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class PearlMoore

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Pearl Moore is a customer. She lives in the Uptown region. Pearl is the NPC with long white hair with bangs!

```
public class PearlMoore : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [PearlMoore](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Pearl Moore. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class RayHoffman

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Ray Hoffman is a customer. He lives in the Uptown region. Ray is the NPC that owns Ray's Realty!

```
public class RayHoffman : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← RayHoffman

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Ray Hoffman. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class TobiasWentworth

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Tobias Wentworth is a customer. He lives in the Uptown region. Tobias is the balding NPC with extremely light brown hair and small black glasses!

```
public class TobiasWentworth : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [TobiasWentworth](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Tobias Wentworth. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class WalterCussler

Namespace: [S1API.Entities.NPCs.Uptown](#)

Assembly: S1API.dll

Walter Cussler is a customer. He lives in the Uptown region. Walter is the NPC with white hair and dressed as a priest!

```
public class WalterCussler : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [WalterCussler](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Walter Cussler. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.NPCs.Westville

Classes

[CharlesRowland](#)

Charles Rowland is a customer. He lives in the Westville region. Charles is the bald NPC with black glasses!

[DeanWebster](#)

Dean Webster is a customer. He lives in the Westville region. Dean is the NPC that owns Top Tattoo!

[DorisLubbin](#)

Doris Lubbin is a customer. She lives in the Westville region. Doris is the NPC with light brown, wavy hair and black glasses!

[GeorgeGreene](#)

George Greene is a customer. He lives in the Westville region. George is the NPC with light brown, spiky hair and gold glasses!

[JerryMontero](#)

Jerry Montero is a customer. He lives in the Westville region. Jerry is the NPC with a green hat and black glasses!

[JoyceBall](#)

Joyce Ball is a customer. She lives in the Westville region. Joyce is the NPC with light brown hair and wrinkles!

[KeithWagner](#)

Keith Wagner is a customer. He lives in the Westville region. Keith is the NPC with blonde spiky hair and always angry!

[KimDelaney](#)

Kim Delaney is a customer. She lives in the Westville region. Kim is the NPC with long, black hair with bangs!

[MegCooley](#)

Meg Cooley is a customer. She lives in the Westville region. Meg is the npc with a mustard yellow bowl cut hairstyle!

[MollyPresley](#)

Molly Presley is a dealer. She lives in the Westville region. Molly is the dealer with gold shades and a red backward cap!

ShirleyWatts

Shirley Watts is a supplier. She lives in the Westville region. Shirley is the supplier for pseudo!

TrentSherman

Trent Sherman is a customer. He lives in the Westville region. Trent is the NPC with short black hair and dark-colored skin!

Class CharlesRowland

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Charles Rowland is a customer. He lives in the Westville region. Charles is the bald NPC with black glasses!

```
public class CharlesRowland : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← CharlesRowland

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Charles Rowland. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class DeanWebster

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Dean Webster is a customer. He lives in the Westville region. Dean is the NPC that owns Top Tattoo!

```
public class DeanWebster : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [DeanWebster](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Dean Webster. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class DorisLubbin

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Doris Lubbin is a customer. She lives in the Westville region. Doris is the NPC with light brown, wavy hair and black glasses!

```
public class DorisLubbin : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← DorisLubbin

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Doris Lubbin. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class GeorgeGreene

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

George Greene is a customer. He lives in the Westville region. George is the NPC with light brown, spiky hair and gold glasses!

```
public class GeorgeGreene : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [GeorgeGreene](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for George Greene. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JerryMontero

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Jerry Montero is a customer. He lives in the Westville region. Jerry is the NPC with a green hat and black glasses!

```
public class JerryMontero : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JerryMontero](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Jerry Montero. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class JoyceBall

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Joyce Ball is a customer. She lives in the Westville region. Joyce is the NPC with light brown hair and wrinkles!

```
public class JoyceBall : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [JoyceBall](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Joyce Ball. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class KeithWagner

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Keith Wagner is a customer. He lives in the Westville region. Keith is the NPC with blonde spiky hair and always angry!

```
public class KeithWagner : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [KeithWagner](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Keith Wagner. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class KimDelaney

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Kim Delaney is a customer. She lives in the Westville region. Kim is the NPC with long, black hair with bangs!

```
public class KimDelaney : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← KimDelaney

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Kim Delaney. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MegCooley

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Meg Cooley is a customer. She lives in the Westville region. Meg is the npc with a mustard yellow bowl cut hairstyle!

```
public class MegCooley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← MegCooley

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Meg Cooley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MollyPresley

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Molly Presley is a dealer. She lives in the Westville region. Molly is the dealer with gold shades and a red backward cap!

```
public class MollyPresley : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [MollyPresley](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Molly Presley. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class ShirleyWatts

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Shirley Watts is a supplier. She lives in the Westville region. Shirley is the supplier for pseudo!

```
public class ShirleyWatts : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [ShirleyWatts](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Shirley Watts. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class TrentSherman

Namespace: [S1API.Entities.NPCs.Westville](#)

Assembly: S1API.dll

Trent Sherman is a customer. He lives in the Westville region. Trent is the NPC with short black hair and dark-colored skin!

```
public class TrentSherman : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [TrentSherman](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Trent Sherman. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class DanSamwell

Namespace: [S1API.Entities.NPCs](#)

Assembly: S1API.dll

UNCONFIRMED: Dan Samwell is a customer. He is the NPC that owns Dan's Hardware! If you confirm this, please let us know so we can update the documentation!

```
public class DanSamwell : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← DanSamwell

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Dan Samwell. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class IgorRomanovich

Namespace: [S1API.Entities.NPCs](#)

Assembly: S1API.dll

Igor Romanovich is a npc. He is Manny's bodyguard. Igor can be found inside the Warehouse!

```
public class IgorRomanovich : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← IgorRomanovich

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Igor Romanovich. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class MannyOakfield

Namespace: [S1API.Entities.NPCs](#)

Assembly: S1API.dll

Manny is a NPC. He provides workers to the player. Manny can be found in the Warehouse!

```
public class MannyOakfield : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [MannyOakfield](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Manny Oakfield. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class OscarHolland

Namespace: [S1API.Entities.NPCs](#)

Assembly: S1API.dll

Oscar Holland is a NPC. He is a supplier located in the Warehouse!

```
public class OscarHolland : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [OscarHolland](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Oscar Holland. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class StanCarney

Namespace: [S1API.Entities.NPCs](#)

Assembly: S1API.dll

Stan Carney is a NPC. He is the NPC that sells weapons. Stan can be found in the Warehouse!

```
public class StanCarney : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← StanCarney

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Stan Carney. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Class UncleNelson

Namespace: [S1API.Entities.NPCs](#)

Assembly: S1API.dll

Uncle Nelson is a NPC. He is the uncle of the main character!

```
public class UncleNelson : NPC, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← [NPC](#) ← [UncleNelson](#)

Implements

[IEntity](#), [IHealth](#)

Inherited Members

[NPC.Responses](#), [NPC.All](#), [NPC.RefreshMessagingIcons\(\)](#), [NPC.PreRegisterPrefabForType\(Type\)](#),
[NPC.PreRegisterAllNpcPrefabs\(\)](#), [NPC.ConfigurePrefab\(NPCPrefabBuilder\)](#),
[NPC.OnResponseLoaded\(Response\)](#), [NPC.OnCreated\(\)](#), [NPC.Revive\(\)](#), [NPC.Damage\(int\)](#), [NPC.Heal\(int\)](#),
[NPC.Kill\(\)](#), [NPC.Unsettle\(float\)](#), [NPC.LerpScale\(float, float\)](#), [NPC.Panic\(\)](#), [NPC.StopPanicking\(\)](#),
[NPC.KnockOut\(\)](#), [NPC.Goto\(Vector3\)](#), [NPC.ClearConversationCategories\(\)](#),
[NPC.SendTextMessage\(string, Response\[\], float, bool\)](#), [NPC.SetEquippable\(string\)](#), [NPC.Get<T>\(\)](#),
[NPC.gameObject](#), [NPC.Position](#), [NPC.Transform](#), [NPC.CustomNpcsReady](#), [NPC.FirstName](#),
[NPC.LastName](#), [NPC.FullName](#), [NPC.ID](#), [NPC.Icon](#), [NPC.IsConscious](#), [NPC.IsInBuilding](#),
[NPC.IsInVehicle](#), [NPC.IsPanicking](#), [NPC.IsUnsettled](#), [NPC.IsVisible](#), [NPC.IsPhysical](#), [NPC.IsDealer](#),
[NPC.Aggressive](#), [NPC.Region](#), [NPC.PanicDuration](#), [NPC.Scale](#), [NPC.IsKnockedOut](#),
[NPC.RequiresRegionUnlocked](#), [NPC.CurrentVehicle](#), [NPC.CurrentHealth](#), [NPC.MaxHealth](#), [NPC.IsDead](#),
[NPC.IsInvincible](#), [NPC.Appearance](#), [NPC.Movement](#), [NPC.CombatBehaviour](#), [NPC.Dialogue](#),
[NPC.Schedule](#), [NPC.Inventory](#), [NPC.Customer](#), [NPC.Dealer](#), [NPC.Relationship](#),
[NPC.ConversationCanBeHidden](#), [NPC.OnDeath](#), [NPC.OnInventoryChanged](#),
[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

NPCId

Static NPC ID for Uncle Nelson. Used to resolve connections during prefab configuration.

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

Property Value

[string ↗](#)

Namespace S1API.Entities.Relation

Classes

[NPCRelationshipDataBuilder](#)

Builder for configuring an NPC's relationship data from code. Supports setting relationship delta, unlock state/type, and connections by ID or wrapper.

Class NPCRelationshipDataBuilder

Namespace: [S1API.Entities.Relation](#)

Assembly: S1API.dll

Builder for configuring an NPC's relationship data from code. Supports setting relationship delta, unlock state/type, and connections by ID or wrapper.

```
public sealed class NPCRelationshipDataBuilder
```

Inheritance

[object](#) ← NPCRelationshipDataBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NPCRelationshipDataBuilder()

```
public NPCRelationshipDataBuilder()
```

Methods

ApplyTo(NPCRelationData, NPC, bool)

INTERNAL: Applies the configured values to a relation data instance.

```
public void ApplyTo(NPCRelationData relationData, NPC owner, bool
```

```
preserveUnlockState = false)
```

Parameters

relationData NPCRelationData

owner NPC

preserveUnlockState [bool](#)

If true, will not modify unlock state if the NPC is already unlocked (preserves save data).

SetUnlockType(UnlockType)

Sets unlock type by enum.

```
public NPCRelationshipDataBuilder SetUnlockType(NPCRelationship.UnlockType type)
```

Parameters

type [NPCRelationship.UnlockType](#)

Returns

[NPCRelationshipDataBuilder](#)

SetUnlockType(string)

Sets unlock type by name ("Recommendation", "DirectApproach").

```
public NPCRelationshipDataBuilder SetUnlockType(string typeName)
```

Parameters

typeName [string](#)

Returns

[NPCRelationshipDataBuilder](#)

SetUnlocked(bool)

Sets whether the NPC is unlocked.

```
public NPCRelationshipDataBuilder SetUnlocked(bool unlocked)
```

Parameters

unlocked [bool](#)

Returns

[NPCRelationshipDataBuilder](#)

WithConnections(params NPC[])

Replaces the connections list using API NPC wrappers. Nulls are ignored.

```
[Obsolete("Use WithConnections<T1, T2, ...>() or WithConnectionsById instead. NPC instances are not available during prefab configuration.")]
public NPCRelationshipDataBuilder WithConnections(params NPC[] npcs)
```

Parameters

npcs [NPC\[\]](#)

Returns

[NPCRelationshipDataBuilder](#)

WithConnections(IEnumerable<NPC>)

Replaces the connections list using API NPC wrappers. Nulls are ignored. For NPCs that aren't spawned yet (during prefab configuration), this will attempt to extract IDs from the NPC type's static NPCId property or by looking them up in the base game's NPC registry. Prefer [WithConnections\(params Type\[\]\)](#) or the generic overloads when you only have types available.

```
[Obsolete("Use WithConnections<T1, T2, ...>() or WithConnectionsById instead. NPC instances are not available during prefab configuration.")]
public NPCRelationshipDataBuilder WithConnections(IEnumerable<NPC> npcs)
```

Parameters

npcs [IEnumerable<NPC>](#)

Returns

[NPCRelationshipDataBuilder](#)

WithConnections(params Type[])

Replaces the connections list using NPC types. This overload works during prefab configuration when NPC instances are not yet available. IDs are resolved from the static NPCId property.

```
public NPCRelationshipDataBuilder WithConnections(params Type[] npcTypes)
```

Parameters

npcTypes [Type\[\]](#)

Returns

[NPCRelationshipDataBuilder](#)

WithConnectionsById(IEnumerable<string>)

Replaces the connections list with the given NPC IDs (case-insensitive).

```
public NPCRelationshipDataBuilder WithConnectionsById(IEnumerable<string> ids)
```

Parameters

ids [IEnumerable<string>](#)

Returns

[NPCRelationshipDataBuilder](#)

WithConnectionsById(params string[])

Replaces the connections list with the given NPC IDs (case-insensitive).

```
public NPCRelationshipDataBuilder WithConnectionsById(params string[] ids)
```

Parameters

ids [string\[\]](#)

Returns

[NPCRelationshipDataBuilder](#)

WithConnections<T1, T2>()

Replaces the connections list using NPC types. This overload works during prefab configuration when NPC instances are not yet available. IDs are resolved from the static NPCId property.

```
public NPCRelationshipDataBuilder WithConnections<T1, T2>() where T1 : NPC where T2 : NPC
```

Returns

[NPCRelationshipDataBuilder](#)

Type Parameters

T1

First NPC type

T2

Second NPC type

WithConnections<T1, T2, T3>()

Replaces the connections list using NPC types. This overload works during prefab configuration when NPC instances are not yet available. IDs are resolved from the static NPCId property.

```
public NPCRelationshipDataBuilder WithConnections<T1, T2, T3>() where T1 : NPC where T2 : NPC where T3 : NPC
```

Returns

[NPCRelationshipDataBuilder](#)

Type Parameters

T1

First NPC type

T2

Second NPC type

Third NPC type

WithDelta(float)

Sets the relationship delta in [0, 5].

```
public NPCRelationshipDataBuilder WithDelta(float delta)
```

Parameters

delta [float](#)

Returns

[NPCRelationshipDataBuilder](#)

WithNormalized(float)

Sets the relationship delta using a normalized [0..1] value.

```
public NPCRelationshipDataBuilder WithNormalized(float normalized)
```

Parameters

normalized [float](#)

Returns

[NPCRelationshipDataBuilder](#)

Namespace S1API.Entities.Schedule

Classes

[DriveToCarParkSpec](#)

Specifies an action that makes an NPC drive to a car park and park a specified vehicle.

[EnsureDealSignalSpec](#)

Specifies an action that ensures a customer deal signal exists under the NPC's schedule.

[HandleDealSpec](#)

Specifies an action that handles a customer deal handover at the active contract location.

[LocationDialogueSpec](#)

Specifies an action that makes an NPC walk to a destination and start a location-based dialogue or interaction.

[PrefabScheduleBuilder](#)

Plan-time schedule builder used during prefab composition. Collects [IScheduleActionSpec](#) entries without requiring a live NPC instance.

[SitSpec](#)

Specifies an action that seats an NPC at an ScheduleOne.AvatarFramework.Animation.AvatarSeatSet.

[StayInBuildingSpec](#)

Specifies an action that makes an NPC remain inside a building for a specified duration.

[UseATMSpec](#)

Specifies an action that makes an NPC use an ATM at a scheduled time.

[UseSlotMachineSpec](#)

Specifies an action that makes an NPC use a slot machine at a scheduled time. Supports single spins, multiple spins, time-based sessions, or gambling until broke.

[UseVendingMachineSpec](#)

Specifies an action that makes an NPC use a vending machine at a scheduled time.

[WalkToSpec](#)

Specifies a walk-to action that moves an NPC to a specific world position at a scheduled time.

Interfaces

[IScheduleActionSpec](#)

Defines a specification for a schedule action that can be applied to an NPC's schedule. This interface provides a modder-facing, game-type-free way to define schedule actions that can be used during prefab configuration.

Class DriveToCarParkSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that makes an NPC drive to a car park and park a specified vehicle.

```
public sealed class DriveToCarParkSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← DriveToCarParkSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This action creates a ScheduleOne.NPCs.Schedules.NPCTSignal_DriveToCarPark that will make the NPC drive the specified vehicle to the designated parking lot and park it. The parking lot and vehicle can be specified using wrapper objects.

Constructors

DriveToCarParkSpec()

```
public DriveToCarParkSpec()
```

Properties

Alignment

Gets or sets the S1API-facing parking alignment to use when parking the vehicle.

```
public ParkingAlignment? Alignment { get; set; }
```

Property Value

[ParkingAlignment?](#)

The parking alignment, or `null` to use default or type-based alignment.

Remarks

This property takes precedence over [ParkingType](#) when both are specified. It provides a more type-safe way to specify parking behavior using the S1API [ParkingAlignment](#) enum.

Name

Gets or sets the optional name for this action.

```
public string Name { get; set; }
```

Property Value

[string](#) ↗

The action name, or `null` to use the default name "DriveToCarPark".

OverrideParkingType

Gets or sets whether to override the default parking type behavior.

```
public bool? OverrideParkingType { get; set; }
```

Property Value

[bool](#)?

`true` to override parking type; otherwise, `null` for default behavior.

Remarks

When set to `true`, the action will use the parking type specified in [ParkingType](#) or [Alignment](#) instead of the default behavior.

ParkingLot

Gets or sets the parking lot wrapper object to avoid GUID lookups.

```
public ParkingLotWrapper ParkingLot { get; set; }
```

Property Value

[ParkingLotWrapper](#)

The parking lot wrapper, or `null` to use GUID-based lookup.

Remarks

This property takes precedence over [ParkingLotGUID](#).

ParkingLotGUID

Gets or sets the GUID of the parking lot where the vehicle should be parked.

```
public string ParkingLotGUID { get; set; }
```

Property Value

[string](#)

The parking lot GUID, or `null` if using wrapper object.

Remarks

This property is used as a fallback if [ParkingLot](#) is not specified. The GUID should match the parking lot's unique identifier in the game.

ParkingLotName

Gets or sets the name of the parking lot GameObject for runtime resolution.

```
public string ParkingLotName { get; set; }
```

Property Value

[string](#)

The GameObject name, or `null` if not using name-based resolution.

Remarks

This property is used at runtime to find the parking lot by GameObject name. Takes precedence over GUID lookup but is overridden by wrapper object.

ParkingType

Gets or sets the parking type to use when parking the vehicle.

```
public int? ParkingType { get; set; }
```

Property Value

[int](#)?

The parking type as an integer, or `null` to use default or alignment-based type.

Remarks

This property is used when [OverrideParkingType](#) is `true` and [Alignment](#) is not specified. The value should correspond to the ScheduleOne.Vehicles.EParkingAlignment enum values.

StartTime

Gets or sets the time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

The start time in minutes (0-1439 for a 24-hour day).

Vehicle

Gets or sets the vehicle wrapper object to avoid GUID lookups.

```
public LandVehicle Vehicle { get; set; }
```

Property Value

[LandVehicle](#)

The vehicle wrapper, or `null` to use GUID-based lookup.

Remarks

This property takes precedence over [VehicleGUID](#) and [VehicleName](#).

VehicleCode

Gets or sets a vehicle code for runtime vehicle creation.

```
public string VehicleCode { get; set; }
```

Property Value

[string](#)

The vehicle code to create (e.g., "Sedan", "SUV"), or `null` if not creating.

Remarks

This property is used at runtime to create a new vehicle if one doesn't exist. Only used if Vehicle, VehicleGUID, and VehicleName are all null or failed resolution.

VehicleGUID

Gets or sets the GUID of the vehicle that should be driven to the parking lot.

```
public string VehicleGUID { get; set; }
```

Property Value

[string](#)

The vehicle GUID, or `null` if using wrapper object.

Remarks

This property is used as a fallback if [Vehicle](#) is not specified. The GUID should match the vehicle's unique identifier in the game.

VehicleName

Gets or sets the name of the vehicle GameObject for runtime resolution.

```
public string VehicleName { get; set; }
```

Property Value

[string](#) ↗

The GameObject name, or `null` if not using name-based resolution.

Remarks

This property is used at runtime to find the vehicle by GameObject name. Takes precedence over GUID lookup but is overridden by wrapper object.

VehicleSpawnPosition

Gets or sets the spawn position for a created vehicle.

```
public Vector3 VehicleSpawnPosition { get; set; }
```

Property Value

[Vector3](#)

The world position where the vehicle should spawn.

Remarks

This property is used when creating a vehicle via [VehicleCode](#). The vehicle will be spawned at this exact position when the schedule action executes.

VehicleSpawnRotation

Gets or sets the spawn rotation for a created vehicle.

```
public Quaternion? VehicleSpawnRotation { get; set; }
```

Property Value

Quaternion?

The world rotation for the vehicle spawn, or `null` to use identity rotation.

Remarks

This property is used when creating a vehicle via [VehicleCode](#). If not specified, the vehicle will spawn with identity rotation (no rotation).

Class EnsureDealSignalSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that ensures a customer deal signal exists under the NPC's schedule.

```
public sealed class EnsureDealSignalSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← EnsureDealSignalSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This action ensures that a ScheduleOne.NPCs.Schedules.NPCSignal_WaitForDelivery component exists on the NPC's schedule manager. This signal is required for customer NPCs to properly handle deal interactions and handovers with the player.

The deal signal allows the NPC to wait for deliveries and toggle customer handover states. If the signal doesn't exist, it will be created and properly initialized with network components and wired to the customer component.

Constructors

EnsureDealSignalSpec()

```
public EnsureDealSignalSpec()
```


Class HandleDealSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that handles a customer deal handover at the active contract location.

```
[Obsolete("HandleDealSpec is no longer needed as of game version 0.4.2f4. Deal handling is now automatic through DealerAttendDealBehaviour.")]
public sealed class HandleDealSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← HandleDealSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

As of v0.4.2f4, deal handling is now automatic through the DealerAttendDealBehaviour system. This spec is kept for backwards compatibility but is a no-op. Dealer NPCs set up with EnsureDealer() will automatically handle deals when contracts are assigned.

Constructors

HandleDealSpec()

```
public HandleDealSpec()
```

Properties

Name

Optional custom name for the action.

```
public string Name { get; set; }
```

Property Value

[string](#) ↗

StartTime

The time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#) ↗

Interface IScheduleActionSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Defines a specification for a schedule action that can be applied to an NPC's schedule. This interface provides a modder-facing, game-type-free way to define schedule actions that can be used during prefab configuration.

```
public interface IScheduleActionSpec
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

ApplyTo(NPCSchedule)

Applies this action specification to the given NPC schedule.

```
void ApplyTo(NPCSchedule schedule)
```

Parameters

schedule [NPCSchedule](#)

The NPC schedule to apply this action to.

Remarks

This method should create and configure the appropriate schedule action on the provided schedule instance. The implementation should handle any necessary object resolution, validation, and error handling.

Class LocationDialogueSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that makes an NPC walk to a destination and start a location-based dialogue or interaction.

```
public sealed class LocationDialogueSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← LocationDialogueSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This action creates a ScheduleOne.NPCs.Schedules.NPCEvent_LocationDialogue that will make the NPC walk to the specified destination and then trigger a dialogue or interaction at that location. This is useful for creating NPCs that wait at specific locations for player interaction.

Constructors

LocationDialogueSpec()

```
public LocationDialogueSpec()
```

Properties

ChoiceToEnable

Gets or sets the choice index to enable when the NPC reaches the destination.

```
public int ChoiceToEnable { get; set; }
```

Property Value

[int ↗](#)

The choice index, or -1 to disable. Default is -1.

Remarks

This allows the NPC to present a specific dialogue choice when the player interacts with them at this location. Set to -1 to use the default choice behavior.

Destination

Gets or sets the world position where the NPC should walk to.

```
public Vector3 Destination { get; set; }
```

Property Value

[Vector3](#)

The destination coordinates in world space.

FaceDestinationDirection

Gets or sets whether the NPC should face the destination direction when walking.

```
public bool FaceDestinationDirection { get; set; }
```

Property Value

[bool](#) ↗

`true` to make the NPC face the destination; otherwise, `false`. Default is `true`.

Forward

Gets or sets the optional forward direction for the destination marker.

```
public Vector3? Forward { get; set; }
```

Property Value

[Vector3?](#)

The forward direction vector, or `null` to auto-calculate from NPC position.

Remarks

If specified, this vector will be used to orient the destination marker. If not specified, the direction will be calculated from the NPC's current position to the destination.

GreetingOverrideToEnable

Gets or sets the greeting override index to enable when the NPC reaches the destination.

```
public int GreetingOverrideToEnable { get; set; }
```

Property Value

[int](#) ↗

The greeting override index, or -1 to disable. Default is -1.

Remarks

This allows the NPC to use a specific greeting override when the player interacts with them at this location. Set to -1 to use the default greeting behavior.

Name

Gets or sets the optional name for this action.

```
public string Name { get; set; }
```

Property Value

[string](#)

The action name, or `null` to use the default name "LocationDialogue".

StartTime

Gets or sets the time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

The start time in minutes (0-1439 for a 24-hour day).

WarpIfSkipped

Gets or sets whether the NPC should be warped to the destination if the action is skipped.

```
public bool WarpIfSkipped { get; set; }
```

Property Value

[bool](#) ↗

`true` to warp if skipped; otherwise, `false`. Default is `false`.

Remarks

If the action is skipped (e.g., due to time jumps or schedule changes), setting this to `true` will teleport the NPC directly to the destination.

Within

Gets or sets the distance threshold within which the NPC is considered to have arrived.

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

Property Value

[float](#) ↗

The arrival threshold in world units. Default is 1.0f.

Remarks

The NPC will stop walking and consider the action complete when they are within this distance of the destination. Must be greater than 0.01f.

Class PrefabScheduleBuilder

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Plan-time schedule builder used during prefab composition. Collects [IScheduleActionSpec](#) entries without requiring a live NPC instance.

```
public sealed class PrefabScheduleBuilder
```

Inheritance

[object](#) ← PrefabScheduleBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

IMPORTANT: Avoid scheduling multiple actions at start time 0 (midnight). The game's action sorting comparator has a bug that can cause inconsistent sort results when multiple non-signal actions share the same start time. This issue is most commonly encountered at time 0 when using [EnsureDealSignal\(\)](#) which creates a signal at time 0.

To avoid this issue, schedule your first action at time 1 or later (e.g., 10 minutes = 0:10 AM).

Constructors

PrefabScheduleBuilder()

```
public PrefabScheduleBuilder()
```

Methods

Add(IScheduleActionSpec)

Adds a custom schedule action using an S1API action specification.

```
public PrefabScheduleBuilder Add(IScheduleActionSpec spec)
```

Parameters

spec [IScheduleActionSpec](#)

The action specification to add to the prefab schedule.

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method allows adding custom schedule actions using the [IScheduleActionSpec](#) interface. The specification will be stored and applied when the prefab is configured.

If the specification is `null`, this method does nothing and returns the builder unchanged.

DriveToCarPark(ParkingLotWrapper, LandVehicle, int, ParkingAlignment?, bool?, string)

Adds a "Drive to Car Park" action that makes the NPC drive a vehicle to a parking lot using wrapper objects.

```
public PrefabScheduleBuilder DriveToCarPark(ParkingLotWrapper lot, LandVehicle vehicle, int startTime, ParkingAlignment? alignment = null, bool? overrideParkingType = null, string name = null)
```

Parameters

lot [ParkingLotWrapper](#)

The parking lot wrapper where the vehicle should be parked.

vehicle [LandVehicle](#)

The vehicle wrapper that should be driven to the parking lot.

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

alignment [ParkingAlignment?](#)

The optional parking alignment to use when parking the vehicle.

overrideParkingType [bool](#)?

Whether to override the default parking type behavior.

name [string](#)

The optional name for this action. If [null](#), uses "DriveToCarPark".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates a [DriveToCarParkSpec](#) that will be applied when the prefab is configured. The specification will create a ScheduleOne.NPCs.Schedules.NPCTrigger_DriveToCarPark action that makes the NPC drive the specified vehicle to the designated parking lot and park it.

DriveToCarPark(string, string, int, ParkingAlignment?, bool?, string)

Adds a "Drive to Car Park" action that makes the NPC drive a vehicle to a parking lot using GUIDs.

```
public PrefabScheduleBuilder DriveToCarPark(string parkingLotGUID, string vehicleGUID, int startTime, ParkingAlignment? alignment = null, bool? overrideParkingType = null, string name = null)
```

Parameters

parkingLotGUID [string](#)

The GUID of the parking lot where the vehicle should be parked.

vehicleGUID [string](#)

The GUID of the vehicle that should be driven to the parking lot.

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

alignment [ParkingAlignment?](#)

The optional parking alignment to use when parking the vehicle.

overrideParkingType [bool](#)?

Whether to override the default parking type behavior.

name [string](#)

The optional name for this action. If [null](#), uses "DriveToCarPark".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates a [DriveToCarParkSpec](#) that will be applied when the prefab is configured. The specification will create a ScheduleOne.NPCs.Schedules.NPCSignal_DriveToCarPark action that makes the NPC drive the specified vehicle to the designated parking lot and park it. The GUIDs will be resolved to their corresponding objects at runtime.

DriveToCarParkByName(string, string, int, ParkingAlignment?, bool?, string)

Adds a "Drive to Car Park" action using GameObject names for runtime resolution. More reliable than GUIDs for modders as names are more predictable across different players.

```
public PrefabScheduleBuilder DriveToCarParkByName(string parkingLotName, string  
vehicleName, int startTime, ParkingAlignment? alignment = null, bool?  
overrideParkingType = null, string name = null)
```

Parameters

parkingLotName [string](#) ↗

The GameObject name of the parking lot where the vehicle should be parked.

vehicleName [string](#) ↗

The GameObject name of the vehicle that should be driven.

startTime [int](#) ↗

The time when this action should start, in minutes from midnight (0-1439).

alignment [ParkingAlignment?](#)

The optional parking alignment to use when parking the vehicle.

overrideParkingType [bool](#) ↗?

Whether to override the default parking type behavior.

name [string](#) ↗

The optional name for this action. If [null](#), uses "DriveToCarPark".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates a [DriveToCarParkSpec](#) that uses name-based resolution. The parking lot and vehicle will be found by GameObject name at runtime, which is more reliable than GUID-based lookup for mods that need to work across different players.

DriveToCarParkWithCreateVehicle(string, string, int, Vector3, Quaternion?, ParkingAlignment?, bool?, string)

Adds a "Drive to Car Park" action that creates a vehicle using a vehicle code. Useful when you don't have an existing vehicle to reference.

```
public PrefabScheduleBuilder DriveToCarParkWithCreateVehicle(string parkingLotName,  
string vehicleCode, int startTime, Vector3 vehicleSpawnPosition, Quaternion?  
vehicleSpawnRotation = null, ParkingAlignment? alignment = null, bool?  
overrideParkingType = null, string name = null)
```

Parameters

parkingLotName [string](#)

The GameObject name of the parking lot where the vehicle should be parked.

vehicleCode [string](#)

The vehicle code to create (e.g., "Sedan", "SUV", etc.).

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

vehicleSpawnPosition [Vector3](#)

The world position where the vehicle should spawn.

vehicleSpawnRotation [Quaternion?](#)

The optional rotation for the vehicle spawn. If not specified, uses identity rotation.

alignment [ParkingAlignment?](#)

The optional parking alignment to use when parking the vehicle.

`overrideParkingType` `bool`?

Whether to override the default parking type behavior.

`name` `string`

The optional name for this action. If `null`, uses "DriveToCarPark".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates a [DriveToCarParkSpec](#) that will create a new vehicle instance at runtime using the provided vehicle code. This is useful when you don't have an existing vehicle to reference but need the NPC to drive somewhere.

The vehicle will be spawned at the specified `vehicleSpawnPosition` and rotation when the schedule action executes.

[EnsureDealSignal\(\)](#)

Ensures that a customer deal signal exists under the schedule for handling deal interactions.

```
public PrefabScheduleBuilder EnsureDealSignal()
```

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates an [EnsureDealSignalSpec](#) that will be applied when the prefab is configured. The specification ensures that a ScheduleOne.NPCs.Schedules.NPCTrigger_WaitForDelivery component exists

on the NPC's schedule manager for proper customer deal handling.

HandleDeal(int, string)

Adds a handle-deal action for dealer-type NPCs.

```
[Obsolete("HandleDeal is no longer needed as of game version 0.4.2f4. Deal handling  
is now automatic through DealerAttendDealBehaviour.")]  
public PrefabScheduleBuilder HandleDeal(int startTime, string name = null)
```

Parameters

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

name [string](#)

Optional custom name for this action; defaults to "HandleDeal".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

As of v0.4.2f4, deal handling is now automatic through the DealerAttendDealBehaviour system. This method is kept for backwards compatibility but is a no-op. Dealer NPCs set up with EnsureDealer() will automatically handle deals when contracts are assigned.

LocationDialogue(Vector3, int, bool, float, bool, int, int, string)

Adds a location-dialogue action that moves to a destination and enables dialogue.

```
public PrefabScheduleBuilder LocationDialogue(Vector3 destination, int startTime,  
bool faceDestinationDir = true, float within = 1, bool warpIfSkipped = false, int  
greetingOverrideToEnable = -1, int choiceToEnable = -1, string name = null)
```

Parameters

destination Vector3

The world position where the NPC should walk to.

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

faceDestinationDir [bool](#)

Whether the NPC should face the destination direction when walking. Default is [true](#).

within [float](#)

The distance threshold (in world units) within which the NPC is considered to have arrived. Default is 1.0f.

warpIfSkipped [bool](#)

Whether to warp the NPC to the destination if the action is skipped. Default is [false](#).

greetingOverrideToEnable [int](#)

Greeting override index to enable upon arrival; use -1 to disable. Default is -1.

choiceToEnable [int](#)

Choice index to enable upon arrival; use -1 to disable. Default is -1.

name [string](#)

Optional custom name for this action; defaults to "LocationDialogue".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

Creates a [LocationDialogueSpec](#) that, when applied, configures a ScheduleOne.NPCs.Schedules.NPCEvent_LocationDialogue. The NPC walks to the destination and then sets dialogue-related overrides for player interaction.

SitAtSeatSet(string, int, bool, string)

Adds a seating action that moves the NPC to an available seat within the specified seat set.

```
public PrefabScheduleBuilder SitAtSeatSet(string seatSetName, int startTime, bool  
warpIfSkipped = false, string name = null)
```

Parameters

seatSetName [string](#)

The GameObject name of the [AvatarSeatSet](#) to use.

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

warpIfSkipped [bool](#)

Whether the NPC should be warped to the seat if the action is skipped. Default is [false](#).

name [string](#)

Optional custom name for this action; defaults to "Sit".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This convenience overload creates a [SitSpec](#) that resolves a seat set by name. For more advanced lookup scenarios (GUIDs, transform paths, direct references) instantiate [SitSpec](#) manually and add it via [Add\(IScheduleActionSpec\)](#).

StayInBuilding(Building, int, int, int?, string)

Adds a "Stay in Building" action that makes the NPC remain inside a building for a specified duration.

```
public PrefabScheduleBuilder StayInBuilding(Building building, int startTime, int durationMinutes = 60, int? doorIndex = null, string name = null)
```

Parameters

building [Building](#)

The building wrapper where the NPC should stay.

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439). **Avoid using 0** to prevent sort comparison issues.

durationMinutes [int](#)

The duration for which the NPC should remain in the building. Default is 60 minutes.

doorIndex [int?](#)

The optional door index to use when entering the building.

name [string](#)

The optional name for this action. If [null](#), uses "StayInBuilding".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates a [StayInBuildingSpec](#) that will be applied when the prefab is configured. The specification will create a ScheduleOne.NPCs.Schedules.NPCEvent_StayInBuilding action that keeps the NPC inside the specified building for the given duration.

If the building is `null`, this method does nothing and returns the builder unchanged.

UseATM(int, string, string)

Adds an ATM usage action at the specified time.

```
public PrefabScheduleBuilder UseATM(int startTime, string atmGUID = null, string name = null)
```

Parameters

`startTime` [int](#)

The time when this action should start, in minutes from midnight (0-1439).

`atmGUID` [string](#)

Optional GUID of a specific ATM to use; if null, the ATM should be resolved by gameplay systems.

`name` [string](#)

Optional custom name for this action; defaults to "UseATM".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

Creates a [UseATMSpec](#) that, when applied, configures a ScheduleOne.NPCs.Schedules.NPCSignal_UseATM under the NPC's schedule manager.

UseSlotMachine(int, Vector3, int, GamblingSessionMode, float, string)

Adds a slot machine usage action at the specified time.

```
public PrefabScheduleBuilder UseSlotMachine(int startTime, Vector3 machinePosition,  
    int betAmount = 10, GamblingSessionMode sessionMode =  
    GamblingSessionMode.SingleSpin, float maxSearchDistance = 5, string name = null)
```

Parameters

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

machinePosition [Vector3](#)

The world position of the slot machine to use.

betAmount [int](#)

The amount to bet in dollars (default: 10).

sessionMode [GamblingSessionMode](#)

The gambling session mode (default: single spin).

maxSearchDistance [float](#)

Maximum distance to search for a slot machine from the position (default: 5.0).

name [string](#)

Optional custom name for this action; defaults to "UseSlotMachine".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

Creates a [UseSlotMachineSpec](#) that makes the NPC walk to the slot machine location and play it according to the session mode. The NPC must have sufficient cash in their inventory to place bets.

UseSlotMachineMultipleTimes(int, Vector3, int, int, float, float, string)

Adds a slot machine usage action that plays multiple spins.

```
public PrefabScheduleBuilder UseSlotMachineMultipleTimes(int startTime, Vector3  
machinePosition, int spinCount, int betAmount = 10, float timeBetweenSpins = 10,  
float maxSearchDistance = 5, string name = null)
```

Parameters

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

machinePosition [Vector3](#)

The world position of the slot machine to use.

spinCount [int](#)

The number of spins to play.

betAmount [int](#)

The amount to bet per spin in dollars (default: 10).

timeBetweenSpins [float](#)

Time to wait between spins in seconds (default: 10.0).

maxSearchDistance [float](#)

Maximum distance to search for a slot machine from the position (default: 5.0).

name [string](#)

Optional custom name for this action; defaults to "UseSlotMachine".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

The NPC will play the specified number of spins, waiting between each one. If the NPC runs out of cash before completing all spins, the session will end early.

UseSlotMachineUntilBroke(int, Vector3, int, float, float, string)

Adds a slot machine usage action that plays until the NPC runs out of cash.

```
public PrefabScheduleBuilder UseSlotMachineUntilBroke(int startTime, Vector3  
machinePosition, int betAmount = 10, float timeBetweenSpins = 10, float  
maxSearchDistance = 5, string name = null)
```

Parameters

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

machinePosition [Vector3](#)

The world position of the slot machine to use.

betAmount [int](#)

The amount to bet per spin in dollars (default: 10).

timeBetweenSpins [float](#)

Time to wait between spins in seconds (default: 10.0).

maxSearchDistance [float](#)

Maximum distance to search for a slot machine from the position (default: 5.0).

name [string](#)

Optional custom name for this action; defaults to "UseSlotMachine".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

The NPC will gamble continuously until they no longer have enough cash for another bet. This can result in the NPC gambling away all their money, so ensure they have a reasonable amount of cash or use time-based limits instead.

UseSlotMachineUntilTime(int, int, Vector3, int, float, bool, float, Building, string)

Adds a slot machine usage action that plays until a specific time.

```
public PrefabScheduleBuilder UseSlotMachineUntilTime(int startTime, int endTime,
Vector3 machinePosition, int betAmount = 10, float timeBetweenSpins = 10, bool
stopIfBroke = true, float maxSearchDistance = 5, Building building = null, string
name = null)
```

Parameters

startTime [int](#)

The time when this action should start, in minutes from midnight (0-1439).

endTime [int](#)

The time when gambling should stop, in minutes from midnight (0-1439).

machinePosition [Vector3](#)

The world position of the slot machine to use.

betAmount [int](#)

The amount to bet per spin in dollars (default: 10).

timeBetweenSpins [float](#)

Time to wait between spins in seconds (default: 10.0).

stopIfBroke [bool](#)

If true, stops gambling when out of cash; if false, only stops at end time (default: true).

maxSearchDistance [float](#)

Maximum distance to search for a slot machine from the position (default: 5.0).

building [Building](#)

name [string](#)

Optional custom name for this action; defaults to "UseSlotMachine".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

The NPC will gamble continuously until the end time is reached. If `stopIfBroke` is true, the session will also end if the NPC runs out of cash.

UseVendingMachine(int, string, string)

Adds a vending machine usage action at the specified time.

```
public PrefabScheduleBuilder UseVendingMachine(int startTime, string machineGUID = null, string name = null)
```

Parameters

startTime [int ↗](#)

The time when this action should start, in minutes from midnight (0-1439).

machineGUID [string ↗](#)

Optional GUID of a specific vending machine to use; if null, the nearest reachable machine will be used.

name [string ↗](#)

Optional custom name for this action; defaults to "UseVending".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

Creates a [UseVendingMachineSpec](#) that, when applied, configures a ScheduleOne.NPCs.Schedules.NPCSignal_UseVendingMachine under the NPC's schedule manager.

WalkTo(Vector3, int, bool, float, bool, Vector3?, string)

Adds a walk-to action that moves the NPC to a specific world position at the given start time.

```
public PrefabScheduleBuilder WalkTo(Vector3 destination, int startTime, bool faceDestinationDir = true, float within = 1, bool warpIfSkipped = false, Vector3? forward = null, string name = null)
```

Parameters

destination Vector3

The world position where the NPC should walk to.

startTime [int ↗](#)

The time when this action should start, in minutes from midnight (0-1439). **Avoid using 0** to prevent sort comparison issues.

faceDestinationDir [bool](#)

Whether the NPC should face the destination direction when walking. Default is `true`.

within [float](#)

The distance threshold within which the NPC is considered to have arrived. Default is 1.0f.

warpIfSkipped [bool](#)

Whether the NPC should be warped to the destination if the action is skipped. Default is `false`.

forward [Vector3](#)

The optional forward direction vector for the destination marker. If `null`, the direction will be calculated from the NPC's position to the destination.

name [string](#)

The optional name for this action. If `null`, uses "WalkTo".

Returns

[PrefabScheduleBuilder](#)

This builder instance for method chaining.

Remarks

This method creates a [WalkToSpec](#) that will be applied when the prefab is configured. The specification is stored and will create a ScheduleOne.NPCs.Schedules.NPCSignal_WalkToLocation action when applied to the actual NPC schedule.

If `forward` is specified, it will be used to orient the destination marker. Otherwise, the direction will be automatically calculated from the NPC's current position to the destination.

Class SitSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that seats an NPC at an ScheduleOne.AvatarFramework.Animation.AvatarSeatSet.

```
public sealed class SitSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← SitSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

The specification resolves an ScheduleOne.AvatarFramework.Animation.AvatarSeatSet using one of the provided lookup mechanisms and configures a ScheduleOne.NPCs.Schedules.NPCEvent_Sit instance on the target schedule. This allows mods to declaratively schedule seating behaviour through the S1API builder.

Constructors

SitSpec()

```
public SitSpec()
```

Properties

IncludeInactiveSearch

Gets or sets whether lookups should consider inactive seat sets. Defaults to `true`.

```
public bool IncludeInactiveSearch { get; set; }
```

Property Value

[bool](#)

Name

Gets or sets the optional display name for this action. Defaults to "Sit".

```
public string Name { get; set; }
```

Property Value

[string](#)

SeatSetName

Gets or sets the GameObject name of the desired seat set.

```
public string SeatSetName { get; set; }
```

Property Value

[string](#)

Remarks

Name lookups search all seat sets in the scene and are case-insensitive.

SeatSetPath

Gets or sets a transform path (e.g. "@Locations/Cafe/Seats/Booth01") used to locate the seat set.

```
public string SeatSetPath { get; set; }
```

Property Value

[string](#)

Remarks

Paths are matched case-insensitively against full transform hierarchies. Inactive objects are included when [IncludeInactiveSearch](#) is `true`.

SeatSetReference

Gets or sets a direct Unity object reference that contains the desired seat set.

```
public Object SeatSetReference { get; set; }
```

Property Value

[Object](#)

Remarks

The reference may be an `ScheduleOne.AvatarFramework.Animation.AvatarSeatSet`, a `UnityEngine.GameObject`, or any `UnityEngine.Component` that is part of the seat set hierarchy.

StartTime

Gets or sets the start time for this action, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

WarpIfSkipped

Gets or sets whether the NPC should be warped to the seat if the action is skipped.

```
public bool WarpIfSkipped { get; set; }
```

Property Value

[bool](#)

Class StayInBuildingSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that makes an NPC remain inside a building for a specified duration.

```
public sealed class StayInBuildingSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← StayInBuildingSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This action creates a ScheduleOne.NPCs.Schedules.NPCEvent_StayInBuilding that will keep the NPC inside the specified building for the given duration. The building can be identified by name-based lookup.

Constructors

StayInBuildingSpec()

```
public StayInBuildingSpec()
```

Properties

BuildingGUID

Gets or sets the GUID of the building where the NPC should stay.

```
public string BuildingGUID { get; set; }
```

Property Value

[string](#) ↗

The building GUID, or `null` if using name-based lookup.

Remarks

The building GUID is typically generated at runtime and may not be stable across game sessions. For modder-facing APIs prefer using name-based lookup via `Building.GetName(string)` or typed identifiers via `Building.Get<T>()`. Use the GUID only if you have a reliable runtime reference to the exact game object.

BuildingName

Gets or sets the name of the building where the NPC should stay.

```
public string BuildingName { get; set; }
```

Property Value

[string](#) ↗

The building name, or `null` if using GUID-based lookup.

Remarks

The building name takes precedence over [BuildingGUID](#) and is the recommended identifier for mod developers. It should match a building registered in the S1API building registry (see `Building.GetName(string)` and `Building.Get<T>()`). Names are stable across game sessions and preferred for persistence and prefab configuration.

DoorIndex

Gets or sets the optional door index to use when entering the building.

```
public int? DoorIndex { get; set; }
```

Property Value

[int](#)?

The door index, or `null` to use the default entrance.

Remarks

Some buildings have multiple entrances. This specifies which door the NPC should use when entering the building. If not specified, the default entrance is used.

DurationMinutes

Gets or sets the duration for which the NPC should remain in the building.

```
public int DurationMinutes { get; set; }
```

Property Value

[int](#)

The duration in minutes. Default is 60 minutes.

Remarks

The NPC will stay in the building for this duration starting from the start time. Must be at least 1 minute.

Name

Gets or sets the optional name for this action.

```
public string Name { get; set; }
```

Property Value

[string](#)

The action name, or `null` to use the default name "StayInBuilding".

StartTime

Gets or sets the time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

The start time in minutes (0-1439 for a 24-hour day).

Methods

ApplyTo(NPCSchedule)

Applies this action specification to the given NPC schedule.

```
public void ApplyTo(NPCSchedule schedule)
```

Parameters

`schedule` [NPCSchedule](#)

The NPC schedule to apply this action to.

Remarks

This method should create and configure the appropriate schedule action on the provided schedule instance. The implementation should handle any necessary object resolution, validation, and error handling.

Class UseATMSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that makes an NPC use an ATM at a scheduled time.

```
public sealed class UseATMSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← UseATMSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This action creates a ScheduleOne.NPCs.Schedules.NPCTSignal_UseATM that will make the NPC walk to and interact with an ATM. If [ATMGUID](#) is not set the action expects ATM to be assigned at runtime by gameplay systems.

Constructors

UseATMSpec()

```
public UseATMSpec()
```

Properties

ATMGUID

Optional specific ATM GUID to target.

```
public string ATMGUID { get; set; }
```

Property Value

[string](#)

Name

Optional custom name for the action.

```
public string Name { get; set; }
```

Property Value

[string](#)

StartTime

The time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

Class UseSlotMachineSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that makes an NPC use a slot machine at a scheduled time. Supports single spins, multiple spins, time-based sessions, or gambling until broke.

```
public sealed class UseSlotMachineSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← UseSlotMachineSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This action creates a custom schedule entry that makes the NPC walk to a slot machine location and play it according to the specified session mode. The NPC will use cash from their inventory, and any winnings will be added back to their inventory.

Constructors

UseSlotMachineSpec()

```
public UseSlotMachineSpec()
```

Properties

BetAmount

Gets or sets the bet amount in dollars.

```
public int BetAmount { get; set; }
```

Property Value

[int](#)

The amount to bet per spin, in dollars.

Remarks

Common bet amounts are 5, 10, 25, 50, or 100. The NPC must have this amount in their inventory (as cash items) to use the machine.

Building

Gets or sets the optional building that contains the slot machine.

```
public Building Building { get; set; }
```

Property Value

[Building](#)

The building wrapper, or `null` to auto-detect or skip building entry.

Remarks

If specified, the NPC will first enter this building before walking to the slot machine. This is useful when the slot machine is inside a building and the NPC needs to enter first. If not specified, the system will attempt to pathfind directly to the slot machine position.

EndTime

Gets or sets the end time in minutes from midnight when using time-based session modes.

```
public int EndTime { get; set; }
```

Property Value

[int ↗](#)

The end time in minutes (0-1439 for a 24-hour day). Default is 0.

Remarks

Used with [UntilTime](#) and [UntilTimeOrBroke](#).

MachinePosition

Gets or sets the world position of the slot machine to use.

```
public Vector3 MachinePosition { get; set; }
```

Property Value

[Vector3](#)

The 3D position of the slot machine in world space.

Remarks

The NPC will search for the nearest slot machine to this position. Ensure the position is accurate to avoid the NPC using the wrong machine.

MaxSearchDistance

Gets or sets the maximum search distance from the machine position.

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

Property Value

[float](#)

The search radius in world units. Default is 5.0.

Name

Gets or sets the optional name for this action.

```
public string Name { get; set; }
```

Property Value

[string](#)

The action name, or [null](#) to use the default name "UseSlotMachine".

SessionMode

Gets or sets the gambling session mode.

```
public GamblingSessionMode SessionMode { get; set; }
```

Property Value

[GamblingSessionMode](#)

The session mode that determines how long the NPC gambles. Default is [SingleSpin](#).

Remarks

[SingleSpin](#): Play once and stop.

[SpinCount](#): Play until [SpinCount](#) spins are completed.

[UntilTime](#): Play until [EndTime](#) is reached.

[UntilBroke](#): Play until the NPC can't afford another bet.

[UntilTimeOrBroke](#): Play until [EndTime](#) OR out of cash.

SpinCount

Gets or sets the number of spins to play when using [SpinCount](#).

```
public int SpinCount { get; set; }
```

Property Value

[int](#)

The number of spins. Default is 1.

StartTime

Gets or sets the time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

The start time in minutes (0-1439 for a 24-hour day).

TimeBetweenSpins

Gets or sets the time to wait between spins in seconds.

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

Property Value

[float](#)

The delay between spins in seconds. Default is 10.0 seconds.

Remarks

This delay occurs after the slot machine finishes spinning and displaying results, before the NPC starts the next spin. This makes the gambling behavior more realistic.

Class UseVendingMachineSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies an action that makes an NPC use a vending machine at a scheduled time.

```
public sealed class UseVendingMachineSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← UseVendingMachineSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This action creates a ScheduleOne.NPCs.Schedules.NPCTSignal_UseVendingMachine that will make the NPC interact with a vending machine. The machine can be specified by GUID, or if not specified, the NPC will use the nearest available vending machine.

Constructors

UseVendingMachineSpec()

```
public UseVendingMachineSpec()
```

Properties

MachineGUID

Gets or sets the GUID of the specific vending machine to use.

```
public string MachineGUID { get; set; }
```

Property Value

[string](#) ↗

The vending machine GUID, or `null` to use the nearest machine.

Remarks

If specified, the NPC will use this specific vending machine. If not specified or if the machine cannot be found, the NPC will use the nearest available vending machine instead.

Name

Gets or sets the optional name for this action.

```
public string Name { get; set; }
```

Property Value

[string](#) ↗

The action name, or `null` to use the default name "UseVending".

StartTime

Gets or sets the time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int ↗](#)

The start time in minutes (0-1439 for a 24-hour day).

Class WalkToSpec

Namespace: [S1API.Entities.Schedule](#)

Assembly: S1API.dll

Specifies a walk-to action that moves an NPC to a specific world position at a scheduled time.

```
public sealed class WalkToSpec : IScheduleActionSpec
```

Inheritance

[object](#) ← WalkToSpec

Implements

[IScheduleActionSpec](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This action creates a ScheduleOne.NPCs.Schedules.NPCTSignal_WalkToLocation that will make the NPC walk to the specified destination. The NPC will face the destination direction by default and will be considered to have arrived when within the specified threshold distance.

Constructors

WalkToSpec()

```
public WalkToSpec()
```

Properties

Destination

Gets or sets the world position where the NPC should walk to.

```
public Vector3 Destination { get; set; }
```

Property Value

Vector3

The destination coordinates in world space.

FaceDestinationDirection

Gets or sets whether the NPC should face the destination direction when walking.

```
public bool FaceDestinationDirection { get; set; }
```

Property Value

[bool](#)

`true` to make the NPC face the destination; otherwise, `false`. Default is `true`.

Forward

Gets or sets the optional forward direction for the destination marker.

```
public Vector3? Forward { get; set; }
```

Property Value

Vector3?

The forward direction vector, or `null` to auto-calculate from NPC position to destination.

Remarks

If specified, this vector will be used to orient the destination marker. If not specified, the direction will be calculated from the NPC's current position to the destination.

Name

Gets or sets the optional name for this action.

```
public string Name { get; set; }
```

Property Value

[string](#)

The action name, or `null` to use the default name "WalkTo".

StartTime

Gets or sets the time when this action should start, in minutes from midnight.

```
public int StartTime { get; set; }
```

Property Value

[int](#)

The start time in minutes (0-1439 for a 24-hour day).

WarpIfSkipped

Gets or sets whether the NPC should be warped to the destination if the action is skipped.

```
public bool WarpIfSkipped { get; set; }
```

Property Value

[bool](#)

`true` to warp if skipped; otherwise, `false`. Default is `false`.

Remarks

If the action is skipped (e.g., due to time jumps or schedule changes), setting this to `true` will teleport the NPC directly to the destination.

Within

Gets or sets the distance threshold within which the NPC is considered to have arrived.

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

Property Value

[float](#)

The arrival threshold in world units. Default is 1.0f.

Remarks

The NPC will stop walking and consider the action complete when they are within this distance of the destination. Must be greater than 0.01f.

Class NPC

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Abstract base class for creating custom NPCs with modular architecture supporting both physical and non-physical NPCs. Physical NPCs are visible in the game world with 3D models, movement, and direct interaction. Non-physical NPCs are invisible contacts primarily used for messaging and phone interactions.

```
public abstract class NPC : Saveable, IEntity, IHealth
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← NPC

Implements

[IEntity](#), [IHealth](#)

Derived

[DanSamwell](#), [AnnaChesterfield](#), [BillyKramer](#), [CrankyFrank](#), [GenghisBarn](#), [JaneLucero](#), [JavierPerez](#),
[LisaGardener](#), [MacCooper](#), [MarcoBaron](#), [MelissaWood](#), [SalvadorMoreno](#), [BradCrosby](#), [ElizabethHomley](#),
[EugeneBuckley](#), [GregFliggle](#), [JeffGilmore](#), [JenniferRivera](#), [KevinOakley](#), [LouisFourier](#), [LucyPennington](#),
[PhilipWentworth](#), [RandyCaulfield](#), [IgorRomanovich](#), [MannyOakfield](#), [AlbertHoover](#), [AustinSteiner](#),
[BenjiColeman](#), [BethPenn](#), [ChloeBowers](#), [DonnaMartin](#), [GeraldinePoon](#), [JessiWaters](#), [KathyHenderson](#),
[KyleCooley](#), [LudwigMeyer](#), [MickLubbin](#), [Ming](#), [PeggyMyers](#), [PeterFile](#), [SamThompson](#), [OscarHolland](#),
[OfficerBailey](#), [OfficerCooper](#), [OfficerGreen](#), [OfficerHoward](#), [OfficerJackson](#), [OfficerLee](#), [OfficerLopez](#),
[OfficerMurphy](#), [OfficerOakley](#), [StanCarney](#), [AlisonKnight](#), [CarlBundy](#), [ChrisSullivan](#), [DennisKennedy](#),
[HankStevenson](#), [HaroldColt](#), [JackKnight](#), [JackieStevenson](#), [JeremyWilkinson](#), [KarenKennedy](#), [WeiLong](#),
[UncleNelson](#), [FionaHancock](#), [HerbertBleuball](#), [JenHeard](#), [LeoRivers](#), [LilyTurner](#), [MichaelBoog](#), [PearlMoore](#),
[RayHoffman](#), [TobiasWentworth](#), [WalterCussler](#), [CharlesRowland](#), [DeanWebster](#), [DorisLubbin](#),
[GeorgeGreene](#), [JerryMontero](#), [JoyceBall](#), [KeithWagner](#), [KimDelaney](#), [MegCooley](#), [MollyPresley](#),
[ShirleyWatts](#), [TrentSherman](#)

Inherited Members

[Saveable.RequestGameSave\(bool\)](#), [Saveable.OnLoaded\(\)](#), [Saveable.OnSaved\(\)](#), [Saveable.LoadOrder](#),
[Registerable.OnDestroyed\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#),
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

NPCs provide access to component systems: [Appearance](#), [Dialogue](#), [Schedule](#), [Customer](#), [Relationship](#), [Inventory](#), and [Movement](#). Customer, relationship, and schedule configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#) for proper save/load behavior.

Constructors

NPC()

Base constructor for a new NPC. Identity is configured via [ConfigurePrefab\(NPCPrefabBuilder\)](#) using [WithIdentity\(string, string, string\)](#) and optionally [WithIcon\(Sprite?\)](#).

```
protected NPC()
```

Remarks

Not intended for direct instancing. Create your derived class and let S1API handle instancing. Identity information (ID, firstName, lastName, icon) must be provided in [ConfigurePrefab\(NPCPrefabBuilder\)](#) using the builder methods.

NPC(string, string?, string?, Sprite?)

Backwards-compatible constructor for non-physical NPCs that provides identity directly via parameters. This constructor is intended for backwards compatibility with mods that used the old constructor pattern.

```
[Obsolete("Use the parameterless constructor and configure identity via  
ConfigurePrefab with NPCPrefabBuilder.WithIdentity. This constructor is provided for  
backwards compatibility with non-physical NPCs.")]  
protected NPC(string id, string? firstName, string? lastName, Sprite? icon = null)
```

Parameters

id [string](#)

Unique identifier for the NPC.

firstName [string](#)

The first name for the NPC.

lastName [string](#)

The last name for the NPC. Can be null.

icon [Sprite](#)

The icon sprite for the NPC. Can be null to use default.

Remarks

This constructor is marked as obsolete. For new code, use the parameterless constructor and configure identity via [ConfigurePrefab\(NPCPrefabBuilder\)](#) using [WithIdentity\(string, string, string\)](#) and optionally [WithIcon\(Sprite?\)](#).

This constructor is appropriate for non-physical NPCs (where [IsPhysical](#) returns [false](#)) that don't require prefab configuration. Physical NPCs should use [ConfigurePrefab\(NPCPrefabBuilder\)](#) for proper network spawn support.

Fields

All

List of all NPCs within the base game and modded.

```
public static readonly List<NPC> All
```

Field Value

[List<NPC>](#)

Responses

A list of text responses you've added to your NPC.

```
protected readonly List<Response> Responses
```

Field Value

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

Properties

Aggressiveness

How aggressive this NPC is towards others.

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

Property Value

[float](#)

Appearance

Access to the appearance customization system for visual avatar management.

```
public NPCAppearance Appearance { get; }
```

Property Value

[NPCAppearance](#)

CombatBehaviour

The current [CombatBehaviour](#) instance.

```
public CombatBehaviour CombatBehaviour { get; }
```

Property Value

[CombatBehaviour](#)

ConversationCanBeHidden

Set's whether the text message can be deleted/hidden

```
public bool ConversationCanBeHidden { get; set; }
```

Property Value

[bool](#)

CurrentHealth

The current health the NPC has.

```
public float CurrentHealth { get; }
```

Property Value

[float](#)

CurrentVehicle

The current vehicle the NPC is occupying, if any.

```
public LandVehicle? CurrentVehicle { get; }
```

Property Value

[LandVehicle](#)

CustomNpcsReady

Whether all custom NPCs have been instantiated and finalized. This flag is set to true once all custom NPC types have been spawned and initialized. Mods can check this to ensure custom NPCs are ready before performing operations that depend on them.

```
public static bool CustomNpcsReady { get; }
```

Property Value

[bool](#)

Customer

Access to the customer behavior system for NPCs that act as business customers.

```
public NPCCustomer Customer { get; }
```

Property Value

[NPCCustomer](#)

Dealer

Access to the dealer system for NPCs that act as product distributors.

```
public NPCDealer Dealer { get; }
```

Property Value

[NPCDealer](#)

Dialogue

Access to the dialogue system for interactive conversations and dialogue trees.

```
public NPCDialogue Dialogue { get; }
```

Property Value

[NPCDialogue](#)

FirstName

The first name of this NPC.

```
public string FirstName { get; set; }
```

Property Value

[string](#) ↗

FullName

The full name of this NPC. If there is no last name, it will just return the first name.

```
public string FullName { get; }
```

Property Value

[string](#) ↗

ID

The unique identifier to assign to this NPC. Used when saving and loading. Probably other things within the base game code.

```
public string ID { get; protected set; }
```

Property Value

[string](#) ↗

Icon

The icon assigned to this NPC.

```
public Sprite Icon { get; set; }
```

Property Value

Sprite

Inventory

Access to the inventory system for item management.

```
public NPCInventory Inventory { get; }
```

Property Value

[NPCInventory](#)

IsConscious

Whether the NPC is currently conscious or not.

```
public bool IsConscious { get; }
```

Property Value

[bool](#)

IsDead

Whether the NPC is dead or not.

```
public bool IsDead { get; }
```

Property Value

[bool](#)

IsDealer

Determines if the NPC has dealer functionality. Override as true for NPCs that should be dealers.

```
public virtual bool IsDealer { get; }
```

Property Value

[bool](#) ↗

Remarks

Dealer NPCs ([true](#)): Can manage customers, handle contracts, accept cash payments, and track inventory for sales. When true, the NPC prefab will use the "Dealer" network prefab instead of "CivilianNPC". Non-dealer NPCs ([false](#)): Regular NPCs without dealer-specific functionality.

IsInBuilding

Whether the NPC is currently inside a building or not.

```
public bool IsInBuilding { get; }
```

Property Value

[bool](#) ↗

IsInVehicle

Whether the NPC is currently inside a vehicle or not.

```
public bool IsInVehicle { get; }
```

Property Value

[bool](#) ↗

IsInvincible

Whether the NPC is invincible or not.

```
public bool IsInvincible { get; set; }
```

Property Value

[bool](#) ↗

IsKnockedOut

Whether the NPC is knocked out or not.

```
public bool IsKnockedOut { get; }
```

Property Value

[bool](#) ↗

IsPanicking

Whether the NPC is currently panicking or not.

```
public bool IsPanicking { get; }
```

Property Value

[bool](#) ↗

IsPhysical

Determines if the NPC is visible in the game world. Override as true for physical NPCs with 3D models, movement, and direct interaction.

```
public virtual bool IsPhysical { get; }
```

Property Value

[bool](#) ↗

Remarks

Physical NPCs ([true](#)): Visible in world, have collision detection, can move and follow schedules, can be damaged/healed. Non-physical NPCs ([false](#)): Invisible, primarily for messaging and phone contacts, cannot move or be directly interacted with.

IsUnsettled

Whether the NPC is currently unsettled or not.

```
public bool IsUnsettled { get; }
```

Property Value

[bool](#) ↗

IsVisible

UNCONFIRMED: Whether the NPC is currently visible to the player or not. If you confirm this, please let us know so we can update the documentation!

```
public bool IsVisible { get; }
```

Property Value

[bool](#) ↗

LastName

The last name of this NPC.

```
public string LastName { get; set; }
```

Property Value

[string](#)

MaxHealth

The maximum health the NPC has.

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

Property Value

[float](#)

Movement

Access to the movement system for controlling NPC movement and navigation.

```
public NPCMovement Movement { get; }
```

Property Value

[NPCMovement](#)

NPCId

Static NPC ID for this NPC type. Used to resolve connections during prefab configuration when NPC instances are not yet available. Override this in derived classes to provide the NPC ID string (e.g., "kyle_cooley", "ludwig_meyer").

```
public static string? NPCId { get; }
```

Property Value

[string](#) ↗

Remarks

For built-in NPC wrappers, this should return the ID string that matches the base game NPC. For custom NPCs, this should return the ID configured via [WithIdentity\(string, string, string\)](#).

PanicDuration

UNCONFIRMED: How long the NPC will panic for. If you confirm this, please let us know so we can update the documentation!

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

Property Value

[float](#) ↗

Position

The world position of the NPC.

```
public Vector3 Position { get; set; }
```

Property Value

Vector3

Region

The region the NPC is associated with. Note: Not the region they're in currently. Just the region they're designated to.

```
public Region Region { get; set; }
```

Property Value

[Region](#)

Relationship

Access to the relationship system for social connections and relationships with the player.

```
public NPCRelationship Relationship { get; }
```

Property Value

[NPCRelationship](#)

RequiresRegionUnlocked

UNCONFIRMED: Whether the NPC requires the region unlocked in order to deal to. If you confirm this, please let us know so we can update the documentation!

```
public bool RequiresRegionUnlocked { get; set; }
```

Property Value

[bool](#)

Scale

Sets the scale of the NPC.

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

Property Value

[float](#)

Schedule

Access to the schedule system for movement and activity scheduling.

```
public NPCSchedule Schedule { get; }
```

Property Value

[NPCSchedule](#)

Transform

The transform of the NPC. Please do not set the properties of this transform.

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

Property Value

[Transform](#)

gameObject

INTERNAL: Tracking for the GameObject associated with this NPC. Not intended for use by modders!

```
public GameObject gameObject { get; }
```

Property Value

GameObject

Methods

ClearConversationCategories()

Clears the NPC's conversation categories, removing any badge (C/S/D) from the messages UI. This makes the NPC appear like Uncle Nelson - present in messages but without a category badge.

```
public void ClearConversationCategories()
```

ConfigurePrefab(NPCPrefabBuilder)

Override to configure NPC components and default behavior before the NPC is spawned. Called during prefab creation to set up spawn position, customer behavior, relationships, and schedules.

```
protected virtual void ConfigurePrefab(NPCPrefabBuilder builder)
```

Parameters

builder [NPCPrefabBuilder](#)

Prefab builder for configuring this NPC type.

Remarks

Customer, relationship, and schedule configuration must be done here for proper save/load behavior and network compatibility. Use the builder pattern for fluent configuration. Runtime initialization should be done in [OnCreated\(\)](#) instead.

Damage(int)

Deals damage to the NPC.

```
public void Damage(int amount)
```

Parameters

amount [int](#)

The amount of damage to deal.

Get<T>()

Gets the instance of an NPC. Supports base NPCs as well as other mod NPCs. For base NPCs, [S1API.Entities.NPCs](#).

```
public static NPC? Get<T>() where T : NPC
```

Returns

[NPC](#)

Type Parameters

T

The NPC class to get the instance of.

Goto(Vector3)

Tells the NPC to travel to a specific position in world space.

```
public void Goto(Vector3 position)
```

Parameters

position Vector3

The position to travel to.

Heal(int)

Heals the NPC.

```
public void Heal(int amount)
```

Parameters

amount int ↗

The amount of health to heal.

Kill()

Kills the NPC.

```
public void Kill()
```

KnockOut()

Knocks the NPC out. NOTE: Does not work for invincible NPCs.

```
public void KnockOut()
```

LerpScale(float, float)

Smoothly scales the NPC over lerpTime.

```
public void LerpScale(float scale, float lerpTime)
```

Parameters

scale [float](#)

The scale you want set.

lerpTime [float](#)

The time to scale over.

OnCreated()

Called when the NPC is fully created and spawned. Override for runtime initialization after all components are set up. Use this to configure appearance, dialogue systems, subscribe to events, enable schedule system, and set basic properties.

```
protected override void OnCreated()
```

Remarks

Called after the NPC is instantiated and all components are initialized. Appearance, dialogue, and schedule setup should be done here rather than in the constructor.

OnResponseLoaded(Response)

Called when a text message response is loaded from the save file. Override to re-attach callbacks to loaded responses.

```
protected virtual void OnResponseLoaded(Response response)
```

Parameters

response [Response](#)

The response that was loaded from save data.

Panic()

Causes the NPC to become panicked.

```
public void Panic()
```

PreRegisterAllNpcPrefabs()

Scans loaded assemblies for subclasses of S1API.Entities.NPC and pre-registers their prefabs.

```
public static void PreRegisterAllNpcPrefabs()
```

PreRegisterPrefabForType(Type)

Pre-registers a per-type NPC prefab into FishNet spawnables without creating a live instance. Should be called on both server and client before any NPC instances are spawned.

```
public static void PreRegisterPrefabForType(Type npcType)
```

Parameters

npcType [Type](#)

RefreshMessagingIcons()

```
public void RefreshMessagingIcons()
```

Revive()

Revives the NPC.

```
public void Revive()
```

SendTextMessage(string, Response[], float, bool)

Sends a text message from this NPC to the players. Supports responses with callbacks for additional logic.

```
public void SendTextMessage(string message, Response[]? responses = null, float responseDelay = 1, bool network = true)
```

Parameters

message [string](#)

The message you want the player to see. Unity rich text is allowed.

responses [Response](#)[]

Instances of [Response](#) to display.

responseDelay [float](#)

The delay between when the message is sent and when the player can reply.

network [bool](#)

Whether this should propagate to all players or not.

SetEquippable(string)

Sets an equippable item for the NPC.

```
public void SetEquippable(string assetPath)
```

Parameters

assetPath [string](#)

The asset path to the equippable item. UnityEngine.UI.Misc can be used here.

StopPanicking()

Causes the NPC to stop panicking, if they are currently.

```
public void StopPanicking()
```

Unsettle(float)

Causes the NPC to become unsettled. UNCONFIRMED: Will panic them for PanicDuration amount of time. If you confirm this, please let us know so we can update the documentation!

```
public void Unsettle(float duration)
```

Parameters

duration [float](#)

Length of time they should stay unsettled.

Events

OnDeath

Called when the NPC died.

```
public event Action OnDeath
```

Event Type

[Action ↗](#)

OnInventoryChanged

Called when the NPC's inventory contents change.

```
public event Action OnInventoryChanged
```

Event Type

[Action ↗](#)

Class NPCAppearance

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing appearance customization system for NPCs. Provides builders to configure visual appearance including physical features, clothing, and accessories. Appearance configuration is done in [OnCreated\(\)](#).

```
public class NPCAppearance
```

Inheritance

[object](#) ← NPCAppearance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Use the builder pattern to configure customization fields, face layers, body layers, and accessory layers. Always call [Build\(\)](#) at the end to generate the mugshot and apply the appearance to the avatar.

Methods

Build()

Finalizes the appearance by generating the NPC's mugshot. This can be called after setting all appearance attributes.

```
public NPCAppearance Build()
```

Returns

[NPCAppearance](#)

The [NPCAppearance](#) instance with the generated mugshot.

GenerateRandomAppearance()

Generates a random appearance for the S1API.Entities.NPCAppearance.NPC

```
public void GenerateRandomAppearance()
```

Set<T>(object)

Sets an appearance field within ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance Set<T>(object appearanceValue) where T : BaseAppearance
```

Parameters

appearanceValue [object](#)

The value to set

Returns

[NPCAppearance](#)

Type Parameters

T

The appearance type

WithAccessoryLayer<T>(string, uint)

Adds a Accessory Layer within the ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance WithAccessoryLayer<T>(string path, uint hexColor) where T : BaseAccessoryAppearance
```

Parameters

path [string](#)

The asset path

hexColor [uint](#)

The color in Hexadecimals

Returns

[NPCAppearance](#)

Type Parameters

T

WithAccessoryLayer<T>(string, Color)

Adds a Accessory Layer within the ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance WithAccessoryLayer<T>(string path, Color color) where T : BaseAccessoryAppearance
```

Parameters

path [string](#)

The asset path

color [Color](#)

The color instance

Returns

[NPCAppearance](#)

Type Parameters

T

WithBodyLayer<T>(string, uint)

Adds a Body Layer within the ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance WithBodyLayer<T>(string path, uint hexColor) where T : BaseBodyAppearance
```

Parameters

path [string](#)

The asset path

hexColor [uint](#)

The color in Hexadecimals

Returns

[NPCAppearance](#)

Type Parameters

T

WithBodyLayer<T>(string, Color)

Adds a Body Layer within the ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance WithBodyLayer<T>(string path, Color color) where T : BaseBodyAppearance
```

Parameters

path [string](#)

The asset path

color [Color](#)

The color instance

Returns

[NPCAppearance](#)

Type Parameters

T

WithFaceLayer<T>(string, uint)

Adds a Face Layer within the ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance WithFaceLayer<T>(string path, uint hexColor) where T : BaseFaceAppearance
```

Parameters

path [string](#)

The asset path

hexColor [uint](#)

The color in Hexadecimals

Returns

[NPCAppearance](#)

Type Parameters

T

WithFaceLayer<T>(string, Color)

Adds a Face Layer within the ScheduleOne.AvatarFramework.AvatarSettings

```
public NPCAppearance WithFaceLayer<T>(string path, Color color) where T  
: BaseFaceAppearance
```

Parameters

path [string](#)

The asset path

color [Color](#)

The color instance

Returns

[NPCAppearance](#)

Type Parameters

T

Class NPCCustomer

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing customer wrapper for an NPC. Provides helpers to configure and interact with customer behavior, including deal offers, contracts, and customer events. Customer configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

```
public sealed class NPCCustomer
```

Inheritance

[object](#) ← NPCCustomer

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Use this to enable NPCs to act as business customers that buy products from the player. Subscribe to events like [OnDealCompleted](#) and [OnUnlocked](#) for dynamic customer interactions.

Properties

IsCustomer

Returns whether this NPC currently has a Customer component.

```
public bool IsCustomer { get; }
```

Property Value

[bool](#)

Methods

EnsureCustomer()

Ensures this NPC has a Customer component, creating one if absent.

```
public void EnsureCustomer()
```

ForceDealOffer()

Forces the NPC to generate and offer a random contract based on their Customer data and relationship with the player.

The generated contract is based on:

- Available products listed for sale in ProductManagerApp
- NPC's drug type affinities and preferences
- NPC's spending budget (adjusted by relationship level)
- NPC's quality standards
- Current addiction level

For custom contracts with specific products/prices, use [OfferContract\(ContractInfo\)](#) instead.

```
public bool ForceDealOffer()
```

Returns

[bool](#)

True if a contract was generated and offered, false if generation failed.

OfferContract(ContractInfo)

Offers a custom contract to this customer using an API-friendly [ContractInfo](#). This method allows you to specify exactly what products, quantities, prices, and delivery details the contract should have. For automatic contract generation based on NPC preferences, use [ForceDealOffer\(\)](#) instead.

```
public bool OfferContract(ContractInfo info)
```

Parameters

info [ContractInfo](#)

The contract info containing the specific products, quantities, prices, and delivery details.

Returns

[bool](#)

True if the contract was successfully offered, false otherwise.

RecommendDealer(NPCDealer)

Recommends a dealer to the player. This marks the dealer as recommended and shows UI feedback.

```
public void RecommendDealer(NPCDealer dealer)
```

Parameters

dealer [NPCDealer](#)

The dealer NPC to recommend.

RequestProduct(Player)

Requests a product from the specified player (or local player if null).

```
public void RequestProduct(Player player = null)
```

Parameters

player [Player](#)

SetAwaitingDelivery(bool)

Sets whether the customer is awaiting delivery.

```
public void SetAwaitingDelivery(bool awaiting)
```

Parameters

awaiting [bool](#)

SetupDialog()

Sets up customer dialogue

```
public void SetupDialog()
```

Unlock()

Marks this customer as unlocked (visible to the player systems).

```
public void Unlock()
```

Events

OnContractAssigned

Subscribe to contract assigned event. Provides payment, product count, and delivery window via callback.

```
public event Action<float, int, int, int> OnContractAssigned
```

Event Type

[Action](#)<[float](#), [int](#), [int](#), [int](#)>

OnDealCompleted

Subscribe to deal completed event.

```
public event Action OnDealCompleted
```

Event Type

[Action](#)

OnUnlocked

Subscribe to customer unlocked event.

```
public event Action OnUnlocked
```

Event Type

[Action](#)

Class NPCDealer

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing dealer wrapper for an NPC. Provides helpers to configure and interact with dealer behavior, including customer assignment, cash management, and contract handling. Dealer configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

```
public sealed class NPCDealer
```

Inheritance

[object](#) ← NPCDealer

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Use this to enable NPCs to act as dealers that sell products to assigned customers. Subscribe to events like [OnRecruited](#) and [OnContractAccepted](#) for dynamic dealer interactions.

Properties

Home

Gets or sets the home building for this dealer.

```
public Building? Home { get; set; }
```

Property Value

IsDealer

Returns whether this NPC currently has dealer functionality.

```
public bool IsDealer { get; }
```

Property Value

[bool](#) ↗

Methods

AssignCustomer(NPC)

Assigns a customer to this dealer.

```
public void AssignCustomer(NPC customer)
```

Parameters

[customer](#) [NPC](#)

ChangeCash(float)

Changes the dealer's cash balance by the specified amount.

```
public void ChangeCash(float amount)
```

Parameters

amount [float](#)

CollectCash()

Collects all cash from the dealer and transfers it to the player.

```
public void CollectCash()
```

EnsureDealer()

Ensures this NPC has dealer functionality, initializing it if present.

```
public void EnsureDealer()
```

Remarks

Note: Since Dealer inherits from NPC in the base game (not a component), this will only work if the wrapped NPC is already a Dealer instance. For custom NPCs created via S1API, dealer functionality must be configured at prefab creation time using [EnsureDealer\(\)](#). This method is called automatically when the NPC spawns if [EnsureDealer\(\)](#) was used.

GetAssignedCustomers()

Gets the list of assigned customers.

```
public List<NPC> GetAssignedCustomers()
```

Returns

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

GetCash()

Gets the current cash balance held by this dealer.

```
public float GetCash()
```

Returns

[float](#)

HasBeenRecommended()

Gets whether the dealer has been recommended.

```
public bool HasBeenRecommended()
```

Returns

[bool](#)

IsRecruited()

Gets whether the dealer has been recruited.

```
public bool IsRecruited()
```

Returns

[bool](#)

MarkAsRecommended()

Marks this dealer as recommended (by another NPC).

```
public void MarkAsRecommended()
```

RecruitDealer()

Marks this dealer as recruited (hired by the player).

```
public void RecruitDealer()
```

RemoveCustomer(NPC)

Removes a customer assignment from this dealer.

```
public void RemoveCustomer(NPC customer)
```

Parameters

customer [NPC](#)

Events

OnContractAccepted

Subscribe to contract accepted event.

```
public event Action OnContractAccepted
```

Event Type

[Action](#) ↗

OnRecommended

Subscribe to dealer recommended event.

```
public event Action OnRecommended
```

Event Type

[Action ↗](#)

OnRecruited

Subscribe to dealer recruited event.

```
public event Action OnRecruited
```

Event Type

[Action ↗](#)

Class NPCDialogue

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing dialogue wrapper for an NPC. Provides helpers to create interactive conversations with branching dialogue trees, choice-based interactions, and dynamic responses. Use [BuildAndRegisterContainer\(string, Action<DialogueContainerBuilder>\)](#) to define custom conversations.

```
public sealed class NPCDialogue
```

Inheritance

[object](#) ← NPCDialogue

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Dialogue configuration is done in [OnCreated\(\)](#). Use [BuildAndSetDatabase\(Action<DialogueDatabaseBuilder>\)](#) for dialogue entries and [BuildAndRegisterContainer\(string, Action<DialogueContainerBuilder>\)](#) for conversation flows. Subscribe to choice and node events for dynamic dialogue behavior.

Properties

IsDialogueInProgress

Whether a dialogue is currently in progress for this NPC.

```
public bool IsDialogueInProgress { get; }
```

Property Value

[bool](#)

Methods

BuildAndRegisterContainer(string, Action<DialogueContainerBuilder>)

Builds a DialogueContainer with choice-based flow and registers it by name. Use this to define custom conversations for this NPC entirely from code.

If a custom NPC, you must also call [BuildAndSetDatabase](#) for this to work

```
public void BuildAndRegisterContainer(string containerName,  
Action<DialogueContainerBuilder> configure)
```

Parameters

containerName [string](#)

configure [Action](#)<DialogueContainerBuilder>

BuildAndSetDatabase(Action<DialogueDatabaseBuilder>)

Builds a dialogue database at runtime from string data and installs it on this NPC. Does not require asset bundles.

```
public void BuildAndSetDatabase(Action<DialogueDatabaseBuilder> configure)
```

Parameters

configure [Action](#)<DialogueDatabaseBuilder>

ClearCallbacks()

Removes all registered dialogue callbacks for this NPC.

```
public void ClearCallbacks()
```

End()

Ends any active dialogue.

```
public void End()
```

JumpTo(string, string, bool)

Immediately navigates this NPC's dialogue to a specific container and entry node. Returns true on success.

```
public bool JumpTo(string containerName, string entryNodeLabel, bool enableBehaviour  
= false)
```

Parameters

`containerName` [string](#)

`entryNodeLabel` [string](#)

`enableBehaviour` [bool](#)

Returns

[bool](#)

OnChoiceSelected(string, Action)

Register a callback to run when a choice with the given label is selected. Label must match the DialogueChoiceData.ChoiceLabel in your container.

```
public NPCDialogue OnChoiceSelected(string choiceLabel, Action callback)
```

Parameters

choiceLabel [string](#)

callback [Action](#)

Returns

[NPCDialogue](#)

OnConversationStart(Action)

Register a callback to run when a conversation starts with this NPC. This fires before the dialogue is displayed, allowing you to prepare or rebuild dialogue containers.

```
public NPCDialogue OnConversationStart(Action callback)
```

Parameters

callback [Action](#)

Returns

[NPCDialogue](#)

OnNodeDisplayed(string, Action)

Register a callback to run when a dialogue node with the given label is displayed.

```
public NPCDialogue OnNodeDisplayed(string nodeLabel, Action callback)
```

Parameters

nodeLabel [string](#)

callback [Action](#)

Returns

[NPCDialogue](#)

OverrideText(string)

Overrides the shown dialogue text (e.g., for temporary notifications). You generally won't want to use this

```
public void OverrideText(string text)
```

Parameters

text [string](#)

PlayReaction(string, float, bool)

Plays a reaction by key. If duration is -1 the underlying system decides duration.

```
public void PlayReaction(string key, float durationSeconds = -1, bool network  
= false)
```

Parameters

key [string](#)

durationSeconds [float](#)

`network bool`

ShowWorldText(string, float)

Shows worldspace dialogue text at the NPC for a duration.

```
public void ShowWorldText(string text, float durationSeconds)
```

Parameters

`text string`

`durationSeconds float`

Start(string, bool, string)

Starts a dialogue by container name present on the NPC's handler.

```
public void Start(string containerName, bool enableBehaviour = true, string entryNodeLabel = "ENTRY")
```

Parameters

`containerName string`

`enableBehaviour bool`

`entryNodeLabel string`

StopOverride()

Stops any active override and resumes normal dialogue display.

```
public void StopOverride()
```

UseContainerOnInteract(string)

When the player interacts with this NPC, force using the named container for the next dialogue. Returns true if the container was found and applied.

```
public bool UseContainerOnInteract(string containerName)
```

Parameters

containerName [string](#)

Returns

[bool](#)

UseContainerOnInteractOnce(string)

When the player interacts with this NPC, force using the named container once for the next dialogue. After the conversation begins, the override is automatically cleared so subsequent interactions use normal flow. Returns true if the container was found and applied.

```
public bool UseContainerOnInteractOnce(string containerName)
```

Parameters

containerName [string](#)

Returns

[bool](#)

Class NPCInventory

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing inventory wrapper for an S1API.Entities.NPCInventory.NPC. Provides helpers to query capacity and insert items safely.

```
public sealed class NPCInventory
```

Inheritance

[object](#) ← NPCInventory

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

CanItemFit(string, int)

Returns true if an item with the given ID and quantity can fit in this NPC's inventory.

```
public bool CanItemFit(string itemId, int quantity = 1)
```

Parameters

itemId [string](#)

quantity [int](#)

Returns

[bool](#)

EnsureInitialized()

Ensures the underlying ScheduleOne.NPCs.NPCInventory exists and has slots. Some custom NPCs added at runtime may not have had their slots populated yet. This method properly initializes slots without duplication, handling the fact that ScheduleOne.ItemFramework.ItemSlot.SetSlotOwner(ScheduleOne.ItemFramework.IItemSlotOwner) automatically adds slots to the owner's ItemSlots list.

```
public void EnsureInitialized()
```

GetCapacityForItem(string, int)

Returns how many units of the given item ID could fit right now.

```
public int GetCapacityForItem(string itemId, int quantity = 1)
```

Parameters

itemId [string](#)

quantity [int](#)

Returns

[int](#)

TryInsert(string, int, bool)

Attempts to insert an item created from the ID and quantity. Returns true if insertion was performed (fit was sufficient).

```
public bool TryInsert(string itemId, int quantity = 1, bool network = true)
```

Parameters

`itemId` [string](#)

`quantity` [int](#)

`network` [bool](#)

Returns

[bool](#)

Class NPCMovement

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing movement wrapper for an S1API.Entities.NPCMovement.NPC. Provides navigation, warping, facing, and reachability helpers. Keep in mind that as of API version 1.8.1, custom NPCs do not have an Avatar or an NPCMovement component.

```
public class NPCMovement
```

Inheritance

[object](#) ← NPCMovement

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

CurrentDestination

Current destination being navigated to, if any.

```
public Vector3 CurrentDestination { get; }
```

Property Value

Vector3

DefaultWalkSpeed

Gets the default walking speed of the NPC.

```
public float DefaultWalkSpeed { get; }
```

Property Value

[float](#)

FootPosition

Current foot position of the NPC in world space.

```
public Vector3 FootPosition { get; }
```

Property Value

[Vector3](#)

IsMoving

Whether the NPC is currently moving along a path.

```
public bool IsMoving { get; }
```

Property Value

[bool](#)

SpeedMultiplier

Gets or sets the overall speed multiplier that affects all movement.

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

Property Value

[float](#)

Methods

AddSpeedControl(string, int, float)

Adds or updates a speed control with the specified parameters. Higher priority values override lower ones.

```
public void AddSpeedControl(string id, int priority, float speed)
```

Parameters

[id](#) [string](#)

Unique identifier for this speed control.

[priority](#) [int](#)

Priority level (higher values take precedence).

[speed](#) [float](#)

Movement speed multiplier.

CanGetTo(Vector3)

Returns whether the NPC can path to a position.

```
public bool CanGetTo(Vector3 position)
```

Parameters

position Vector3

Target position.

Returns

[bool](#)

CanGetTo(Vector3, float)

Returns whether the NPC can path to a position within a threshold.

```
public bool CanGetTo(Vector3 position, float within)
```

Parameters

position Vector3

Target position.

within [float](#)

Acceptable distance to target.

Returns

[bool](#)

DoesSpeedControlExist(string)

Checks if a speed control with the given ID exists.

```
public bool DoesSpeedControlExist(string id)
```

Parameters

id [string](#)

The ID to check for.

Returns

[bool](#)

True if the speed control exists, false otherwise.

FaceDirection(Vector3)

Rotates to face a world-space direction.

```
public void FaceDirection(Vector3 forward)
```

Parameters

forward [Vector3](#)

Forward vector to face.

FacePoint(Vector3)

Rotates to face a world-space point.

```
public void FacePoint(Vector3 position)
```

Parameters

position [Vector3](#)

World position to face.

RemoveSpeedControl(string)

Removes a speed control by its ID.

```
public void RemoveSpeedControl(string id)
```

Parameters

id [string](#)

The ID of the speed control to remove.

SetDestination(Vector3)

Sets a new pathfinding destination in world space.

```
public void SetDestination(Vector3 position)
```

Parameters

position Vector3

Target destination.

Stop()

Stops any active movement/pathing immediately.

```
public void Stop()
```

Warp(Vector3)

Instantly moves the NPC to the given world position.

```
public void Warp(Vector3 position)
```

Parameters

position Vector3

Target position.

Class NPCPrefabBuilder

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Builder for composing NPC prefab configuration before network spawn. Use to declare networked components, spawn position, customer behavior, relationships, schedules, and appearance defaults.

```
public sealed class NPCPrefabBuilder
```

Inheritance

[object](#) ← NPCPrefabBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#) for proper save/load behavior. All builder methods return the builder instance for fluent chaining.

Methods

EnsureCustomer()

Adds customer behavior component to the NPC. Required before configuring customer defaults.

```
public NPCPrefabBuilder EnsureCustomer()
```

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Enables the NPC to act as a business customer that can buy products from the player.

EnsureDealer()

Adds dealer behavior to the NPC. Required before configuring dealer defaults.

```
public NPCPrefabBuilder EnsureDealer()
```

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Enables the NPC to act as a dealer that sells products to assigned customers. Note: Since Dealer inherits from NPC in the base game, dealer functionality is applied through configuration rather than component addition. This marks the NPC type as dealer-capable. When the NPC spawns, [EnsureDealer\(\)](#) will be called automatically to initialize dealer functionality and ensure the messaging app displays the correct Dealer category badge.

WithAppearanceDefaults(Action<AvatarDefaultsBuilder>)

Declares appearance defaults via a wrapper builder. Values are embedded as an AvatarSettings asset reference on the prefab and applied to the runtime avatar on spawn (server and clients).

```
public NPCPrefabBuilder  
WithAppearanceDefaults(Action<NPCPrefabBuilder.AvatarDefaultsBuilder> configure)
```

Parameters

configure `Action<NPCPrefabBuilder.AvatarDefaultsBuilder>`

Returns

[NPCPrefabBuilder](#)

WithCustomerDefaults(Action<CustomerDataBuilder>)

Configures customer behavior defaults using the [CustomerDataBuilder](#). Requires [EnsureCustomer\(\)](#) to be called first.

```
public NPCPrefabBuilder WithCustomerDefaults(Action<CustomerDataBuilder> configure)
```

Parameters

configure `Action<CustomerDataBuilder>`

Action to configure customer defaults using the builder.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Configure spending behavior, order frequency, customer standards, product preferences, and relationship requirements. This configuration is essential for proper save/load behavior and must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

WithDealerDefaults(Action<DealerDataBuilder>)

Configures dealer behavior defaults using the [DealerDataBuilder](#). Requires [EnsureDealer\(\)](#) to be called first.

```
public NPCPrefabBuilder WithDealerDefaults(Action<DealerDataBuilder> configure)
```

Parameters

configure [Action<DealerDataBuilder>](#)

Action to configure dealer defaults using the builder.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Configure dealer settings such as signing fee, commission cut, dealer type, quality restrictions, and deal tracking. This configuration is essential for proper save/load behavior and must be done in [Configure Prefab\(NPCPrefabBuilder\)](#).

WithIcon(Sprite?)

Declares the icon sprite to be embedded on the prefab. This sprite is used for UI elements such as messages, contacts, and relationships. Should be 64x64 or 128x128 pixels. Uses default if not set.

```
public NPCPrefabBuilder WithIcon(Sprite? icon)
```

Parameters

icon [Sprite](#)

Optional sprite for UI elements. Uses default if null.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

WithIdentity(string, string, string)

Declares identity defaults (ID, first and last name) to be embedded on the prefab. These values are applied on spawn on both server and clients.

```
public NPCPrefabBuilder WithIdentity(string id, string firstName, string lastName)
```

Parameters

`id` [string](#)

`firstName` [string](#)

`lastName` [string](#)

Returns

[NPCPrefabBuilder](#)

WithInventoryDefaults(Action<RandomInventoryItemsBuilder>)

Declares default inventory configuration for this NPC type. Supports startup items (always present) and random cash (varies on each sleep).

```
public NPCPrefabBuilder WithInventoryDefaults(Action<RandomInventoryItemsBuilder> configure)
```

Parameters

`configure` [Action](#)<[RandomInventoryItemsBuilder](#)>

Action to configure inventory defaults using the builder.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

All configurations are optional. Applied when the NPC is spawned.

WithRelationshipDefaults(Action<NPCRelationshipDataBuilder>)

Configures default relationship settings (delta, unlock type, connections) for this NPC type. Applied to the instance after spawn and before save-data hydration.

```
public NPCPrefabBuilder WithRelationshipDefaults(Action<NPCRelationshipDataBuilder> configure)
```

Parameters

[configure Action<NPCRelationshipDataBuilder>](#)

Action to configure relationship defaults using the builder.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Configure starting relationship level, unlock state, and connections to other NPCs. This configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#) for proper save/load behavior.

WithSchedule(params IScheduleActionSpec[])

Declares a schedule using a params array of specs for convenience.

```
public NPCPrefabBuilder WithSchedule(params IScheduleActionSpec[] specs)
```

Parameters

specs [IScheduleActionSpec](#)

Array of schedule action specifications.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

WithSchedule(Action<PrefabScheduleBuilder>)

Defines the NPC's schedule using the [PrefabScheduleBuilder](#). Schedule actions are planned and pre-created on the prefab.

```
public NPCPrefabBuilder WithSchedule(Action<PrefabScheduleBuilder> configure)
```

Parameters

configure [Action](#)<[PrefabScheduleBuilder](#)>

Action to configure schedule using the builder.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Use to configure movement patterns, building visits, and timed activities. The plan is applied at runtime to activate precreated actions. Schedule configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#) for proper save/load behavior.

WithSchedule(IEnumerable<IScheduleActionSpec>)

Declares a schedule using a prebuilt set of specs. Use when composing plans externally or sharing between NPC types.

```
public NPCPrefabBuilder WithSchedule( IEnumerable<IScheduleActionSpec> specs )
```

Parameters

specs [IEnumerable<IScheduleActionSpec>](#)

Enumerable collection of schedule action specifications.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

WithSpawnPosition(Vector3)

Sets the spawn position with default rotation. Applied every time the NPC is spawned.

```
public NPCPrefabBuilder WithSpawnPosition( Vector3 position )
```

Parameters

position Vector3

World position where the NPC will spawn.

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

WithSpawnPosition(Vector3, Quaternion)

Sets the spawn position and rotation for this NPC type. Applied every time the NPC is spawned (new games and loaded games).

```
public NPCPrefabBuilder WithSpawnPosition(Vector3 position, Quaternion rotation)
```

Parameters

position Vector3

World position where the NPC will spawn.

rotation Quaternion

Rotation for the NPC (defaults to Quaternion.identity).

Returns

[NPCPrefabBuilder](#)

The builder instance for fluent chaining.

Remarks

Use world coordinates. Consider building entrances, roads, and safe spawn areas. Position should be on a walkable surface.

Class NPCPrefabBuilder.AvatarDefaultsBuilder

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Wrapper for authoring appearance defaults without exposing game types to modders.

```
public sealed class NPCPrefabBuilder.AvatarDefaultsBuilder
```

Inheritance

[object](#) ← NPCPrefabBuilder.AvatarDefaultsBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

AvatarDefaultsBuilder()

```
public AvatarDefaultsBuilder()
```

Properties

EyeBallTint

```
public Color EyeBallTint { get; set; }
```

Property Value

Color

EyeballMaterialIdentifier

```
public string EyeballMaterialIdentifier { get; set; }
```

Property Value

[string](#)

EyebrowRestingAngle

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

Property Value

[float](#)

EyebrowRestingHeight

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

Property Value

[float](#)

EyebrowScale

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

PropertyValue

[float](#)

EyebrowThickness

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

PropertyValue

[float](#)

Gender

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

PropertyValue

[float](#)

HairColor

```
public Color HairColor { get; set; }
```

PropertyValue

Color

HairPath

```
public string HairPath { get; set; }
```

Property Value

[string](#)

Height

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

Property Value

[float](#)

LeftEye

```
public (float topLidOpen, float bottomLidOpen) LeftEye { get; set; }
```

Property Value

([float](#) [topLidOpen](#), [float](#) [bottomLidOpen](#))

LeftEyeLidColor

```
public Color LeftEyeLidColor { get; set; }
```

Property Value

Color

PupilDilation

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

Property Value

[float](#)

RightEye

```
public (float topLidOpen, float bottomLidOpen) RightEye { get; set; }
```

Property Value

([float](#) [topLidOpen](#), [float](#) [bottomLidOpen](#))

RightEyeLidColor

```
public Color RightEyeLidColor { get; set; }
```

Property Value

Color

SkinColor

```
public Color32 SkinColor { get; set; }
```

Property Value

Weight

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

Property Value

[float](#)

Methods

WithAccessoryLayer(string, Color)

```
public NPCPrefabBuilder.AvatarDefaultsBuilder WithAccessoryLayer(string path,  
Color color)
```

Parameters

[path](#) [string](#)

[color](#) Color

Returns

[NPCPrefabBuilder](#).[AvatarDefaultsBuilder](#)

WithBodyLayer(string, Color)

```
public NPCPrefabBuilder.AvatarDefaultsBuilder WithBodyLayer(string path,  
Color color)
```

Parameters

path [string](#)

color [Color](#)

Returns

[NPCPrefabBuilder](#).[AvatarDefaultsBuilder](#)

WithFaceLayer(string, Color)

```
public NPCPrefabBuilder.AvatarDefaultsBuilder WithFaceLayer(string path,  
Color color)
```

Parameters

path [string](#)

color [Color](#)

Returns

[NPCPrefabBuilder](#).[AvatarDefaultsBuilder](#)

Class NPCRelationship

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing wrapper for an NPC's relationship data. Provides safe access to relationship values, unlock state, connections, and convenience helpers which bridge to the base game's NPCRelationData. Relationship configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

```
public sealed class NPCRelationship
```

Inheritance

[object](#) ← NPCRelationship

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Use this to manage NPC social connections, unlock states, and relationship levels with the player and other NPCs. Subscribe to [OnChanged](#) and [OnUnlocked](#) events for dynamic relationship interactions.

Properties

ConnectionIDs

Returns the IDs of current connection NPCs (safe across Mono and IL2CPP).

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

Property Value

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

Delta

Current relationship delta, clamped to [0, 5]. Setting invokes the game's SetRelationship.

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

Property Value

[float](#)

IsKnown

True if the NPC is known to the player (directly or via mutual connections).

```
public bool IsKnown { get; }
```

Property Value

[bool](#)

IsMutuallyKnown

True if the NPC is known via mutual connections even if not directly unlocked.

```
public bool IsMutuallyKnown { get; }
```

Property Value

[bool](#)

IsUnlocked

True if the NPC has been unlocked (known) to the player systems.

```
public bool IsUnlocked { get; }
```

Property Value

[bool](#)

Normalized

Normalized relationship delta in [0, 1].

```
public float Normalized { get; }
```

Property Value

[float](#)

Type

The way this NPC was unlocked. Setting attempts to assign the base game's UnlockType via reflection without unlocking.

```
public NPCRelationship.UnlockType Type { get; set; }
```

Property Value

[NPCRelationship.UnlockType](#)

Methods

Add(float, bool)

Adds to the relationship delta (optionally networked).

```
public void Add(float delta, bool network = true)
```

Parameters

delta [float](#)

network [bool](#)

BuildAndSetRelationshipData(Action<NPCRelationshipDataBuilder>)

Deprecated: Declare defaults in [ConfigurePrefab\(NPCPrefabBuilder\)](#) via [WithRelationshipDefaults\(Action<NPCRelationshipDataBuilder>\)](#). Runtime mutation is no longer supported to preserve save/load consistency.

```
[Obsolete("Declare defaults in NPC.ConfigurePrefab via  
NPCPrefabBuilder.WithRelationshipDefaults. Runtime mutation is disabled.")]  
public void BuildAndSetRelationshipData(Action<NPCRelationshipDataBuilder>  
configure)
```

Parameters

configure [Action](#)<NPCRelationshipDataBuilder>

SetUnlockType(UnlockType)

Sets the underlying unlock type without changing locked state. Use [Unlock\(UnlockType, bool\)](#) if you intend to unlock as well.

```
public void SetUnlockType(NPCRelationship.UnlockType type)
```

Parameters

type [NPCRelationship.UnlockType](#)

Unlock(UnlockType, bool)

Unlocks this NPC with the specified type.

```
public void Unlock(NPCRelationship.UnlockType type = UnlockType.DirectApproach, bool  
notify = true)
```

Parameters

type [NPCRelationship.UnlockType](#)

notify [bool](#)

UnlockConnections()

Unlocks all connected NPCs (recommendation unlocking).

```
public void UnlockConnections()
```

Events

OnChanged

Subscribes to relationship change events. Callback receives the change amount. Best-effort under IL2CPP; silently no-ops if delegate bridging is unavailable.

```
public event Action<float> OnChanged
```

Event Type

[Action](#)<[float](#)>

OnUnlocked

Subscribes to unlocked events. Callback receives unlock type and notify flag. Best-effort under IL2CPP; silently no-ops if delegate bridging is unavailable.

```
public event Action<NPCRelationship.UnlockType, bool> OnUnlocked
```

Event Type

[Action](#)<[NPCRelationship.UnlockType](#), [bool](#)>

Enum NPCRelationship.UnlockType

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Unlock types mirrored from the base game.

```
public enum NPCRelationship.UnlockType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Fields

DirectApproach = 1

Recommendation = 0

Class NPCSchedule

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Modder-facing scheduling wrapper for an NPC. Exposes the underlying schedule manager to enable, disable, and manage scheduled actions and curfew modes. Schedule configuration must be done in [ConfigurePrefab\(NPCPrefabBuilder\)](#).

```
public sealed class NPCSchedule
```

Inheritance

[object](#) ← NPCSchedule

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Use this to control NPC movement patterns, building visits, and timed activities. Schedules are defined in [ConfigurePrefab\(NPCPrefabBuilder\)](#) using [WithSchedule\(Action<PrefabScheduleBuilder>\)](#) and managed at runtime via this wrapper.

Properties

CurfewModeEnabled

Whether the schedule is currently in curfew mode.

```
public bool CurfewModeEnabled { get; }
```

Property Value

[bool](#)

IsEnabled

Whether the schedule is currently enabled.

```
public bool IsEnabled { get; }
```

Property Value

[bool](#)

Methods

ClearActions(bool, bool)

Removes all actions under the schedule manager with optional filtering by action type.

```
public void ClearActions(bool includeSignals = true, bool includeEvents = true)
```

Parameters

[includeSignals](#) [bool](#)

Whether to remove signal-type actions (e.g., WalkTo, DriveToCarPark). Default is [true](#).

[includeEvents](#) [bool](#)

Whether to remove event-type actions (e.g., StayInBuilding, LocationDialogue). Default is [true](#).

Remarks

This method removes all schedule actions from the NPC's schedule manager. Actions are disabled and reset instead of being destroyed to maintain FishNet network component indices and avoid network synchronization issues.

After clearing actions, the schedule manager is re-initialized to update the action order and timing.

Use with caution as this will completely reset the NPC's scheduled behavior.

Disable()

Disables the NPC's schedule.

```
public void Disable()
```

Enable()

Enables the NPC's schedule.

```
public void Enable()
```

EnforceState()

Forces the manager to enforce state immediately (e.g., after toggles or time jumps).

```
public void EnforceState()
```

EnsureDealSignal()

Ensures that a deal-wait signal exists under the schedule manager for customer handover functionality.

```
public void EnsureDealSignal()
```

Remarks

This method ensures that a ScheduleOne.NPCs.Schedules.NPCSignal_WaitForDelivery component exists on the NPC's schedule manager. This signal is required for customer NPCs to properly handle deal interactions and handovers with the player.

If the signal already exists, it will be properly initialized and wired to the customer component. If it doesn't exist, a warning will be logged indicating that it should be added via [ConfigurePrefab\(NPCPrefabBuilder\)](#).

The deal signal allows the NPC to wait for deliveries and toggle customer handover states.

GetActionNames()

Returns the names of all currently configured actions, including inactive and disabled ones.

```
public IReadOnlyList<string> GetActionNames()
```

Returns

[IReadOnlyList<string>](#)

A read-only list of action names. Returns an empty list if no schedule manager exists.

Remarks

This method retrieves the names of all schedule actions currently configured on the NPC, regardless of their active state. This can be useful for debugging or monitoring the NPC's schedule configuration.

The returned list includes both signal-type and event-type actions.

GetActiveActionName()

Returns the active action label, if any.

```
public string GetActiveActionName()
```

Returns

[string](#)

SetCurfewMode(bool)

Sets or clears curfew mode.

```
public void SetCurfewMode(bool enabled)
```

Parameters

enabled [bool](#)

Class Player

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Represents a player within the game.

```
public class Player : IEntity, IHealth
```

Inheritance

[object](#) ← Player

Implements

[IEntity](#), [IHealth](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Fields

All

All players currently in the game.

```
public static readonly List<Player> All
```

Field Value

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

Properties

CrimeData

The player's crime data, including wanted level, crimes committed, and pursuit state.

```
public PlayerCrimeData CrimeData { get; }
```

Property Value

[PlayerCrimeData](#)

Crouched

Is player crouching

```
public bool Crouched { get; }
```

Property Value

[bool](#)

CurrentAvatarSettings

The player's current avatar settings (appearance configuration).

```
public object CurrentAvatarSettings { get; }
```

Property Value

[object](#)

CurrentHealth

The current health of the player.

```
public float CurrentHealth { get; }
```

Property Value

[float](#)

CurrentProperty

Property this player is currently in

```
public PropertyWrapper? CurrentProperty { get; }
```

Property Value

[PropertyWrapper](#)

CurrentRegion

Region this player is currently in

```
public Region CurrentRegion { get; }
```

Property Value

[Region](#)

IsArrested

If player is currently under arrest

```
public bool IsArrested { get; }
```

Property Value

[bool](#) ↗

IsDead

Whether the player is dead or not.

```
public bool IsDead { get; }
```

Property Value

[bool](#) ↗

IsInVehicle

If player is currently in a vehicle

```
public bool IsInVehicle { get; }
```

Property Value

[bool](#) ↗

IsInvincible

Whether the player is invincible or not.

```
public bool IsInvincible { get; set; }
```

PropertyValue

[bool](#)

IsLocal

Whether this player is the client player or a networked player.

```
public bool IsLocal { get; }
```

PropertyValue

[bool](#)

IsRagdolled

Is Player currently ragdolled

```
public bool IsRagdolled { get; }
```

PropertyValue

[bool](#)

IsReadyToSleep

If player is ready for sleep

```
public bool IsReadyToSleep { get; }
```

PropertyValue

[bool](#)

IsSkating

If player is currently using a skateboard

```
public bool IsSkating { get; }
```

Property Value

[bool](#)

IsSleeping

If player is currently sleeping

```
public bool IsSleeping { get; }
```

Property Value

[bool](#)

IsTased

Is player currently under tased effect

```
public bool IsTased { get; }
```

Property Value

[bool](#)

IsUnconscious

If player is unconscious

```
public bool IsUnconscious { get; }
```

Property Value

[bool](#)

LastDrivenVehicle

The Last vehicle this player has driven

```
public LandVehicle? LastDrivenVehicle { get; }
```

Property Value

[LandVehicle](#)

LastVisitedProperty

Last Property this player visted (May only be owned properties)

```
public PropertyWrapper? LastVisitedProperty { get; }
```

Property Value

[PropertyWrapper](#)

Local

The current client player (player executing your code).

```
public static Player Local { get; }
```

Property Value

[Player](#)

MaxHealth

The maximum health of the player. Note: In the base game this is a constant (100). Changing it at runtime is unsupported. Setting this will clamp the player's current health to the specified value.

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

Property Value

[float](#) ↗

Name

The name of the player. For single player, this appears to always return [Player](#).

```
public string Name { get; }
```

Property Value

[string](#) ↗

Position

The world position of the player.

```
public Vector3 Position { get; set; }
```

Property Value

Vector3

Scale

The scale of the player.

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

Property Value

[float](#)

TimeSinceVehicleExit

Time since player has exited a vehicle

```
public float TimeSinceVehicleExit { get; }
```

Property Value

[float](#)

Transform

The transform of the player. Please do not set the properties of the Transform.

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

Property Value

Transform

Methods

Damage(int)

Deals damage to the player.

```
public void Damage(int amount)
```

Parameters

amount [int](#)

The amount of damage to deal.

Heal(int)

Heals the player.

```
public void Heal(int amount)
```

Parameters

amount [int](#)

The amount of healing to apply to the player.

Kill()

Kills the player.

```
public void Kill()
```

Revive()

Revives the player.

```
public void Revive()
```

Events

LocalPlayerSpawned

Fired when the local (client-owned) player finishes client startup.

```
public static event Action<Player>? LocalPlayerSpawned
```

Event Type

[Action](#)<Player>

OnDeath

Called when the player dies.

```
public event Action OnDeath
```

Event Type

[Action](#)

PlayerDespawned

Fired when a player is destroyed/removed.

```
public static event Action<Player>? PlayerDespawned
```

Event Type

[Action<Player>](#)

PlayerSpawned

Fired when any player finishes client startup.

```
public static event Action<Player>? PlayerSpawned
```

Event Type

[Action<Player>](#)

Class RandomInventoryItemsBuilder

Namespace: [S1API.Entities](#)

Assembly: S1API.dll

Builder for configuring startup items and random cash for NPCs. Public surface uses strings/primitives only. All configurations are optional.

```
public sealed class RandomInventoryItemsBuilder
```

Inheritance

[object](#) ← RandomInventoryItemsBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

WithClearInventoryEachNight(bool)

Configures whether the inventory should be cleared each night. Default: true if only random items are configured, false if startup items are configured (to preserve them).

```
public RandomInventoryItemsBuilder WithClearInventoryEachNight(bool clearEachNight)
```

Parameters

clearEachNight [bool](#)

If true, inventory is cleared on sleep. If false, items persist across sleep cycles.

Returns

[RandomInventoryItemsBuilder](#)

WithRandomCash(int, int)

Enables and configures random cash generation.

```
public RandomInventoryItemsBuilder WithRandomCash(int min, int max)
```

Parameters

min [int](#)

Minimum cash amount.

max [int](#)

Maximum cash amount.

Returns

[RandomInventoryItemsBuilder](#)

WithStartupItem(string)

Adds a single startup item that will be present in the NPC's inventory when spawned.

```
public RandomInventoryItemsBuilder WithStartupItem(string itemId)
```

Parameters

itemId [string](#)

The ID of the item to add.

Returns

[RandomInventoryItemsBuilder](#)

WithStartupItems(params string[])

Adds multiple startup items that will be present in the NPC's inventory when spawned.

```
public RandomInventoryItemsBuilder WithStartupItems(params string[] itemIds)
```

Parameters

itemIds [string](#) ↗

Array of item IDs to add.

Returns

[RandomInventoryItemsBuilder](#)

Namespace S1API.GameTime

Classes

[TimeManager](#)

Provides access to various time management functions in the game.

Structs

[GameDateTime](#)

Represents an in-game datetime (elapsed days and 24-hour time).

Enums

[Day](#)

Represents all days available in the week.

Enum Day

Namespace: [S1API.GameTime](#)

Assembly: S1API.dll

Represents all days available in the week.

```
public enum Day
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`Friday = 4`

Represents the fifth day of the week.

`Monday = 0`

Represents the first day of the week.

`Saturday = 5`

Represents the sixth day of the week.

`Sunday = 6`

Represents the seventh day of the week.

`Thursday = 3`

Represents the fourth day of the week.

`Tuesday = 1`

Represents the second day of the week.

`Wednesday = 2`

Represents the third day of the week.

Struct GameDateTime

Namespace: [S1API.GameTime](#)

Assembly: S1API.dll

Represents an in-game datetime (elapsed days and 24-hour time).

```
public struct GameDateTime
```

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

GameDateTime(GameDateTime)

Constructs a GameDateTime from the internal GameDateTime struct.

```
public GameDateTime(GameDateTime gameDateTime)
```

Parameters

gameDateTime GameDateTime

GameDateTime(GameDateTimeData)

Constructs a GameDateTime from an internal GameDateTimeData.

```
public GameDateTime(GameDateTimeData data)
```

Parameters

data GameDateTimeData

GameDateTime(int)

Constructs a GameDateTime from total minutes.

```
public GameDateTime(int minSum)
```

Parameters

minSum int ↗

GameDateTime(int, int)

Constructs a GameDateTime from elapsed days and 24-hour time.

```
public GameDateTime(int elapsedDays, int time)
```

Parameters

elapsedDays int ↗

time int ↗

Fields

ElapsedDays

```
public int ElapsedDays
```

Field Value

[int ↗](#)

Time

```
public int Time
```

Field Value

[int ↗](#)

Methods

AddMinutes(int)

Returns a new GameDateTime with additional minutes added.

```
public GameDateTime AddMinutes(int minutes)
```

Parameters

[minutes int ↗](#)

Returns

[GameDateTime](#)

GetFormattedTime()

Returns the current time formatted as a 12-hour AM/PM string. Example: "12:30 PM"

```
public string GetFormattedTime()
```

Returns

[string](#)

GetMinSum()

Returns the total minute sum (days * 1440 + minutes of day).

```
public int GetMinSum()
```

Returns

[int](#)

IsNightTime()

Returns true if the time is considered nighttime. (Before 6AM or after 6PM)

```
public bool IsNightTime()
```

Returns

[bool](#)

IsSameDay(GameDateTime)

Returns true if the two GameDateTimes are on the same day (ignores time).

```
public bool IsSameDay(GameDateTime other)
```

Parameters

other [GameDateTime](#)

Returns

[bool](#)

IsSameTime(GameDateTime)

Returns true if the two GameDateTimes are at the same day and time.

```
public bool IsSameTime(GameDateTime other)
```

Parameters

other [GameDateTime](#)

Returns

[bool](#)

ToS1()

Converts this wrapper to the internal GameDateTime struct.

```
public GameDateTime ToS1()
```

Returns

[GameDateTime](#)

ToString()

String representation: "Day 3, 2:30 PM"

```
public override string ToString()
```

Returns

[string](#)

Operators

operator +(GameDateTime, GameDateTime)

```
public static GameDateTime operator +(GameDateTime a, GameDateTime b)
```

Parameters

a [GameDateTime](#)

b [GameDateTime](#)

Returns

[GameDateTime](#)

operator >(GameDateTime, GameDateTime)

```
public static bool operator >(GameDateTime a, GameDateTime b)
```

Parameters

a [GameDateTime](#)

b [GameDateTime](#)

Returns

[bool](#) ↗

operator <(GameDateTime, GameDateTime)

```
public static bool operator <(GameDateTime a, GameDateTime b)
```

Parameters

a [GameDateTime](#)

b [GameDateTime](#)

Returns

[bool](#) ↗

operator -(GameDateTime, GameDateTime)

```
public static GameDateTime operator -(GameDateTime a, GameDateTime b)
```

Parameters

a [GameDateTime](#)

b [GameDateTime](#)

Returns

[GameDateTime](#)

Class TimeManager

Namespace: [S1API.GameTime](#)

Assembly: S1API.dll

Provides access to various time management functions in the game.

```
public static class TimeManager
```

Inheritance

[object](#) ← TimeManager

Inherited Members

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

Fields

OnDayPass

Called when a new in-game day starts.

```
public static Action OnDayPass
```

Field Value

[Action](#)

OnSleepEnd

Called when the player finishes sleeping. Parameter: total minutes skipped during sleep.

```
public static Action<int> OnSleepEnd
```

Field Value

[Action](#)<int>

OnSleepStart

Called when the player starts sleeping.

```
public static Action OnSleepStart
```

Field Value

[Action](#)

OnTick

Called at every tick of gametime.

```
public static Action OnTick
```

Field Value

[Action](#)

OnWeekPass

Called when a new in-game week starts.

```
public static Action OnWeekPass
```

Field Value

Properties

CurrentDay

The current in-game day (Monday, Tuesday, etc.).

```
public static Day CurrentDay { get; }
```

Property Value

[Day](#)

CurrentTime

The current 24-hour time (e.g., 1330 for 1:30 PM).

```
public static int CurrentTime { get; }
```

Property Value

[int ↗](#)

ElapsedDays

The number of in-game days elapsed.

```
public static int ElapsedDays { get; }
```

Property Value

[int](#)

IsEndOfDay

Whether the game is currently at the end of the day (4:00 AM).

```
public static bool IsEndOfDay { get; }
```

Property Value

[bool](#)

IsNight

Whether it is currently nighttime in-game.

```
public static bool IsNight { get; }
```

Property Value

[bool](#)

NormalizedTime

The current normalized time of day (0.0 = start, 1.0 = end).

```
public static float NormalizedTime { get; }
```

Property Value

[float](#)

Playtime

Total playtime (in seconds).

```
public static float Playtime { get; }
```

Property Value

[float](#)

SleepInProgress

Whether the player is currently sleeping.

```
public static bool SleepInProgress { get; }
```

Property Value

[bool](#)

TimeOverridden

Whether the time is currently overridden (frozen or custom).

```
public static bool TimeOverridden { get; }
```

Property Value

[bool](#)

Methods

FastForwardToWakeTime()

Fast-forwards time to morning wake time (7:00 AM).

```
public static void FastForwardToWakeTime()
```

Get24HourTimeFromMinutes(int)

Converts total minutes into 24-hour time format.

```
public static int Get24HourTimeFromMinutes(int minutes)
```

Parameters

minutes [int](#)

Returns

[int](#)

GetFormatted12HourTime()

Gets the current time formatted in 12-hour AM/PM format.

```
public static string GetFormatted12HourTime()
```

Returns

[string](#)

GetMinutesFrom24HourTime(int)

Converts 24-hour time to total minutes.

```
public static int GetMinutesFrom24HourTime(int time24h)
```

Parameters

time24h [int](#)

Returns

[int](#)

IsCurrentTimeWithinRange(int, int)

Returns true if the current time is within the specified 24-hour range.

```
public static bool IsCurrentTimeWithinRange(int startTime24h, int endTime24h)
```

Parameters

startTime24h [int](#)

endTime24h [int](#)

Returns

[bool](#)

SetElapsedDays(int)

Sets the number of elapsed in-game days.

```
public static void SetElapsedDays(int days)
```

Parameters

days [int ↗](#)

SetTime(int, bool)

Sets the current time manually.

```
public static void SetTime(int time24h, bool local = false)
```

Parameters

time24h [int ↗](#)

local [bool ↗](#)

Namespace S1API.Graffiti

Classes

[GraffitiEvents](#)

Provides events for graffiti-related gameplay actions.

[GraffitiManager](#)

Provides access to graffiti-related game systems and spray surfaces.

[SpraySurface](#)

Represents a surface that can be spray painted with graffiti.

Class GraffitiEvents

Namespace: [S1API.Graffiti](#)

Assembly: S1API.dll

Provides events for graffiti-related gameplay actions.

```
public static class GraffitiEvents
```

Inheritance

[object](#) ← GraffitiEvents

Inherited Members

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

Events

GraffitiCompleted

Fired when a player completes a graffiti piece and receives XP/rewards. This event is triggered when the player closes the graffiti UI after painting.

```
public static event Action<SpraySurface>? GraffitiCompleted
```

Event Type

[Action](#)<[SpraySurface](#)>

Class GraffitiManager

Namespace: [S1API.Graffiti](#)

Assembly: S1API.dll

Provides access to graffiti-related game systems and spray surfaces.

```
public static class GraffitiManager
```

Inheritance

[object](#) ← GraffitiManager

Inherited Members

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

Properties

UntaggedSpraySurfaces

Gets all spray surfaces that have not been drawn on yet (no strokes).

```
public static List<SpraySurface> UntaggedSpraySurfaces { get; }
```

Property Value

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

Methods

FindNearestUntaggedSurface(Vector3?)

Finds the nearest untagged spray surface to a given position.

```
public static SpraySurface? FindNearestUntaggedSurface(Vector3? position)
```

Parameters

position Vector3

The position to search from.

Returns

[SpraySurface](#)

The nearest untagged spray surface, or null if none found.

GetAllSpraySurfaces()

Gets all spray surfaces in the game.

```
public static List<SpraySurface> GetAllSpraySurfaces()
```

Returns

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

A list of all spray surfaces, wrapped in S1API SpraySurface objects.

GetUntaggedSpraySurfaces()

Gets all spray surfaces that have not been drawn on yet (no strokes).

```
public static List<SpraySurface> GetUntaggedSpraySurfaces()
```

Returns

[List](#) <SpraySurface>

A list of untagged spray surfaces.

Class SpraySurface

Namespace: [S1API.Graffiti](#)

Assembly: S1API.dll

Represents a surface that can be spray painted with graffiti.

```
public sealed class SpraySurface
```

Inheritance

[object](#) ← SpraySurface

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

DrawingOutputTexture

The output texture for the drawing on this surface.

```
public Texture DrawingOutputTexture { get; }
```

Property Value

Texture

GUID

The globally unique identifier for this spray surface.

```
public Guid GUID { get; }
```

Property Value

[Guid](#) ↗

HasDrawingBeenFinalized

Whether the drawing has been finalized (player closed the graffiti UI).

```
public bool HasDrawingBeenFinalized { get; }
```

Property Value

[bool](#) ↗

PaintedPixelCount

The number of painted pixels on this surface.

```
public int PaintedPixelCount { get; }
```

Property Value

[int](#) ↗

Position

The world position of the bottom-left reference point of this spray surface.

```
public Vector3 Position { get; }
```

Property Value

Vector3

StrokeCount

The number of strokes drawn on this surface.

```
public int StrokeCount { get; }
```

Property Value

[int↗](#)

Events

OnDrawingChanged

Event fired when the drawing on this surface changes.

```
public event Action OnDrawingChanged
```

Event Type

[Action↗](#)

Namespace S1API.Growing

Classes

[MushroomBedInstance](#)

Represents an instance of a mushroom bed in the world.

[PlantInstance](#)

Represents an instance of a growing plant in the world.

[SeedCreator](#)

The seed Creator for custom seeds to be added.

[SeedDefinition](#)

Represents the definition of a Seed item (what you buy in shops).

[SeedInstance](#)

Represents an instance of a functional seed in the world. (Not just the definition – this is the physical object you interact with.)

[ShroomColonyInstance](#)

Represents an instance of a mushroom colony growing in the world.

Class MushroomBedInstance

Namespace: [S1API.Growing](#)

Assembly: S1API.dll

Represents an instance of a mushroom bed in the world.

```
public sealed class MushroomBedInstance
```

Inheritance

[object](#) ← MushroomBedInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Mushroom beds are specialized growing containers that can hold mushroom colonies. They require cool temperatures (maintained by AC units) for optimal mushroom growth.

Properties

CurrentColony

The mushroom colony currently growing in this bed, or null if empty.

```
public ShroomColonyInstance CurrentColony { get; }
```

Property Value

[ShroomColonyInstance](#)

GameObject

The GameObject of the mushroom bed.

```
public GameObject GameObject { get; }
```

Property Value

GameObject

Transform

The transform of the mushroom bed.

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

Property Value

Transform

Methods

ContainsGrowable()

Whether this mushroom bed currently contains a growing mushroom colony.

```
public bool ContainsGrowable()
```

Returns

[bool](#) ↗

GetAverageTileTemperature()

Gets the average temperature of the tiles under this mushroom bed.

```
public float GetAverageTileTemperature()
```

Returns

[float](#)

The average temperature in Fahrenheit.

GetGrowSurfaceSideLength()

Gets the side length of the grow surface in world units.

```
public float GetGrowSurfaceSideLength()
```

Returns

[float](#)

The side length of the internal grow surface.

GetGrowthProgressNormalized()

Gets the current growth progress of the mushroom colony (0.0 to 1.0). Returns 0 if no colony is present.

```
public float GetGrowthProgressNormalized()
```

Returns

[float](#)

IsReadyForHarvest(out string)

Checks if the mushroom bed is ready for harvest.

```
public bool IsReadyForHarvest(out string reason)
```

Parameters

reason [string](#)

Output parameter that contains the reason if not ready for harvest.

Returns

[bool](#)

True if ready for harvest, false otherwise.

Class PlantInstance

Namespace: [S1API.Growing](#)

Assembly: S1API.dll

Represents an instance of a growing plant in the world.

```
public class PlantInstance
```

Inheritance

[object](#) ← PlantInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

IsFullyGrown

Whether the plant is fully grown.

```
public bool IsFullyGrown { get; }
```

Property Value

[bool](#)

NormalizedGrowth

The current growth stage as a float from 0.0 to 1.0.

```
public float NormalizedGrowth { get; }
```

Property Value

[float](#) ↗

Quality

The quality level of this plant.

```
public float Quality { get; }
```

Property Value

[float](#) ↗

SeedDefinition

The SeedDefinition that this plant originated from.

```
public SeedDefinition SeedDefinition { get; }
```

Property Value

[SeedDefinition](#)

YieldMultiplier

The yield multiplier of this plant.

```
public float YieldMultiplier { get; }
```

Property Value

[float](#)

Methods

Destroy(bool)

Destroys this plant in-game.

```
public void Destroy(bool dropScraps = false)
```

Parameters

[dropScraps](#) [bool](#)

Whether to drop plant scraps (trash) at the plant's location.

Class SeedCreator

Namespace: [S1API.Growing](#)

Assembly: S1API.dll

The seed Creator for custom seeds to be added.

```
public static class SeedCreator
```

Inheritance

[object](#) ← SeedCreator

Inherited Members

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

Methods

CreateSeed(string, string, string, int, GameObject?, GameObject?, Sprite?)

```
public static SeedDefinition CreateSeed(string id, string name, string description,  
int stackLimit = 10, GameObject? functionSeedPrefab = null, GameObject? plantPrefab  
= null, Sprite? icon = null)
```

Parameters

`id` [string](#)

`name` [string](#)

`description` [string](#)

`stackLimit` [int](#)

`functionSeedPrefab` [GameObject](#)

`plantPrefab` [GameObject](#)

 Sprite

Returns

[SeedDefinition](#)

Class SeedDefinition

Namespace: [S1API.Growing](#)

Assembly: S1API.dll

Represents the definition of a Seed item (what you buy in shops).

```
public class SeedDefinition : ItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← SeedDefinition

Inherited Members

[ItemDefinition.CreateInstance\(int\)](#) , [ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) ,
[ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) ,
[ItemDefinition.Category](#) , [ItemDefinition.Icon](#) , [ItemDefinition.AvailableInDemo](#) ,
[ItemDefinition.LegalStatus](#) , [ItemDefinition.LabelDisplayColor](#) , [ItemDefinition.Keywords](#) ,
[ItemDefinition.GUID](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

FunctionalSeedPrefab

The prefab that is spawned when planting this seed.

```
public GameObject? FunctionalSeedPrefab { get; }
```

Property Value

GameObject

PlantPrefab

The plant prefab this seed grows into.

```
public GameObject? PlantPrefab { get; }
```

Property Value

GameObject

Methods

CreateSeedInstance()

Creates an instance of this seed in the world (FunctionalSeed prefab).

```
public GameObject CreateSeedInstance()
```

Returns

GameObject

Class SeedInstance

Namespace: [S1API.Growing](#)

Assembly: S1API.dll

Represents an instance of a functional seed in the world. (Not just the definition – this is the physical object you interact with.)

```
public class SeedInstance
```

Inheritance

[object](#) ← SeedInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

HasExitedVial

Whether the seed currently has exited its vial.

```
public bool HasExitedVial { get; }
```

Property Value

[bool](#)

Methods

ForceExitVial()

Force the seed to exit the vial manually.

```
public void ForceExitVial()
```

Class ShroomColonyInstance

Namespace: [S1API.Growing](#)

Assembly: S1API.dll

Represents an instance of a mushroom colony growing in the world.

```
public sealed class ShroomColonyInstance
```

Inheritance

[object](#) ← ShroomColonyInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

Mushroom colonies grow in mushroom beds and require cool temperatures ($\leq 15^{\circ}\text{F}$) to grow properly. Each colony can produce multiple individual mushrooms that can be harvested.

Fields

MaxTemperatureForGrowth

The maximum temperature (in Fahrenheit) at which mushrooms can grow.

```
public const float MaxTemperatureForGrowth = 15
```

Field Value

[float](#)

Properties

BaseShroomYield

The base yield of mushrooms this colony will produce when fully grown.

```
public int BaseShroomYield { get; }
```

Property Value

[int ↗](#)

GameObject

The GameObject of the colony.

```
public GameObject GameObject { get; }
```

Property Value

GameObject

GrownMushroomCount

The number of individual mushrooms currently grown in this colony.

```
public int GrownMushroomCount { get; }
```

Property Value

[int ↗](#)

GrowthProgress

The current growth progress as a float from 0.0 to 1.0.

```
public float GrowthProgress { get; }
```

Property Value

[float](#)

IsFullyGrown

Whether the colony is fully grown and ready for harvest.

```
public bool IsFullyGrown { get; }
```

Property Value

[bool](#)

IsTooHotToGrow

Whether the colony's temperature is too hot for growth. Mushrooms require temperatures at or below 15°F to grow.

```
public bool IsTooHotToGrow { get; }
```

Property Value

[bool](#)

NormalizedQuality

The normalized quality level of this colony (0.0 to 1.0).

```
public float NormalizedQuality { get; }
```

Property Value

[float](#) ↗

Methods

Destroy()

Destroys this mushroom colony in-game.

```
public void Destroy()
```

Remarks

This will remove the colony and all its mushrooms from the world. Use with caution as this operation cannot be undone.

Namespace S1API.Input

Classes

[Controls](#)

Modder-facing facade over the base game's input state to keep S1API consumers decoupled from the underlying `ScheduleOne.GameInput` type across Mono/IL2CPP.

Class Controls

Namespace: [S1API.Input](#)

Assembly: S1API.dll

Modder-facing facade over the base game's input state to keep S1API consumers decoupled from the underlying [ScheduleOne.GameInput](#) type across Mono/IL2CPP.

```
public static class Controls
```

Inheritance

[object](#) ← Controls

Inherited Members

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

Properties

IsTyping

Gets or sets whether the player is currently typing in a UI field. When true, gameplay input should generally be ignored by systems listening for controls.

```
public static bool IsTyping { get; set; }
```

Property Value

[bool](#)

Namespace S1API.Internal

Namespaces

[S1API.Internal.Abstraction](#)

[S1API.Internal.Entities](#)

[S1API.Internal.Utils](#)

Namespace S1API.Internal.Abstraction

Classes

[EventHelper](#)

This static class provides us an easy wrapper for subscribing and unsubscribing unity actions.

[Registerable](#)

INTERNAL: A registerable base class for use internally. Not intended for modder use.

[Saveable](#)

Generic wrapper for saveable classes.

Class EventHelper

Namespace: [S1API.Internal.Abstraction](#)

Assembly: S1API.dll

This static class provides us an easy wrapper for subscribing and unsubscribing unity actions.

```
public static class EventHelper
```

Inheritance

[object](#) ← EventHelper

Inherited Members

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

Methods

AddEventTrigger(EventTrigger, EventTriggerType, Action)

Adds an EventTrigger entry in a cross-compatible manner. Use this from Mono mods so IL2CPP handles the actual Entry construction.

```
public static void AddEventTrigger(EventTrigger trigger, EventTriggerType eventType,  
Action listener)
```

Parameters

trigger EventTrigger

Target EventTrigger component.

eventType EventTriggerType

The EventTriggerType to subscribe to.

listener [Action](#)

Callback invoked when the event fires.

AddEventTrigger(EventTrigger, EventTriggerType, Action<BaseEventData>)

Adds an EventTrigger entry with access to BaseEventData.

```
public static void AddEventTrigger(EventTrigger trigger, EventTriggerType eventType,  
Action<BaseEventData> listener)
```

Parameters

trigger EventTrigger

Target EventTrigger component.

eventType EventTriggerType

The EventTriggerType to subscribe to.

listener [Action](#)<BaseEventData>

Callback invoked with BaseEventData when the event fires.

AddListener(Action, UnityEvent)

Adds a listener to the event, as well as the subscription list.

```
public static void AddListener(Action listener, UnityEvent unityEvent)
```

Parameters

listener [Action](#)

The action / method you want to subscribe.

unityEvent UnityEvent

The event you want to subscribe to.

AddListener<T>(Action<T>, UnityEvent<T>)

```
public static void AddListener<T>(Action<T> listener, UnityEvent<T> unityEvent)
```

Parameters

listener [Action](#)<T>

unityEvent [UnityEvent](#)<T>

Type Parameters

T

RemoveListener(Action, UnityEvent)

Removes a listener to the event, as well as the subscription list.

```
public static void RemoveListener(Action listener, UnityEvent unityEvent)
```

Parameters

listener [Action](#)<T>

The action / method you want to unsubscribe.

unityEvent [UnityEvent](#)

The event you want to unsubscribe from.

RemoveListener<T>(Action<T>, UnityEvent<T>)

```
public static void RemoveListener<T>(Action<T> listener, UnityEvent<T> unityEvent)
```

Parameters

listener [Action](#)<T>

unityEvent UnityEvent<T>

Type Parameters

T

Class Registerable

Namespace: [S1API.Internal.Abstraction](#)

Assembly: S1API.dll

INTERNAL: A registerable base class for use internally. Not intended for modder use.

```
public abstract class Registerable
```

Inheritance

[object](#) ← Registerable

Derived

[Saveable](#), [PhoneApp](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Registerable()

```
protected Registerable()
```

Methods

OnCreated()

Override hook for creation/registration completion.

```
protected virtual void OnCreated()
```

OnDestroyed()

Override hook for destruction/unregistration completion.

```
protected virtual void OnDestroyed()
```

Class Saveable

Namespace: [S1API.Internal.Abstraction](#)

Assembly: S1API.dll

Generic wrapper for saveable classes.

```
public abstract class Saveable : Registerable
```

Inheritance

[object](#) ← [Registerable](#) ← Saveable

Derived

[NPC](#), [Quest](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Saveable()

```
protected Saveable()
```

Properties

LoadOrder

Determines when this saveable should load relative to base game saveables.

```
public virtual SaveableLoadOrder LoadOrder { get; }
```

Property Value

[SaveableLoadOrder](#)

Default is [AfterBaseGame](#), which loads after base game entities are loaded. Override this property to return [BeforeBaseGame](#) if your mod data needs to be available before the base game's ISaveables are loaded.

Examples

```
public class EarlyConfigSaveable : Saveable
{
    public override SaveableLoadOrder LoadOrder => SaveableLoadOrder.BeforeBaseGame;

    [SaveableField("config")]
    private ModConfig _config = new ModConfig();

    protected override void OnLoaded()
    {
        // Base game entities are NOT loaded yet
        ApplyGlobalSettings(_config);
    }
}
```

Remarks

AfterBaseGame (default): Your [OnLoaded\(\)](#) method is called after base game entities (NPCs, buildings, vehicles) have been loaded. This is the recommended setting for most mods.

BeforeBaseGame: Your [OnLoaded\(\)](#) method is called before base game entities are loaded. Use this only if you need to set up hooks or state that the base game loading process depends on.

Note: All saveables are saved at the same time (after base game save), regardless of load order.

Methods

OnLoaded()

Called after all saveable fields have been loaded from their respective JSON files. This method can be overridden in derived classes to perform additional initialization or processing after the save data has been restored.

```
protected virtual void OnLoaded()
```

OnSaved()

Called after all saveable fields have been saved to their respective JSON files. This method can be overridden in derived classes to perform additional finalization or processing after the save data has been written to disk.

```
protected virtual void OnSaved()
```

RequestGameSave(bool)

Requests the game to perform a save operation. If a game is not currently loaded, the request is ignored and the method returns false.

```
public static bool RequestGameSave(bool immediate = false)
```

Parameters

immediate [bool](#)

When true, saves immediately; otherwise schedules a short delayed save.

Returns

[bool](#)

True if a save was requested; false if the game is not in a savable state.

Namespace S1API.Internal.Entities

Classes

[NPCPrefabIdentity](#)

INTERNAL: Stores identity and appearance defaults on the prefab so clients receive the same configuration on network spawn without relying on RPCs/SyncVars. On IL2CPP, stores data in a static registry keyed by prefab name to work around field serialization issues with RegisterTypeInIL2CPP components.

Class NPCPrefabIdentity

Namespace: [S1API.Internal.Entities](#)

Assembly: S1API.dll

INTERNAL: Stores identity and appearance defaults on the prefab so clients receive the same configuration on network spawn without relying on RPCs/SyncVars. On IL2CPP, stores data in a static registry keyed by prefab name to work around field serialization issues with RegisterTypeInIL2CPP components.

```
public sealed class NPCPrefabIdentity : MonoBehaviour
```

Inheritance

[object](#) ← NPCPrefabIdentity

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NPCPrefabIdentity()

```
public NPCPrefabIdentity()
```

Fields

AppearanceDefaults

```
public AvatarSettings AppearanceDefaults
```

Field Value

AvatarSettings

ConnectionIDs

```
public List<string> ConnectionIDs
```

Field Value

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

DealerHomeBuildingName

```
public string DealerHomeBuildingName
```

Field Value

[string](#)

FirstName

```
public string FirstName
```

Field Value

[string](#)

Icon

```
public Sprite Icon
```

Field Value

Sprite

Id

```
public string Id
```

Field Value

[string](#) ↗

LastNames

```
public string LastName
```

Field Value

[string](#) ↗

PrefabName

```
public string PrefabName
```

Field Value

[string](#) ↗

RelationDelta

```
public float? RelationDelta
```

Field Value

[float](#)?

UnlockType

```
public NPCRelationship.UnlockType? UnlockType
```

Field Value

[NPCRelationship.UnlockType](#)?

Unlocked

```
public bool? Unlocked
```

Field Value

[bool](#)?

Methods

ApplyRelationshipDataTo(NPC, bool)

Apply stored relationship defaults to a base-game NPC's relation data. Safe to call on both server and clients.

```
public void ApplyRelationshipDataTo(NPC npc, bool preserveUnlockState = false)
```

Parameters

npc NPC

preserveUnlockState [bool ↗](#)

ApplyTo(NPC)

Apply stored defaults to a base-game NPC instance. Safe to call on both server and clients.

```
public void ApplyTo(NPC npc)
```

Parameters

npc NPC

Namespace S1API.Internal.Utils

Classes

[ArrayExtensions](#)

INTERNAL: Extensions for Arrays. This class is intended for internal API use only. Mod developers should use S1API.Utils.ArrayExtensions instead.

[ButtonUtils](#)

INTERNAL: Utility helpers for managing Unity UI UnityEngine.UI.Buttons. This class is intended for internal API use only. Mod developers should use S1API.Utils.ButtonUtils instead.

[ColorUtils](#)

INTERNAL: Utilities for the UnityEngine.Color class. This class is intended for internal API use only. Mod developers should use S1API.Utils.ColorUtils instead.

[ImageUtils](#)

INTERNAL: A utility class to assist with loading images into the game. Useful for icons such as on phone apps, custom NPCs, quests, etc. This class is intended for internal API use only. Mod developers should use S1API.Utils.ImageUtils instead.

[RandomUtils](#)

INTERNAL: A utility class providing random selection functionality for lists and numeric ranges. This class is intended for internal API use only. Mod developers should use S1API.Utils.RandomUtils instead.

[ToggleUtils](#)

INTERNAL: Utilities for subscribing to and managing Toggle value change events in a cross-compatible way between Mono and IL2CPP. Handles Unity versions where Toggle.onValueChanged is exposed as either a field or a property. This class is intended for internal API use only. Mod developers should use S1API.Utils.ToggleUtils instead.

Class ArrayExtensions

Namespace: [S1API.Internal.Utils](#)

Assembly: S1API.dll

INTERNAL: Extensions for Arrays. This class is intended for internal API use only. Mod developers should use S1API.Utils.ArrayExtensions instead.

```
[Obsolete("This class is for internal API use only. Mod developers should use  
S1API.Utils.ArrayExtensions instead. This class will be made internal in a  
future version.")]  
public static class ArrayExtensions
```

Inheritance

[object](#) ← ArrayExtensions

Inherited Members

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

Methods

AddItemToArray<T>(T[], T)

Add's an item to an existing array

```
public static T[] AddItemToArray<T>(this T[]? array, T item)
```

Parameters

array T[]

item T

Returns

T[]

Type Parameters

T

Class ButtonUtils

Namespace: [S1API.Internal.Utils](#)

Assembly: S1API.dll

INTERNAL: Utility helpers for managing Unity UI UnityEngine.UI.Buttons. This class is intended for internal API use only. Mod developers should use S1API.Utils.ButtonUtils instead.

```
[Obsolete("This class is for internal API use only. Mod developers should  
use S1API.Utils.ButtonUtils instead. This class will be made internal in a  
future version.")]  
public static class ButtonUtils
```

Inheritance

[object](#) ← ButtonUtils

Inherited Members

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

Methods

AddListener(Button, Action)

Adds a click listener to the specified button, ensuring compatibility with IL2CPP and Mono.

```
public static void AddListener(Button button, Action action)
```

Parameters

button Button

action [Action](#)

ClearListeners(Button)

Removes all listeners from the specified button safely.

```
public static void ClearListeners(Button button)
```

Parameters

button Button

Disable(Button, Text?, string?)

Disables the button and optionally updates the label.

```
public static void Disable(Button button, Text? label = null, string? text = null)
```

Parameters

button Button

label Text

text [string](#)

Enable(Button, Text?, string?)

Enables the button and optionally updates the label.

```
public static void Enable(Button button, Text? label = null, string? text = null)
```

Parameters

button Button

label Text

text [string](#)

RemoveListener(Button, Action)

Removes a previously added click listener from the specified button.

```
public static void RemoveListener(Button button, Action action)
```

Parameters

button Button

action [Action](#)

SetLabel(Text, string)

Sets the label text of a button with a known Text child.

```
public static void SetLabel(Text label, string text)
```

Parameters

label Text

text [string](#)

SetStyle(Button, Text, string, Color)

Sets the button label and background color.

```
public static void SetStyle(Button button, Text label, string text, Color bg)
```

Parameters

button Button

label Text

text [string](#)

bg Color

Class ColorUtils

Namespace: [S1API.Internal.Utils](#)

Assembly: S1API.dll

INTERNAL: Utilities for the UnityEngine.Color class. This class is intended for internal API use only. Mod developers should use S1API.Utils.ColorUtils instead.

```
[Obsolete("This class is for internal API use only. Mod developers should  
use S1API.Utils.ColorUtils instead. This class will be made internal in a  
future version.")]  
public static class ColorUtils
```

Inheritance

[object](#) ← ColorUtils

Inherited Members

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

Methods

ToColor(uint)

Convert's a [int](#) value to UnityEngine.Color

```
public static Color ToColor(this uint hexColor)
```

Parameters

hexColor [uint](#)

Returns

Color

Class ImageUtils

Namespace: [S1API.Internal.Utils](#)

Assembly: S1API.dll

INTERNAL: A utility class to assist with loading images into the game. Useful for icons such as on phone apps, custom NPCs, quests, etc. This class is intended for internal API use only. Mod developers should use S1API.Utils.ImageUtils instead.

```
[Obsolete("This class is for internal API use only. Mod developers should  
use S1API.Utils.ImageUtils instead. This class will be made internal in a  
future version.")]  
public static class ImageUtils
```

Inheritance

[object](#) ← ImageUtils

Inherited Members

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

Methods

LoadImage(string)

Loads an image from the specified file path and converts it into a Sprite object.

```
public static Sprite? LoadImage(string fileName)
```

Parameters

fileName [string](#)

The name of the file (with path) containing the image to load.

Returns

Sprite

A Sprite object representing the loaded image, or null if the image could not be loaded or the file does not exist.

LoadImageFromResource(Assembly, string, float, FilterMode)

Loads an image from an embedded resource stream and converts it into a Sprite object.

```
public static Sprite? LoadImageFromResource(Assembly assembly, string resourceName,  
float pixelsPerUnit = 100, FilterMode filterMode = 1)
```

Parameters

assembly [Assembly](#)

The assembly containing the embedded resource.

resourceName [string](#)

The fully qualified name of the embedded resource (e.g., "Namespace.Assets.Icon.png").

pixelsPerUnit [float](#)

The pixels per unit for the sprite. Defaults to 100f.

filterMode [FilterMode](#)

The filter mode for the texture. Defaults to FilterMode.Bilinear.

Returns

Sprite

A Sprite object representing the loaded image, or null if the resource could not be found or loaded.

LoadImageRaw(byte[])

Loads an image from a byte array and converts it into a Sprite object.

```
public static Sprite? LoadImageRaw(byte[] data)
```

Parameters

data [byte](#)[]

The byte array containing the image data to load.

Returns

Sprite

A Sprite object representing the loaded image, or null if the image could not be loaded.

TextureToSprite(Texture2D?, float)

Converts a Texture2D to a Sprite.

```
public static Sprite? TextureToSprite(Texture2D? texture, float pixelsPerUnit = 100)
```

Parameters

texture Texture2D

The texture to convert.

pixelsPerUnit [float](#)

The pixels per unit for the sprite. Defaults to 100f.

Returns

Sprite

A Sprite object, or null if texture is null.

Class RandomUtils

Namespace: [S1API.Internal.Utils](#)

Assembly: S1API.dll

INTERNAL: A utility class providing random selection functionality for lists and numeric ranges. This class is intended for internal API use only. Mod developers should use S1API.Utils.RandomUtils instead.

```
[Obsolete("This class is for internal API use only. Mod developers should  
use S1API.Utils.RandomUtils instead. This class will be made internal in a  
future version.")]  
public static class RandomUtils
```

Inheritance

[object](#) ← RandomUtils

Inherited Members

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

Methods

PickMany<T>(IList<T>, int)

Returns a specified number of unique random elements from a list. If the count exceeds the number of available elements, returns all elements in random order.

```
public static List<T> PickMany<T>(this IList<T> list, int count)
```

Parameters

list [IList](#)<T>

The list of items to pick from.

count [int](#)

The number of random items to pick.

Returns

[List](#)<T>

A list containing the selected random items, or an empty list if the input list is null or empty.

Type Parameters

T

PickOne<T>(IList<T>)

Returns a random element from the provided list, or the default value of type T if the list is null or empty.

```
public static T PickOne<T>(this IList<T> list)
```

Parameters

list [IList](#)<T>

The list from which to select a random element.

Returns

T

A randomly selected element from the list, or the default value of type T if the list is null or empty.

Type Parameters

T

PickUnique<T>(IList<T>, Func<T, bool>, int)

Returns a random element from a list that satisfies the given condition, with a maximum number of attempts. If no such element can be found within the allowed attempts, returns the default value of the type.

```
public static T PickUnique<T>(this IList<T> list, Func<T, bool> isDuplicate, int maxTries = 10)
```

Parameters

list [IList](#)<T>

The list of items to pick from.

isDuplicate [Func](#)<T, bool>

A function to determine if the selected item satisfies the duplicate condition.

maxTries [int](#)

The maximum number of attempts to find a valid item.

Returns

T

A randomly selected item that satisfies the condition, or the default value of the type if no valid item is found.

Type Parameters

T

RangelInt(int, int)

Generates a random integer within the specified range.

```
public static int RangeInt(int minInclusive, int maxExclusive)
```

Parameters

`minInclusive` [int](#)

The inclusive lower bound of the random number.

`maxExclusive` [int](#)

The exclusive upper bound of the random number.

Returns

[int](#)

A random integer greater than or equal to `minInclusive` and less than `maxExclusive`.

Class ToggleUtils

Namespace: [S1API.Internal.Utils](#)

Assembly: S1API.dll

INTERNAL: Utilities for subscribing to and managing Toggle value change events in a cross-compatible way between Mono and IL2CPP. Handles Unity versions where Toggle.onValueChanged is exposed as either a field or a property. This class is intended for internal API use only. Mod developers should use S1API.Utils.ToggleUtils instead.

```
[Obsolete("This class is for internal API use only. Mod developers should  
use S1API.Utils.ToggleUtils instead. This class will be made internal in a  
future version.")]  
public static class ToggleUtils
```

Inheritance

[object](#) ← ToggleUtils

Inherited Members

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

Methods

AddListener(Toggle, Action<bool>)

Adds a listener to a Toggle's onValueChanged event in an IL2CPP-safe manner.

```
public static void AddListener(Toggle toggle, Action<bool> listener)
```

Parameters

toggle Toggle

listener [Action<bool>](#)

ClearListeners(Toggle)

Removes all listeners from a Toggle's onValueChanged event.

```
public static void ClearListeners(Toggle toggle)
```

Parameters

toggle Toggle

GetGraphic(Toggle)

Gets the Toggle's checkmark graphic in a version-agnostic manner.

```
public static Graphic? GetGraphic(Toggle toggle)
```

Parameters

toggle Toggle

Returns

Graphic

RemoveListener(Toggle, Action<bool>)

Removes a previously added listener from a Toggle's onValueChanged event.

```
public static void RemoveListener(Toggle toggle, Action<bool> listener)
```

Parameters

toggle Toggle

`listener` `Action<bool>`

SetGraphic(Toggle, Graphic)

Sets the Toggle's checkmark graphic in a version-agnostic manner (field or property).

```
public static void SetGraphic(Toggle toggle, Graphic graphic)
```

Parameters

`toggle` Toggle

`graphic` Graphic

Namespace S1API.Items

Classes

[AvatarEquippablePaths](#)

Provides constants for base game AvatarEquippable prefab paths. These paths can be used with [WithAvatarEquippable\(string, AvatarHand, string\)](#).

[AvatarEquippableRegistry](#)

Manages registration and loading of AvatarEquippable prefabs from AssetBundles. Allows modders to load AvatarEquippable prefabs and register them so they can be used with equippable items.

[BuildableItemCreator](#)

Provides convenient static methods for creating custom buildable items.

[BuildableItemDefinition](#)

Represents a buildable item definition that can be placed in the game world. Extends [StorableItemDefinition](#) with building-specific properties.

[BuildableItemDefinitionBuilder](#)

Builder for composing buildable item definitions at runtime. Use fluent methods to configure buildable item properties before calling [Build\(\)](#).

[ClothingItemCreator](#)

Provides convenient static methods for creating custom clothing items. Use [CreateBuilder\(\)](#) for flexible configuration or [CloneFrom\(string\)](#) for variants.

[ClothingItemDefinition](#)

Represents a clothing item definition that can be worn by the player. Extends [StorableItemDefinition](#) with clothing-specific properties.

[ClothingItemDefinitionBuilder](#)

Builder for composing clothing item definitions at runtime. Use fluent methods to configure clothing properties before calling [Build\(\)](#).

[ClothingItemInstance](#)

Represents a clothing item instance in the game world (physical clothing you own). Extends [ItemInstance](#) with color information.

[Equippable](#)

Represents an equippable component that can be attached to items. Provides a wrapper around the game's native equippable system.

[EquippableBuilder](#)

Builder for creating equippable components that can be attached to items. Use this to create custom equippable behavior for items.

[ItemCreator](#)

Provides convenient static methods for creating custom items. Use [CreateBuilder\(\)](#) for flexible configuration or [CreateItem\(string, string, string, ItemCategory, int, float, float, LegalStatus, Sprite, Equippable\)](#) for quick creation.

[ItemDefinition](#)

Represents an item definition in-game. This is the base wrapper class for all item types.

[ItemInstance](#)

Represents an item instance in the game world (physical item you own).

[ItemManager](#)

Provides access to managing items across the game.

[ItemSlotInstance](#)

Represents an item slot within the game. These are present within storage, the hot bar, etc.

[StorableItemDefinition](#)

Represents an item definition that can be purchased, sold, and stored in inventories. Extends [Item Definition](#) with economic properties.

[StorableItemDefinitionBuilder](#)

Builder for composing item definitions at runtime. Use fluent methods to configure item properties before calling [Build\(\)](#).

Enums

[AvatarHand](#)

Represents which hand holds an equippable item in third-person view.

[BuildSoundType](#)

Specifies the sound type played when a buildable item is placed.

[ClothingApplicationType](#)

Represents how a clothing item is applied to the avatar. Mirrors ScheduleOne.Clothing.ELClothingApplicationType.

[ClothingColor](#)

Represents available clothing colors. Mirrors ScheduleOne.Clothing.ELClothingColor.

ClothingSlot

Represents the slot where a clothing item can be equipped. Mirrors ScheduleOne.Clothing.ELClothingSlot.

ItemCategory

A list of item categories available in-game.

LegalStatus

Represents the legal status of an item (e.g., legal or illegal).

Class AvatarEquippablePaths

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Provides constants for base game AvatarEquippable prefab paths. These paths can be used with [WithAvatarEquippable\(string, AvatarHand, string\)](#).

```
public static class AvatarEquippablePaths
```

Inheritance

[object](#) ← AvatarEquippablePaths

Inherited Members

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

Remarks

All paths are relative to the Resources folder. Example: "avatar/equippables/Baton" refers to Resources/avatar/equippables/Baton.prefab

Fields

BasePath

Base path for all avatar equippable prefabs.

```
public const string BasePath = "avatar/equippables"
```

Field Value

[string](#)

Baton

Baton AvatarEquippable prefab path.

```
public const string Baton = "avatar/equipables/Baton"
```

Field Value

[string](#)

Beer

Beer AvatarEquippable prefab path.

```
public const string Beer = "avatar/equipables/Beer"
```

Field Value

[string](#)

BrokenBottle

Broken Bottle AvatarEquippable prefab path.

```
public const string BrokenBottle = "avatar/equipables/BrokenBottle"
```

Field Value

[string](#)

Coffee

Coffee AvatarEquippable prefab path.

```
public const string Coffee = "avatar/equippables/Coffee"
```

Field Value

[string](#) ↗

Cuke

Cuke (Energy Drink) AvatarEquippable prefab path.

```
public const string Cuke = "avatar/equippables/Cuke"
```

Field Value

[string](#) ↗

Hammer

Hammer AvatarEquippable prefab path.

```
public const string Hammer = "avatar/equippables/Hammer"
```

Field Value

[string](#) ↗

Joint

Joint (Marijuana) AvatarEquippable prefab path.

```
public const string Joint = "avatar/equipables/Joint"
```

Field Value

[string](#) ↗

Knife

Knife AvatarEquippable prefab path.

```
public const string Knife = "avatar/equipables/Knife"
```

Field Value

[string](#) ↗

M1911

M1911 Pistol AvatarEquippable prefab path.

```
public const string M1911 = "avatar/equipables/M1911"
```

Field Value

[string](#) ↗

PhoneLowered

Phone (Lowered) AvatarEquippable prefab path.

```
public const string PhoneLowered = "avatar/equipables/Phone_Lowered"
```

Field Value

[string](#)

PhoneRaised

Phone (Raised) AvatarEquippable prefab path.

```
public const string PhoneRaised = "avatar/equippables/Phone_Raised"
```

Field Value

[string](#)

Pipe

Pipe AvatarEquippable prefab path.

```
public const string Pipe = "avatar/equippables/Pipe"
```

Field Value

[string](#)

Revolver

Revolver AvatarEquippable prefab path.

```
public const string Revolver = "avatar/equippables/Revolver"
```

Field Value

[string](#)

Taser

Taser AvatarEquippable prefab path.

```
public const string Taser = "avatar/equippables/Taser"
```

Field Value

[string](#)

TrashBag

Trash Bag AvatarEquippable prefab path.

```
public const string TrashBag = "avatar/equippables/TrashBag"
```

Field Value

[string](#)

Class AvatarEquippableRegistry

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Manages registration and loading of AvatarEquippable prefabs from AssetBundles. Allows modders to load AvatarEquippable prefabs and register them so they can be used with equippable items.

```
public static class AvatarEquippableRegistry
```

Inheritance

[object](#) ← AvatarEquippableRegistry

Inherited Members

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

Remarks

This class now uses RuntimeResourceRegistry internally for asset registration.

Methods

GetRegisteredPrefab(string)

Gets a registered AvatarEquippable prefab.

```
public static GameObject GetRegisteredPrefab(string assetPath)
```

Parameters

assetPath [string](#)

The Resources path.

Returns

GameObject

The registered prefab, or null if not found.

IsRegistered(string)

Checks if an AvatarEquippable is registered at the given path.

```
public static bool IsRegistered(string assetPath)
```

Parameters

assetPath [string](#)

The Resources path to check.

Returns

bool

True if registered, false otherwise.

LoadAndRegisterFromBundle(WrappedAssetBundle, string, string)

Loads an AvatarEquippable prefab from an AssetBundle and registers it.

```
public static bool LoadAndRegisterFromBundle(WrappedAssetBundle bundle, string  
prefabName, string assetPath)
```

Parameters

bundle [WrappedAssetBundle](#)

The AssetBundle containing the prefab.

prefabName [string](#)

The name of the prefab asset in the bundle.

assetPath [string](#)

The Resources path to register (e.g., "Equippables/MyItem").

Returns

[bool](#)

True if loading and registration were successful.

LoadAndRegisterFromEmbeddedBundle(string, string, string, Assembly)

Loads an AvatarEquippable prefab from an embedded AssetBundle and registers it.

```
public static bool LoadAndRegisterFromEmbeddedBundle(string bundleName, string
prefabName, string assetPath, Assembly assemblyOverride = null)
```

Parameters

bundleName [string](#)

The name of the embedded AssetBundle resource.

prefabName [string](#)

The name of the prefab asset in the bundle.

assetPath [string](#)

The Resources path to register (e.g., "Equippables/MyItem").

assemblyOverride [Assembly](#)

Optional assembly to load the bundle from. If null, uses executing assembly.

Returns

[bool](#)

True if loading and registration were successful.

RegisterAvatarEquippable(string, GameObject)

Registers an AvatarEquippable prefab with a Resources path. After registration, the prefab can be loaded via Resources.Load using the provided assetPath.

```
public static bool RegisterAvatarEquippable(string assetPath, GameObject prefab)
```

Parameters

assetPath [string](#)

The Resources path to register (e.g., "Equippables/MyItem").

prefab [GameObject](#)

The AvatarEquippable prefab GameObject to register.

Returns

[bool](#)

True if registration was successful.

Remarks

The prefab must have an AvatarEquippable component attached. Ensure the assetPath matches what you'll use in WithAvatarEquippable().

Enum AvatarHand

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents which hand holds an equippable item in third-person view.

```
public enum AvatarHand
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`Left = 0`

Left hand holds the item.

`Right = 1`

Right hand holds the item (default).

Enum BuildSoundType

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Specifies the sound type played when a buildable item is placed.

```
public enum BuildSoundType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`Cardboard = 3`

Cardboard building sound.

`Metal = 1`

Metal building sound.

`Plastic = 2`

Plastic building sound.

`Wood = 0`

Wood building sound.

Class BuildableItemCreator

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Provides convenient static methods for creating custom buildable items.

```
public static class BuildableItemCreator
```

Inheritance

[object](#) ← BuildableItemCreator

Inherited Members

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

Remarks

Use [CreateBuilder\(\)](#) for creating items from scratch, or [CloneFrom\(string\)](#) for creating variants of existing buildable items.

Methods

CloneFrom(BuildableItemDefinition)

Creates a new buildable item by cloning from an existing BuildableItemDefinition wrapper.

```
public static BuildableItemDefinitionBuilder CloneFrom\(BuildableItemDefinition source\)
```

Parameters

source [BuildableItemDefinition](#)

The buildable item definition to clone from.

Returns

[BuildableItemDefinitionBuilder](#)

A builder initialized with the source item's properties, ready for customization.

Examples

```
var originalRack = ItemManager.GetItemDefinition("StorageRack-1x0.5")
as BuildableItemDefinition;
var metalRack = BuildableItemCreator.CloneFrom(originalRack)
    .WithBasicInfo("metal_rack_small", "Small Metal Storage Rack", "A
metal version")
    .WithBuildSound(BuildSoundType.Metal)
    .Build();
```

CloneFrom(string)

Creates a new buildable item by cloning an existing item's properties. This is useful for creating variants of existing items (e.g., different materials, sizes).

```
public static BuildableItemDefinitionBuilder CloneFrom(string sourceItemId)
```

Parameters

sourceItemId [string](#)

The ID of the existing buildable item to clone from.

Returns

[BuildableItemDefinitionBuilder](#)

A builder initialized with the source item's properties, ready for customization.

Examples

```
var metalRack = BuildableItemCreator.CloneFrom("StorageRack-1x0.5")
    .WithBasicInfo("metal_rack_small", "Small Metal Storage Rack", "A
metal version")
    .WithBuildSound(BuildSoundType.Metal)
    .WithPricing(72f, 0.5f)
    .Build();
```

CreateBuilder()

Creates a new builder for composing a buildable item definition with full flexibility. Use fluent methods to configure the item, then call Build() to register it.

```
public static BuildableItemDefinitionBuilder CreateBuilder()
```

Returns

[BuildableItemDefinitionBuilder](#)

A new BuildableItemDefinitionBuilder instance for fluent configuration.

Examples

```
var item = BuildableItemCreator.CreateBuilder()
    .WithBasicInfo("my_rack", "Custom Storage Rack", "A custom storage rack")
    .WithBuildSound(BuildSoundType.Metal)
    .WithPricing(75f, 0.5f)
    .Build();
```

Class BuildableItemDefinition

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents a buildable item definition that can be placed in the game world. Extends [StorableItemDefinition](#) with building-specific properties.

```
public sealed class BuildableItemDefinition : StorableItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← [StorableItemDefinition](#) ← [BuildableItemDefinition](#)

Inherited Members

[StorableItemDefinition.BasePurchasePrice](#) , [StorableItemDefinition.ResellMultiplier](#) ,
[StorableItemDefinition.IsUnlocked](#) , [ItemDefinition.CreateInstance\(int\)](#) , [ItemDefinition.Equals\(object\)](#) ,
[ItemDefinition.GetHashCode\(\)](#) , [ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) ,
[ItemDefinition.StackLimit](#) , [ItemDefinition.Category](#) , [ItemDefinition.Icon](#) ,
[ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) , [ItemDefinition.Keywords](#) ,
[ItemDefinition.GUID](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Use [CreateBuilder\(\)](#) to create new buildable items, or [CloneFrom\(string\)](#) to create variants of existing items.

Properties

BuildSoundType

The sound type played when this item is built.

```
public BuildSoundType BuildSoundType { get; set; }
```

Property Value

[BuildSoundType](#)

LabelDisplayColor

The color displayed on the item's label in the UI.

```
public Color LabelDisplayColor { get; set; }
```

Property Value

Color

Class BuildableItemDefinitionBuilder

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Builder for composing buildable item definitions at runtime. Use fluent methods to configure buildable item properties before calling [Build\(\)](#).

```
public sealed class BuildableItemDefinitionBuilder
```

Inheritance

[object](#) ← BuildableItemDefinitionBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

Build()

Builds the buildable item definition, registers it with the game's registry, and returns a wrapper.

```
public BuildableItemDefinition Build()
```

Returns

[BuildableItemDefinition](#)

A wrapper around the created buildable item definition.

WithBasicInfo(string, string, string)

Sets the basic information for the buildable item.

```
public BuildableItemDefinitionBuilder WithBasicInfo(string id, string name,  
string description)
```

Parameters

id [string](#)

Unique identifier for the item (e.g., "my_custom_rack").

name [string](#)

Display name shown in UI.

description [string](#)

Item description shown in tooltips.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithBuildSound(BuildSoundType)

Sets the sound type played when this item is built.

```
public BuildableItemDefinitionBuilder WithBuildSound(BuildSoundType soundType)
```

Parameters

soundType [BuildSoundType](#)

The build sound type.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithCategory(ItemCategory)

Sets the category for inventory organization.

```
public BuildableItemDefinitionBuilder WithCategory(ItemCategory category)
```

Parameters

category [ItemCategory](#)

The item category.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithEquippable(Equippable)

Attaches an equippable component to this item, allowing it to be equipped by the player.

```
public BuildableItemDefinitionBuilder WithEquippable(Equippable equippable)
```

Parameters

equippable [Equippable](#)

The equippable wrapper to attach.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithIcon(Sprite)

Sets the icon sprite displayed for this item in UI.

```
public BuildableItemDefinitionBuilder WithIcon(Sprite icon)
```

Parameters

icon Sprite

The sprite to use as the item icon.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithKeywords(params string[])

Sets keywords used for filtering and searching this item.

```
public BuildableItemDefinitionBuilder WithKeywords(params string[] keywords)
```

Parameters

keywords [string](#)[]

Array of keywords.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithLabelColor(Color)

Sets the color of the label displayed in UI.

```
public BuildableItemDefinitionBuilder WithLabelColor(Color color)
```

Parameters

color Color

The color to use for the item label.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithLegalStatus(LegalStatus)

Sets the legal status of the item.

```
public BuildableItemDefinitionBuilder WithLegalStatus(LegalStatus status)
```

Parameters

status [LegalStatus](#)

Whether the item is legal or illegal.

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithPricing(float, float)

Configures the economic properties of the item.

```
public BuildableItemDefinitionBuilder WithPricing(float basePurchasePrice, float  
resellMultiplier = 0.5)
```

Parameters

basePurchasePrice [float](#)

Base price when buying from shops.

resellMultiplier [float](#)

Fraction of purchase price recovered when selling (0.0 to 1.0).

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithStackLimit(int)

Sets the maximum stack size for this item.

```
public BuildableItemDefinitionBuilder WithStackLimit(int limit)
```

Parameters

limit [int](#)

Maximum quantity per inventory slot (1-999).

Returns

[BuildableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

Enum ClothingApplicationType

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents how a clothing item is applied to the avatar. Mirrors ScheduleOne.Clothing.ELoadingType.

```
public enum ClothingApplicationType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Accessory = 2

Applied as a 3D accessory (separate mesh).

BodyLayer = 0

Applied as a body layer (flat texture on body mesh).

FaceLayer = 1

Applied as a face layer (flat texture on face mesh).

Enum ClothingColor

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents available clothing colors. Mirrors ScheduleOne.Clothing.EClothingColor.

```
public enum ClothingColor
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Beige = 12

Black = 4

Blue = 19

BrightPink = 25

Brown = 10

Charcoal = 3

Coral = 11

Crimson = 7

Cyan = 17

DarkGreen = 16

DarkGrey = 2

DeepBlue = 20

DeepPurple = 22

HotPink = 26

LightGreen = 15

LightGrey = 1

LightRed = 5

Lime = 14

Magenta = 24

Navy = 21

Orange = 8

Purple = 23

Red = 6

SkyBlue = 18

Tan = 9

White = 0

Yellow = 13

Class ClothingItemCreator

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Provides convenient static methods for creating custom clothing items. Use [CreateBuilder\(\)](#) for flexible configuration or [CloneFrom\(string\)](#) for variants.

```
public static class ClothingItemCreator
```

Inheritance

[object](#) ← ClothingItemCreator

Inherited Members

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

Methods

CloneFrom(ClothingItemDefinition)

Creates a new clothing item by cloning an existing one. Returns a builder pre-configured with all properties of the source item.

```
public static ClothingItemDefinitionBuilder CloneFrom(ClothingItemDefinition source)
```

Parameters

source [ClothingItemDefinition](#)

The clothing item definition to clone.

Returns

[ClothingItemDefinitionBuilder](#)

A builder pre-configured with the source item's properties.

Examples

```
var existingCap = ItemManager.GetItemDefinition("cap") as ClothingItemDefinition;
var variant = ClothingItemCreator.CloneFrom(existingCap)
    .WithBasicInfo("variant_cap", "Cap Variant", "A variant of the cap")
    .Build();
```

CloneFrom(string)

Creates a new clothing item by cloning an existing one by ID. Returns a builder pre-configured with all properties of the source item. You can then override specific properties before calling Build().

```
public static ClothingItemDefinitionBuilder CloneFrom(string sourceItemId)
```

Parameters

sourceItemId [string](#)

The ID of the clothing item to clone.

Returns

[ClothingItemDefinitionBuilder](#)

A builder pre-configured with the source item's properties.

Examples

```
// Clone the base game cap and customize it
var customCap = ClothingItemCreator.CloneFrom("cap")
    .WithBasicInfo("stay_silly_cap", "Stay Silly Cap", "A silly custom cap")
    .WithClothingAsset("BigWillyMod/Accessories/StaySillyCap")
    .WithColorable(false)
    .Build();
```

CreateBuilder()

Creates a new builder for composing a clothing item definition with full flexibility. Use fluent methods to configure the item, then call Build() to register it.

```
public static ClothingItemDefinitionBuilder CreateBuilder()
```

Returns

[ClothingItemDefinitionBuilder](#)

A new ClothingItemDefinitionBuilder instance for fluent configuration.

Examples

```
var hat = ClothingItemCreator.CreateBuilder()
    .WithBasicInfo("my_hat", "Custom Hat", "A fancy custom hat")
    .WithSlot(ClothingSlot.Head)
    .WithApplicationType(ClothingApplicationType.Accessory)
    .WithClothingAsset("MyMod/Accessories/CustomHat")
    .WithDefaultColor(ClothingColor.Black)
    .Build();
```

Class ClothingItemDefinition

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents a clothing item definition that can be worn by the player. Extends [StorableItemDefinition](#) with clothing-specific properties.

```
public sealed class ClothingItemDefinition : StorableItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← [StorableItemDefinition](#) ← [ClothingItemDefinition](#)

Inherited Members

[StorableItemDefinition.BasePurchasePrice](#) , [StorableItemDefinition.ResellMultiplier](#) ,
[StorableItemDefinition.IsUnlocked](#) , [ItemDefinition.CreateInstance\(int\)](#) , [ItemDefinition.Equals\(object\)](#) ,
[ItemDefinition.GetHashCode\(\)](#) , [ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) ,
[ItemDefinition.StackLimit](#) , [ItemDefinition.Category](#) , [ItemDefinition.Icon](#) ,
[ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) , [ItemDefinition.LabelDisplayColor](#) ,
[ItemDefinition.Keywords](#) , [ItemDefinition.GUID](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Use [CreateBuilder\(\)](#) to create new clothing items, or [CloneFrom\(string\)](#) to create variants of existing items.

Properties

ApplicationType

How this clothing item is applied to the avatar.

```
public ClothingApplicationType ApplicationType { get; set; }
```

Property Value

[ClothingApplicationType](#)

ClothingAssetPath

The asset path to the clothing prefab or layer in Resources.

```
public string ClothingAssetPath { get; set; }
```

Property Value

[string](#)

Colorable

Whether this clothing item can be colored by the player.

```
public bool Colorable { get; set; }
```

Property Value

[bool](#)

DefaultColor

The default color for this clothing item.

```
public ClothingColor DefaultColor { get; set; }
```

Property Value

[ClothingColor](#)

Slot

The clothing slot this item occupies.

```
public ClothingSlot Slot { get; set; }
```

Property Value

[ClothingSlot](#)

SlotsToBlock

List of clothing slots this item blocks when equipped.

```
public List<ClothingSlot> SlotsToBlock { get; set; }
```

Property Value

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

Class ClothingItemDefinitionBuilder

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Builder for composing clothing item definitions at runtime. Use fluent methods to configure clothing properties before calling [Build\(\)](#).

```
public sealed class ClothingItemDefinitionBuilder
```

Inheritance

[object](#) ← ClothingItemDefinitionBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

Build()

Builds the clothing item definition, registers it with the game's registry, and returns a wrapper.

```
public ClothingItemDefinition Build()
```

Returns

[ClothingItemDefinition](#)

A wrapper around the created clothing item definition.

WithApplicationType(ClothingApplicationType)

Sets how this clothing item is applied to the avatar.

```
public ClothingItemDefinitionBuilder WithApplicationType(ClothingApplicationType applicationType)
```

Parameters

applicationType [ClothingApplicationType](#)

The application type.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithBasicInfo(string, string, string)

Sets the basic information for the clothing item.

```
public ClothingItemDefinitionBuilder WithBasicInfo(string id, string name, string description)
```

Parameters

id [string](#)

Unique identifier for the item (e.g., "my_custom_hat").

name [string](#)

Display name shown in UI.

description [string](#)

Item description shown in tooltips.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithBlockedSlots(params ClothingSlot[])

Sets the list of clothing slots this item blocks when equipped.

```
public ClothingItemDefinitionBuilder WithBlockedSlots(params ClothingSlot[] slots)
```

Parameters

`slots` [ClothingSlot\[\]](#)

Array of slots to block.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithClothingAsset(string)

Sets the asset path to the clothing prefab or layer.

```
public ClothingItemDefinitionBuilder WithClothingAsset(string assetPath)
```

Parameters

`assetPath` [string](#)

Resources path to the clothing asset.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithColorable(bool)

Sets whether this clothing item can be colored.

```
public ClothingItemDefinitionBuilder WithColorable(bool colorable)
```

Parameters

[colorable](#) [bool](#)

True if colorable, false otherwise.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithDefaultColor(ClothingColor)

Sets the default color for this clothing item.

```
public ClothingItemDefinitionBuilder WithDefaultColor(ClothingColor color)
```

Parameters

[color](#) [ClothingColor](#)

The default color.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithIcon(Sprite)

Sets the icon sprite displayed for this item in UI.

```
public ClothingItemDefinitionBuilder WithIcon(Sprite icon)
```

Parameters

`icon` Sprite

The sprite to use as the item icon.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithKeywords(params string[])

Sets keywords used for filtering and searching this item.

```
public ClothingItemDefinitionBuilder WithKeywords(params string[] keywords)
```

Parameters

`keywords` `string[]`

Array of keywords.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithLabelColor(Color)

Sets the color of the label displayed in UI.

```
public ClothingItemDefinitionBuilder WithLabelColor(Color color)
```

Parameters

color Color

The color to use for the item label.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithPricing(float, float)

Configures the economic properties of the item.

```
public ClothingItemDefinitionBuilder WithPricing(float basePurchasePrice, float resellMultiplier = 0.5)
```

Parameters

basePurchasePrice [float](#)

Base price when buying from shops.

`resellMultiplier` [float](#)

Fraction of purchase price recovered when selling (0.0 to 1.0).

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithSlot(ClothingSlot)

Sets the clothing slot this item occupies.

```
public ClothingItemDefinitionBuilder WithSlot(ClothingSlot slot)
```

Parameters

`slot` [ClothingSlot](#)

The clothing slot.

Returns

[ClothingItemDefinitionBuilder](#)

The builder instance for fluent chaining.

Class ClothingItemInstance

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents a clothing item instance in the game world (physical clothing you own). Extends [ItemInstance](#) with color information.

```
public class ClothingItemInstance : ItemInstance
```

Inheritance

[object](#) ← [ItemInstance](#) ← ClothingItemInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

Color

The color of this clothing instance.

```
public ClothingColor Color { get; set; }
```

Property Value

[ClothingColor](#)

Definition

The clothing definition (template) this instance was created from.

```
public ClothingItemDefinition Definition { get; }
```

Property Value

[ClothingItemDefinition](#)

Enum ClothingSlot

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents the slot where a clothing item can be equipped. Mirrors ScheduleOne.Clothing. EClothingSlot.

```
public enum ClothingSlot
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Bottom = 1

Bottom slot (pants, shorts).

Eyes = 7

Eyes slot (glasses, sunglasses).

Feet = 0

Feet slot (shoes, boots).

Hands = 5

Hands slot (gloves).

Head = 8

Head slot (hats, caps, helmets).

Neck = 6

Neck slot (necklaces, scarves).

Outerwear = 4

Outerwear slot (jackets, coats).

Top = 3

Top slot (shirts).

Waist = 2

Waist slot (belts).

wrist = 9

Wrist slot (watches, bracelets).

Class Equippable

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents an equippable component that can be attached to items. Provides a wrapper around the game's native equippable system.

```
public class Equippable
```

Inheritance

[object](#) ← Equippable

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

CanInteractWhenEquipped

Gets or sets whether the player can interact with objects when this item is equipped.

```
public bool CanInteractWhenEquipped { get; set; }
```

Property Value

[bool](#)

CanPickUpWhenEquipped

Gets or sets whether the player can pick up items when this item is equipped.

```
public bool CanPickUpWhenEquipped { get; set; }
```

Property Value

[bool](#) ↗

Methods

Equip(ItemInstance)

Called when this item is equipped by the player. Override this in derived classes to implement custom equip behavior.

```
public virtual void Equip(ItemInstance item)
```

Parameters

[item](#) [ItemInstance](#)

The item instance being equipped.

Unequip()

Called when this item is unequipped by the player. Override this in derived classes to implement custom unequip behavior.

```
public virtual void Unequip()
```

Class EquippableBuilder

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Builder for creating equippable components that can be attached to items. Use this to create custom equippable behavior for items.

```
public sealed class EquippableBuilder
```

Inheritance

[object](#) ← EquippableBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

EquippableBuilder()

```
public EquippableBuilder()
```

Methods

Build()

Builds and finalizes the equippable component. Configures the GameObject to be persistent and inactive (prefab-like state).

```
public Equippable Build()
```

Returns

[Equippable](#)

A wrapper around the created equippable component.

CreateBasicEquippable(string)

Creates a basic equippable GameObject with the default Equippable component.

```
public EquippableBuilder CreateBasicEquippable(string name = null)
```

Parameters

[name](#) `string` ↗

Optional name for the GameObject.

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

CreateEquippable<T>(string)

Creates an equippable GameObject with the specified equippable component type.

```
public EquippableBuilder CreateEquippable<T>(string name = null) where T : Equippable
```

Parameters

name [string](#)

Optional name for the GameObject. If not provided, uses the type name.

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

Type Parameters

T

The type of equippable component to create. Must inherit from the game's Equippable class.

CreateViewModelEquippable(string)

Creates a viewmodel equippable GameObject with Equippable_Visual component. This allows the item to be held in first-person with a 3D model.

```
public EquippableBuilder CreateViewModelEquippable(string name = null)
```

Parameters

name [string](#)

Optional name for the GameObject.

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

WithAvatarEquippable(string, AvatarHand, string)

Configures the third-person avatar equippable animation. Only applies to.viewmodel equippables created with [CreateViewModelEquippable\(string\)](#).

```
public EquippableBuilder WithAvatarEquippable(string assetPath, AvatarHand hand = AvatarHand.Right, string animationTrigger = "RightArm_Hold_ClosedHand")
```

Parameters

assetPath [string](#)

Resources path to the AvatarEquippable prefab (e.g., "Equippables/MyItem").

hand [AvatarHand](#)

Which hand holds the item in third-person (Left or Right).

animationTrigger [string](#)

Animation trigger/bool name for third-person animation.

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

WithInteraction(bool, bool)

Configures interaction and pickup capabilities when this item is equipped.

```
public EquippableBuilder WithInteraction(bool canInteract, bool canPickup)
```

Parameters

canInteract [bool](#)

Whether the player can interact with objects when this is equipped.

canPickup [bool](#)

Whether the player can pick up items when this is equipped.

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

WithUseCallback(Action<ItemInstance>)

Registers a callback to be invoked when the player uses this item (clicks while holding it). Only applies to viewmodel equippables created with [CreateViewModelEquippable\(string\)](#). The callback receives the ItemInstance being used.

```
public EquippableBuilder WithUseCallback(Action<ItemInstance> callback)
```

Parameters

`callback Action<ItemInstance>`

Callback to invoke when the item is used.

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

WithViewModelTransform(Vector3, Vector3, Vector3?)

Configures the first-person viewmodel transform settings. Only applies to viewmodel equippables created with [CreateViewModelEquippable\(string\)](#).

```
public EquippableBuilder WithViewModelTransform(Vector3 position, Vector3 rotation,  
Vector3? scale = null)
```

Parameters

position Vector3

Local position offset for the viewmodel.

rotation Vector3

Local euler angles for the viewmodel.

scale Vector3?

Local scale for the viewmodel (default: Vector3.one).

Returns

[EquippableBuilder](#)

The builder instance for fluent chaining.

Enum ItemCategory

Namespace: [S1API.Items](#)

Assembly: S1API.dll

A list of item categories available in-game.

```
public enum ItemCategory
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`Cash = 6`

Represents cash-based items.

`Clothing = 11`

Represents clothing items.

`Consumable = 7`

Represents items such as Cuke, Energy Drink, etc.

`Decoration = 10`

Represents items such as GoldBar, WallClock, WoodSign, etc.

`Equipment = 8`

Represents items such as Drying Rack, Brick Press, Mixing Station, etc.

`Furniture = 4`

Represents items such as TV, Trash Can, Bed, etc.

`Growing = 2`

Represents items such as Soil, Fertilizer, Pots, etc.

`Ingredient = 9`

Represents items such as Acid, Banana, Chili, etc.

Lighting = 5

Represents items such as Floor Lamps, Halogen Lights, etc.

Packaging = 1

Represents items such as Baggies, Bricks, Jars, etc.

Product = 0

Represents items such as Cocaine, Weed, etc. Oddly, SpeedGrow is in this category as of (v0.3.4f8).

Tools = 3

Represents equipment tools such as the clippers. Oddly, trash bags is in this category as of (v0.3.4f8).

Class ItemCreator

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Provides convenient static methods for creating custom items. Use [CreateBuilder\(\)](#) for flexible configuration or [CreateItem\(string, string, string, ItemCategory, int, float, float, LegalStatus, Sprite, Equippable\)](#) for quick creation.

```
public static class ItemCreator
```

Inheritance

[object](#) ← ItemCreator

Inherited Members

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

Remarks

All items in Schedule One are storable items (StorableItemDefinition), so both methods create the same type.

Methods

CreateBuilder()

Creates a new builder for composing an item definition with full flexibility. Use fluent methods to configure the item, then call Build() to register it.

```
public static StorableItemDefinitionBuilder CreateBuilder()
```

Returns

[StorableItemDefinitionBuilder](#)

A new StorableItemDefinitionBuilder instance for fluent configuration.

Examples

```
var item = ItemCreator.CreateBuilder()
    .WithBasicInfo("my_tool", "Custom Tool", "A custom tool", ItemCategory.Tools)
    .WithStackLimit(5)
    .WithPricing(25f, 0.3f)
    .Build();
```

CreateEquippableBuilder()

Creates a new equippable builder for creating custom equippable components. Use this to create equippable behavior that can be attached to items.

```
public static EquippableBuilder CreateEquippableBuilder()
```

Returns

[EquippableBuilder](#)

A new EquippableBuilder instance.

Examples

```
var equippable = ItemCreator.CreateEquippableBuilder()
    .CreateBasicEquippable("MyEquippable")
    .WithInteraction(canInteract: true, canPickup: true)
    .Build();
```

CreateItem(string, string, string, ItemCategory, int, float, float, LegalStatus, Sprite, Equippable)

Creates an item with common parameters in a single call. The item is automatically registered with the game's registry.

```
public static StorableItemDefinition CreateItem(string id, string name, string description, ItemCategory category, int stackLimit = 10, float basePurchasePrice = 10, float resellMultiplier = 0.5, LegalStatus legalStatus = LegalStatus.Legal, Sprite icon = null, Equippable equippable = null)
```

Parameters

id [string](#)

Unique identifier for the item (e.g., "my_custom_tool").

name [string](#)

Display name shown in UI.

description [string](#)

Item description shown in tooltips.

category [ItemCategory](#)

Item category for inventory organization.

stackLimit [int](#)

Maximum quantity per inventory slot (default: 10).

basePurchasePrice [float](#)

Base price when buying from shops (default: 10).

resellMultiplier [float](#)

Fraction of purchase price recovered when selling (default: 0.5).

legalStatus [LegalStatus](#)

Whether the item is legal or illegal (default: Legal).

icon [Sprite](#)

Optional sprite to use as the item icon.

equippable [Equippable](#)

Optional equippable component to attach.

Returns

[StorableItemDefinition](#)

A wrapper around the created item definition.

Examples

```
var item = ItemCreator.CreateItem(  
    id: "my_tool",  
    name: "Custom Tool",  
    description: "A custom tool for crafting",  
    category: ItemCategory.Tools,  
    stackLimit: 5,  
    basePurchasePrice: 25f  
)
```

Class ItemDefinition

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents an item definition in-game. This is the base wrapper class for all item types.

```
public class ItemDefinition
```

Inheritance

[object](#) ← ItemDefinition

Derived

[SeedDefinition](#), [StorableItemDefinition](#), [CashDefinition](#), [PackagingDefinition](#), [ProductDefinition](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

For creating custom items, use [ItemCreator](#) which creates StorableItemDefinition instances.
ItemDefinition is primarily used for reading existing items via [GetItemDefinition\(string\)](#).

Properties

AvailableInDemo

Whether this item is available in the demo version of the game.

```
public bool AvailableInDemo { get; set; }
```

Property Value

[bool](#)

Category

The category for inventory sorting.

```
public ItemCategory Category { get; set; }
```

Property Value

[ItemCategory](#)

Description

A short description for this item.

```
public string Description { get; set; }
```

Property Value

[string](#)

GUID

Gets the globally unique identifier (GUID) of the item, which is equivalent to the ID.

```
public string GUID { get; }
```

Property Value

[string](#)

ID

The unique ID of this item.

```
public string ID { get; set; }
```

Property Value

[string](#)

Icon

The icon for this item.

```
public Sprite Icon { get; set; }
```

Property Value

Sprite

Keywords

Any keywords used to filter/search this item.

```
public string[] Keywords { get; set; }
```

Property Value

[string](#)

LabelDisplayColor

The color of the label shown in UI.

```
public Color LabelDisplayColor { get; set; }
```

Property Value

Color

LegalStatus

Legal status of the item (e.g., illegal drugs).

```
public LegalStatus LegalStatus { get; set; }
```

Property Value

[LegalStatus](#)

Name

The display name for this item.

```
public string Name { get; set; }
```

Property Value

[string](#) ↗

StackLimit

Stack limit for this item (max quantity per slot).

```
public int StackLimit { get; set; }
```

Property Value

[int](#)

Methods

CreateInstance(int)

Creates a new item instance with the specified quantity.

```
public virtual ItemInstance CreateInstance(int quantity = 1)
```

Parameters

[quantity](#) [int](#)

Returns

[ItemInstance](#)

Equals(object?)

Determines whether the specified object is equal to the current object.

```
public override bool Equals(object? obj)
```

Parameters

[obj](#) [object](#)

The object to compare with the current object.

Returns

[bool](#)

`true` if the specified object is an [ItemDefinition](#) and has the same S1ItemDefinition; otherwise, `false`.

GetHashCode()

Serves as the default hash function.

```
public override int GetHashCode()
```

Returns

[int](#)

A hash code for the current object based on S1ItemDefinition.

Operators

operator ==(ItemDefinition?, ItemDefinition?)

Determines whether two [ItemDefinition](#) instances are equal.

```
public static bool operator ==(ItemDefinition? a, ItemDefinition? b)
```

Parameters

a [ItemDefinition](#)

The first [ItemDefinition](#) to compare.

b [ItemDefinition](#)

The second [ItemDefinition](#) to compare.

Returns

[bool](#)

`true` if both instances are equal or have the same S1ItemDefinition; otherwise, `false`.

operator !=(ItemDefinition?, ItemDefinition?)

Determines whether two [ItemDefinition](#) instances are not equal.

```
public static bool operator !=(ItemDefinition? a, ItemDefinition? b)
```

Parameters

a [ItemDefinition](#)

The first [ItemDefinition](#) to compare.

b [ItemDefinition](#)

The second [ItemDefinition](#) to compare.

Returns

[bool](#)

`true` if the instances are not equal; otherwise, `false`.

Class ItemInstance

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents an item instance in the game world (physical item you own).

```
public class ItemInstance
```

Inheritance

[object](#) ← ItemInstance

Derived

[ClothingItemInstance](#), [CashInstance](#), [ProductInstance](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

Definition

The definition (template) this instance was created from.

```
public ItemDefinition Definition { get; }
```

Property Value

[ItemDefinition](#)

IsStackable

Whether this instance is stackable (based on StackLimit).

```
public bool IsStackable { get; }
```

Property Value

[bool](#)

Quantity

Current quantity of this item (stacks).

```
public int Quantity { get; set; }
```

Property Value

[int](#)

Class ItemManager

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Provides access to managing items across the game.

```
public static class ItemManager
```

Inheritance

[object](#) ← ItemManager

Inherited Members

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

Methods

GetAllItemDefinitions()

Gets all item definitions registered in the game's registry.

```
public static List<ItemDefinition> GetAllItemDefinitions()
```

Returns

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

A list of all registered item definitions, wrapped with S1API types.

GetItemDefinition(string)

Gets the definition of an item by its ID.

```
public static ItemDefinition GetItemDefinition(string itemID)
```

Parameters

itemID [string](#)

The ID of the item.

Returns

[ItemDefinition](#)

An instance of the item definition.

RegisterItem(ItemDefinition)

Manually registers an item definition with the game's registry. This is typically handled automatically by [ItemCreator](#) methods, but can be used for advanced scenarios.

```
public static void RegisterItem(ItemDefinition definition)
```

Parameters

definition [ItemDefinition](#)

The item definition to register.

Class ItemSlotInstance

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents an item slot within the game. These are present within storage, the hot bar, etc.

```
public class ItemSlotInstance
```

Inheritance

[object](#) ← ItemSlotInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

ItemInstance

The item instance the slot contains.

```
public ItemInstance? ItemInstance { get; }
```

Property Value

[ItemInstance](#)

Quantity

The quantity of item in this slot.

```
public int Quantity { get; }
```

Property Value

[int ↗](#)

Methods

AddQuantity(int)

Adds a quantity to the item in this slot. NOTE: Negative numbers are supported and allowed.

```
public void AddQuantity(int amount)
```

Parameters

amount [int ↗](#)

Enum LegalStatus

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents the legal status of an item (e.g., legal or illegal).

```
public enum LegalStatus
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Illegal = 1

Legal = 0

Class StorableItemDefinition

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Represents an item definition that can be purchased, sold, and stored in inventories. Extends [Item Definition](#) with economic properties.

```
public class StorableItemDefinition : ItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← StorableItemDefinition

Derived

[BuildableItemDefinition](#), [ClothingItemDefinition](#)

Inherited Members

[ItemDefinition.CreateInstance\(int\)](#), [ItemDefinition.Equals\(object\)](#), [ItemDefinition.GetHashCode\(\)](#),
[ItemDefinition.ID](#), [ItemDefinition.Name](#), [ItemDefinition.Description](#), [ItemDefinition.StackLimit](#),
[ItemDefinition.Category](#), [ItemDefinition.Icon](#), [ItemDefinition.AvailableInDemo](#),
[ItemDefinition.LegalStatus](#), [ItemDefinition.LabelDisplayColor](#), [ItemDefinition.Keywords](#),
[ItemDefinition.GUID](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

In Schedule One, all items are StorableItemDefinition or subclasses thereof. The base ItemDefinition class is abstract and not used directly in gameplay.

Properties

BasePurchasePrice

The base purchase price for this item in shops.

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

Property Value

[float](#) ↗

IsUnlocked

Gets whether this item is currently unlocked (available for purchase/use).

```
public bool IsUnlocked { get; }
```

Property Value

[bool](#) ↗

ResellMultiplier

The resell multiplier (0.0 to 1.0) that determines how much of the purchase price can be recovered when selling the item.

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

Property Value

[float](#) ↗

Class StorableItemDefinitionBuilder

Namespace: [S1API.Items](#)

Assembly: S1API.dll

Builder for composing item definitions at runtime. Use fluent methods to configure item properties before calling [Build\(\)](#).

```
public sealed class StorableItemDefinitionBuilder
```

Inheritance

[object](#) ← StorableItemDefinitionBuilder

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

All items in Schedule One are StorableItemDefinition (or subclasses thereof). The base ItemDefinition class is never used directly in the game.

Methods

Build()

Builds the item definition, registers it with the game's registry, and returns a wrapper.

```
public StorableItemDefinition Build()
```

Returns

[StorableItemDefinition](#)

A wrapper around the created storable item definition.

WithBasicInfo(string, string, string, ItemCategory)

Sets the basic information for the item.

```
public StorableItemDefinitionBuilder WithBasicInfo(string id, string name, string description, ItemCategory category)
```

Parameters

id [string](#)

Unique identifier for the item (e.g., "my_custom_tool").

name [string](#)

Display name shown in UI.

description [string](#)

Item description shown in tooltips.

category [ItemCategory](#)

Item category for inventory organization.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithDemoAvailability(bool)

Sets whether this item is available in the demo version of the game.

```
public StorableItemDefinitionBuilder WithDemoAvailability(bool available)
```

Parameters

available `bool`

True if available in demo, false otherwise.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithEquippable(Equippable)

Attaches an equippable component to this item, allowing it to be equipped by the player.

```
public StorableItemDefinitionBuilder WithEquippable(Equippable equippable)
```

Parameters

equippable [Equippable](#)

The equippable wrapper to attach.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithIcon(Sprite)

Sets the icon sprite displayed for this item in UI.

```
public StorableItemDefinitionBuilder WithIcon(Sprite icon)
```

Parameters

icon Sprite

The sprite to use as the item icon.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithKeywords(params string[])

Sets keywords used for filtering and searching this item.

```
public StorableItemDefinitionBuilder WithKeywords(params string[] keywords)
```

Parameters

keywords [string\[\]](#)

Array of keywords.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithLabelColor(Color)

Sets the color of the label displayed in UI.

```
public StorableItemDefinitionBuilder WithLabelColor(Color color)
```

Parameters

color Color

The color to use for the item label.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithLegalStatus(LegalStatus)

Sets the legal status of the item.

```
public StorableItemDefinitionBuilder WithLegalStatus(LegalStatus status)
```

Parameters

status [LegalStatus](#)

Whether the item is legal or illegal.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithPricing(float, float)

Configures the economic properties of the item.

```
public StorableItemDefinitionBuilder WithPricing(float basePurchasePrice, float resellMultiplier = 0.5)
```

Parameters

basePurchasePrice [float](#)

Base price when buying from shops.

resellMultiplier [float](#)

Fraction of purchase price recovered when selling (0.0 to 1.0).

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithStackLimit(int)

Sets the maximum stack size for this item.

```
public StorableItemDefinitionBuilder WithStackLimit(int limit)
```

Parameters

limit [int](#)

Maximum quantity per inventory slot (1-999).

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

WithStoredItem(GameObject)

Assigns a custom StoredItem prefab for this definition.

```
public StorableItemDefinitionBuilder WithStoredItem(GameObject storedItemPrefab)
```

Parameters

storedItemPrefab GameObject

Prefab containing a StoredItem component.

Returns

[StorableItemDefinitionBuilder](#)

The builder instance for fluent chaining.

Namespace S1API.Law

Classes

[CheckpointInfo](#)

Contains comprehensive state information about a checkpoint.

[CheckpointManager](#)

Manages road checkpoints throughout the game world. Provides control over checkpoint activation and officer assignment.

[CurfewManager](#)

Manages the curfew system, including activation state and timing. The curfew restricts player movement during nighttime hours (9 PM to 5 AM).

[FootPatrolRoute](#)

Represents a foot patrol route with waypoints for police officers to follow.

[LawController](#)

Controls law enforcement intensity and automatic activity systems. Manages the automatic evaluation of checkpoints, patrols, and other law enforcement activities.

[LawManager](#)

Provides access to law enforcement and police dispatch functionality. Manages police responses, patrol operations, and wanted levels.

[PatrolGroup](#)

Represents a group of police officers on foot patrol. Manages patrol group movement, waypoints, and member coordination.

[PlayerCrimeData](#)

Manages a player's criminal record, wanted level, and police pursuit state.

[VehiclePatrolRoute](#)

Represents a vehicle patrol route with waypoints for police vehicles to follow.

Enums

[CheckpointLocation](#)

Represents the available checkpoint locations in the game world.

[PursuitLevel](#)

Represents the intensity level of a police pursuit. Higher levels indicate more aggressive police response.

Class CheckpointInfo

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Contains comprehensive state information about a checkpoint.

```
public sealed class CheckpointInfo
```

Inheritance

[object](#) ← CheckpointInfo

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

CheckpointInfo()

```
public CheckpointInfo()
```

Properties

AreBothGatesClosed

Whether both gates are currently closed.

```
public bool AreBothGatesClosed { get; }
```

PropertyValue

[bool](#)

AssignedOfficerCount

The number of police officers currently assigned to this checkpoint.

```
public int AssignedOfficerCount { get; }
```

PropertyValue

[int](#)

IsAnyGateOpen

Whether at least one gate is currently open.

```
public bool IsAnyGateOpen { get; }
```

PropertyValue

[bool](#)

IsEnabled

Whether the checkpoint is currently enabled and operational.

```
public bool IsEnabled { get; }
```

PropertyValue

[bool](#)

IsGate1Open

Whether the checkpoint's first gate is currently open.

```
public bool IsGate1Open { get; }
```

Property Value

[bool](#)

IsGate2Open

Whether the checkpoint's second gate is currently open.

```
public bool IsGate2Open { get; }
```

Property Value

[bool](#)

IsOperational

Whether the checkpoint is enabled and has at least one officer assigned.

```
public bool IsOperational { get; }
```

Property Value

[bool](#)

Location

The checkpoint's location identifier.

```
public CheckpointLocation Location { get; }
```

Property Value

[CheckpointLocation](#)

Position

The checkpoint's position in world space.

```
public Vector3 Position { get; }
```

Property Value

Vector3

Enum CheckpointLocation

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Represents the available checkpoint locations in the game world.

```
public enum CheckpointLocation
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Docks = 1

Docks checkpoint location.

NorthResidential = 2

North Residential checkpoint location.

WestResidential = 3

West Residential checkpoint location.

Western = 0

Western checkpoint location.

Class CheckpointManager

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Manages road checkpoints throughout the game world. Provides control over checkpoint activation and officer assignment.

```
public static class CheckpointManager
```

Inheritance

[object](#) ← CheckpointManager

Inherited Members

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

Remarks

IMPORTANT: The game includes an automatic checkpoint evaluation system that runs every in-game minute. This system may automatically enable checkpoints based on conditions such as:

- Law enforcement intensity level (see [LawController](#))
- Time of day
- Curfew status
- Player distance from checkpoint

Checkpoints require law enforcement intensity ≥ 5 (default) to activate automatically. Other conditions include: time of day, curfew status, player distance (50+ units), and officer availability. To prevent automatic re-enabling, set intensity to 1-4 using [SetIntensityLevel\(int\)](#).

Methods

DisableAllCheckpoints()

Disables all checkpoints in the game world.

```
public static void DisableAllCheckpoints()
```

Remarks

WARNING: The automatic checkpoint evaluation system may re-enable checkpoints based on game conditions. Consider adjusting law enforcement intensity instead.

EnableAllCheckpoints(int)

Enables all checkpoints in the game world.

```
public static void EnableAllCheckpoints(int officersPerCheckpoint = 2)
```

Parameters

officersPerCheckpoint [int](#)

The number of officers to assign to each checkpoint.

GetAllCheckpointInfo()

Gets information about all checkpoints in the game.

```
public static List<CheckpointInfo> GetAllCheckpointInfo()
```

Returns

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

A list of CheckpointInfo objects for all checkpoint locations.

GetAssignedOfficerCount(CheckpointLocation)

Gets the number of officers currently assigned to a checkpoint.

```
public static int GetAssignedOfficerCount(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to query.

Returns

[int](#)

The number of assigned officers, or 0 if the checkpoint is not found.

GetAssignedOfficers(CheckpointLocation)

Gets the NPCs (police officers) currently assigned to a checkpoint.

```
public static List<NPC> GetAssignedOfficers(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to query.

Returns

[List](#)<NPC>

A list of NPCs assigned to the checkpoint, or an empty list if none.

GetCheckpointInfo(CheckpointLocation)

Gets comprehensive information about a checkpoint's current state.

```
public static CheckpointInfo GetCheckpointInfo(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to query.

Returns

[CheckpointInfo](#)

A CheckpointInfo object containing state information, or null if the checkpoint is not found.

GetCheckpointPosition(CheckpointLocation)

Gets the world position of a checkpoint.

```
public static Vector3 GetCheckpointPosition(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to query.

Returns

[Vector3](#)

The checkpoint's position in world space, or Vector3.zero if not found.

IsCheckpointEnabled(CheckpointLocation)

Checks if a checkpoint at the specified location is currently enabled.

```
public static bool IsCheckpointEnabled(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to check.

Returns

[bool](#) ↗

True if the checkpoint is enabled, false otherwise.

IsGate1Open(CheckpointLocation)

Checks if a checkpoint's first gate is currently open.

```
public static bool IsGate1Open(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to check.

Returns

[bool](#) ↗

True if gate 1 is open, false otherwise.

IsGate2Open(CheckpointLocation)

Checks if a checkpoint's second gate is currently open.

```
public static bool IsGate2Open(CheckpointLocation location)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to check.

Returns

[bool](#) ↗

True if gate 2 is open, false otherwise.

SetCheckpointEnabled(CheckpointLocation, bool, int)

Enables or disables a checkpoint at the specified location.

```
public static void SetCheckpointEnabled(CheckpointLocation location, bool enabled,  
int requestedOfficers = 2)
```

Parameters

location [CheckpointLocation](#)

The checkpoint location to modify.

enabled [bool](#) ↗

Whether to enable or disable the checkpoint.

requestedOfficers [int](#) ↗

The number of officers to assign to the checkpoint when enabling.

Remarks

Officers will be pulled from the nearest police station.

If insufficient officers are available, fewer than requested may be assigned.

WARNING: The automatic checkpoint evaluation system may re-enable disabled checkpoints based on game conditions. See [CheckpointManager](#) remarks for details.

Class CurfewManager

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Manages the curfew system, including activation state and timing. The curfew restricts player movement during nighttime hours (9 PM to 5 AM).

```
public static class CurfewManager
```

Inheritance

[object](#) ← CurfewManager

Inherited Members

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

Fields

CurfewEndTime

The time (in 24-hour format) when the curfew ends. Default: 05:00 (5 AM).

```
public const int CurfewEndTime = 500
```

Field Value

[int](#)

CurfewStartTime

The time (in 24-hour format) when the curfew officially begins. Default: 21:00 (9 PM).

```
public const int CurfewStartTime = 2100
```

Field Value

[int ↗](#)

HardCurfewStartTime

The time (in 24-hour format) when the hard curfew begins. During hard curfew, violations result in immediate police response. Default: 21:15 (9:15 PM).

```
public const int HardCurfewStartTime = 2115
```

Field Value

[int ↗](#)

HourBeforeCurfew

The time (in 24-hour format) when curfew warnings begin appearing. Default: 20:00 (8 PM) - Two hours before curfew starts.

```
public const int HourBeforeCurfew = 2000
```

Field Value

[int ↗](#)

WarningTime

The time (in 24-hour format) when the 30-minute warning is displayed. Default: 20:30 (8:30 PM).

```
public const int WarningTime = 2030
```

Field Value

[int ↗](#)

Properties

IsCurrentlyActive

Gets a value indicating whether the curfew is currently active. True between 9 PM and 5 AM when curfew is enabled.

```
public static bool IsCurrentlyActive { get; }
```

Property Value

[bool ↗](#)

IsEnabled

Gets a value indicating whether the curfew system is enabled. When disabled, no curfew violations will occur.

```
public static bool IsEnabled { get; }
```

Property Value

[bool ↗](#)

IsHardCurfewActive

Gets a value indicating whether hard curfew is currently active. During hard curfew (9:15 PM - 5 AM), violations result in immediate police dispatch.

```
public static bool IsHardCurfewActive { get; }
```

Property Value

[bool](#) ↗

Methods

DisableCurfew()

Disables the curfew system. VMS boards will be hidden and no curfew violations will occur.

```
public static void DisableCurfew()
```

EnableCurfew()

Enables the curfew system. VMS boards will display curfew information and violations will be enforced.

```
public static void EnableCurfew()
```

IsWithinCurfewHours()

Checks if the current game time is within the curfew period.

```
public static bool IsWithinCurfewHours()
```

Returns

[bool](#)

True if the current time is between 9 PM and 5 AM.

IsWithinHardCurfewHours()

Checks if the current game time is within the hard curfew period.

```
public static bool IswithinHardCurfewHours()
```

Returns

[bool](#)

True if the current time is between 9:15 PM and 5 AM.

MinutesUntilCurfew()

Gets the number of minutes until curfew starts.

```
public static int MinutesUntilCurfew()
```

Returns

[int](#)

Minutes until curfew, or 0 if curfew is already active or disabled.

MinutesUntilCurfewEnds()

Gets the number of minutes until curfew ends.

```
public static int MinutesUntilCurfewEnds()
```

Returns

[int](#)

Minutes until curfew ends, or 0 if curfew is not currently active.

Class FootPatrolRoute

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Represents a foot patrol route with waypoints for police officers to follow.

```
public sealed class FootPatrolRoute
```

Inheritance

[object](#) ← FootPatrolRoute

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Position

Gets the position of the route object itself.

```
public Vector3 Position { get; }
```

Property Value

Vector3

RouteName

Gets the name of this patrol route.

```
public string RouteName { get; }
```

Property Value

[string](#)

StartWaypointIndex

Gets the starting waypoint index for this route.

```
public int StartWaypointIndex { get; }
```

Property Value

[int](#)

WaypointCount

Gets the number of waypoints in this route.

```
public int WaypointCount { get; }
```

Property Value

[int](#)

Methods

GetWaypointPosition(int)

Gets the position of a waypoint at the specified index.

```
public Vector3 GetWaypointPosition(int index)
```

Parameters

index [int](#)

The waypoint index.

Returns

Vector3

The waypoint position, or Vector3.zero if invalid index.

Class LawController

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Controls law enforcement intensity and automatic activity systems. Manages the automatic evaluation of checkpoints, patrols, and other law enforcement activities.

```
public static class LawController
```

Inheritance

[object](#) ← LawController

Inherited Members

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

Remarks

Law enforcement intensity (1-10) determines which automatic activities are enabled. The automatic evaluation system runs every in-game minute based on intensity, time, day, and curfew status. To prevent automatic checkpoint activation, set intensity to 1-4.

Fields

DailyIntensityDrain

The amount of internal intensity that increases per day naturally. Note: This constant exists in the game code but the actual increase is controlled by IntensityIncreasePerDay.

```
public const float DailyIntensityDrain = 0.05
```

Field Value

[float](#)

MaxIntensity

The maximum law enforcement intensity level.

```
public const int MaxIntensity = 10
```

Field Value

[int ↗](#)

MinIntensity

The minimum law enforcement intensity level.

```
public const int MinIntensity = 1
```

Field Value

[int ↗](#)

Properties

Intensity

Gets the current law enforcement intensity level (1-10).

```
public static int Intensity { get; }
```

Property Value

[int ↗](#)

Remarks

Higher values trigger more aggressive automatic law enforcement activities. Checkpoint activation typically requires intensity level 5 or higher.

InternalIntensity

Gets the internal law enforcement intensity as a normalized value (0.0-1.0).

```
public static float InternalIntensity { get; }
```

Property Value

[float](#)

Remarks

This is the underlying value used by the game. The public [Intensity](#) property is derived from this value mapped to a 1-10 range.

IsUsingOverrideSettings

Gets whether custom activity settings are currently overriding the default day-based settings.

```
public static bool IsUsingOverrideSettings { get; }
```

Property Value

[bool](#)

Remarks

When true, the game uses [OverrideActivitySettings\(LawActivitySettings\)](#) instead of day-specific settings (Monday, Tuesday, etc.).

Methods

ChangeIntensity(float)

Changes the law enforcement intensity by a specified amount.

```
public static void ChangeIntensity(float change)
```

Parameters

change [float](#)

The amount to change intensity by (can be negative). Value is clamped internally.

Remarks

The internal intensity is stored as a 0.0-1.0 value and mapped to 1-10 for display. Small changes (e.g., 0.1) will have a noticeable effect.

ClearActivitySettingsOverride()

Clears any activity settings override and returns to using day-based settings.

```
public static void ClearActivitySettingsOverride()
```

OverrideActivitySettings(LawActivitySettings)

Overrides the default day-based activity settings with custom settings.

```
public static void OverrideActivitySettings(LawActivitySettings settings)
```

Parameters

settings LawActivitySettings

The custom activity settings to use, or null to clear the override.

Remarks

WARNING: Advanced feature. The settings object is not part of the public API and must be obtained from game internals. To simply disable automatic activities, set [Intensity](#) to 1 instead.

SetIntensityLevel(int)

Sets the law enforcement intensity to a specific level (1-10).

```
public static void SetIntensityLevel(int level)
```

Parameters

level int ↗

The intensity level to set (1-10). Will be clamped.

Remarks

This is a convenience method that converts a 1-10 level to the internal 0.0-1.0 range. To prevent automatic checkpoint activation, use level 1-4.

SetInternalIntensity(float)

Sets the law enforcement intensity to a specific normalized value (0.0-1.0).

```
public static void SetInternalIntensity(float intensity)
```

Parameters

intensity float ↗

The intensity value to set (0.0 = minimum, 1.0 = maximum). Will be clamped.

Remarks

This sets the internal intensity directly. The public [Intensity](#) property will reflect the change as a 1-10 value.

Class LawManager

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Provides access to law enforcement and police dispatch functionality. Manages police responses, patrol operations, and wanted levels.

```
public static class LawManager
```

Inheritance

[object](#) ← LawManager

Inherited Members

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

Properties

ActiveOfficerCount

Gets the total number of active police officers currently in the game world.

```
public static int ActiveOfficerCount { get; }
```

Property Value

[int](#)

DispatchOfficerCount

The default number of officers dispatched when police are called.

```
public static int DispatchOfficerCount { get; }
```

Property Value

[int](#)

DispatchVehicleUseThreshold

The distance threshold (in units) above which dispatched officers will use a vehicle. Below this threshold, officers will respond on foot.

```
public static float DispatchVehicleUseThreshold { get; }
```

Property Value

[float](#)

EscalationTimeArresting

Time in seconds before pursuit escalates from Arresting to NonLethal if the player remains visible to police.

```
public static float EscalationTimeArresting { get; }
```

Property Value

[float](#)

EscalationTimeNonLethal

Time in seconds before pursuit escalates from NonLethal to Lethal if the player remains visible to police.

```
public static float EscalationTimeNonLethal { get; }
```

Property Value

[float](#) ↗

SearchTimeArresting

Search time in seconds for Arresting pursuit level. Police will search for this long after losing sight of the player.

```
public static float SearchTimeArresting { get; }
```

Property Value

[float](#) ↗

SearchTimeInvestigating

Search time in seconds for Investigating pursuit level. Police will search for this long after losing sight of the player.

```
public static float SearchTimeInvestigating { get; }
```

Property Value

[float](#) ↗

SearchTimeLethal

Search time in seconds for Lethal pursuit level. Police will search for this long after losing sight of the player.

```
public static float SearchTimeLethal { get; }
```

Property Value

[float](#) ↗

SearchTimeNonLethal

Search time in seconds for NonLethal pursuit level. Police will search for this long after losing sight of the player.

```
public static float SearchTimeNonLethal { get; }
```

Property Value

[float](#) ↗

Methods

CallPolice(Player)

Dispatches police officers to pursue the specified player for a crime. Officers will be sent from the nearest police station.

```
public static void CallPolice(Player target)
```

Parameters

target [Player](#)

The player who committed the crime and will be pursued.

Remarks

This method will not dispatch police during tutorial mode. The number of officers dispatched is determined by [DispatchOfficerCount](#). Officers will use vehicles if the distance exceeds [DispatchVehicleUseThreshold](#).

ClearWantedLevel(Player)

Clears the wanted level for the specified player, ending any active pursuit.

```
public static void ClearWantedLevel(Player target)
```

Parameters

target [Player](#)

The player to clear the wanted level for.

Remarks

This sets the pursuit level to None and clears all crimes from the player's record.

DeescalateWantedLevel(Player)

Decreases the wanted level for the specified player by one level.

```
public static void DeescalateWantedLevel(Player target)
```

Parameters

target [Player](#)

The player to de-escalate the wanted level for.

Remarks

Progression: Lethal → NonLethal → Arresting → Investigating → None. This is a convenience method that calls [Deescalate\(\)](#).

EscalateWantedLevel(Player)

Increases the wanted level for the specified player by one level.

```
public static void EscalateWantedLevel(Player target)
```

Parameters

target [Player](#)

The player to escalate the wanted level for.

Remarks

Progression: None → Investigating → Arresting → NonLethal → Lethal. This is a convenience method that calls [Escalate\(\)](#).

FindFootPatrolRoute(string)

Finds a foot patrol route in the scene by name.

```
public static FootPatrolRoute FindFootPatrolRoute(string routeName)
```

Parameters

routeName [string](#)

The name of the patrol route to find.

Returns

[FootPatrolRoute](#)

The FootPatrolRoute wrapper, or null if not found.

FindVehiclePatrolRoute(string)

Finds a vehicle patrol route in the scene by name.

```
public static VehiclePatrolRoute FindVehiclePatrolRoute(string routeName)
```

Parameters

routeName [string](#)

The name of the patrol route to find.

Returns

[VehiclePatrolRoute](#)

The VehiclePatrolRoute wrapper, or null if not found.

GetAllFootPatrolRoutes()

Gets all foot patrol routes currently in the scene.

```
public static FootPatrolRoute[] GetAllFootPatrolRoutes()
```

Returns

[FootPatrolRoute](#)[]

An array of FootPatrolRoute wrappers.

GetAllVehiclePatrolRoutes()

Gets all vehicle patrol routes currently in the scene.

```
public static VehiclePatrolRoute[] GetAllVehiclePatrolRoutes()
```

Returns

[VehiclePatrolRoute\[\]](#)

An array of VehiclePatrolRoute wrappers.

GetWantedLevel(Player)

Gets the current wanted level (pursuit level) for the specified player.

```
public static PursuitLevel GetWantedLevel(Player target)
```

Parameters

[target Player](#)

The player to get the wanted level for.

Returns

[PursuitLevel](#)

The player's current pursuit level.

IsLethalForceAuthorized(Player)

Checks if police are authorized to use lethal force against the player.

```
public static bool IsLethalForceAuthorized(Player target)
```

Parameters

target [Player](#)

The player to check.

Returns

[bool](#) ↗

True if the player's wanted level is at Lethal.

IsPlayerWanted(Player)

Checks if the player is currently in an active police pursuit.

```
public static bool IsPlayerWanted(Player target)
```

Parameters

target [Player](#)

The player to check.

Returns

[bool](#) ↗

True if the player has any wanted level above None.

IsUnderInvestigation(Player)

Checks if the player is currently being investigated by police. This includes all pursuit levels except None.

```
public static bool IsUnderInvestigation(Player target)
```

Parameters

target [Player](#)

The player to check.

Returns

[bool](#) ↗

True if police are actively pursuing or investigating the player.

SetWantedLevel(Player, PursuitLevel)

Sets the wanted level (pursuit level) for the specified player.

```
public static void SetWantedLevel(Player target, PursuitLevel level)
```

Parameters

target [Player](#)

The player to set the wanted level for.

level [PursuitLevel](#)

The pursuit level to set.

Remarks

Setting to [None](#) will clear all crimes and end the pursuit. This is a convenience method that calls [SetPursuitLevel\(PursuitLevel\)](#).

StartFootPatrol(FootPatrolRoute, int)

Starts a foot patrol using the specified route and number of officers. Officers are pulled from the nearest police station to the route's starting point.

```
public static PatrolGroup StartFootPatrol(FootPatrolRoute route, int  
requestedMembers = 2)
```

Parameters

route [FootPatrolRoute](#)

The foot patrol route to use.

requestedMembers [int](#)

The number of officers to assign to the patrol.

Returns

[PatrolGroup](#)

A PatrolGroup object representing the active patrol, or null if insufficient officers are available.

Remarks

If insufficient officers are available at the nearest police station, the patrol will not be created. Use [FindFootPatrolRoute\(string\)](#) to locate existing patrol routes in the scene.

StartVehiclePatrol(VehiclePatrolRoute)

Starts a vehicle patrol using the specified route. One officer is pulled from the nearest police station and assigned a patrol vehicle.

```
public static bool StartVehiclePatrol(VehiclePatrolRoute route)
```

Parameters

route [VehiclePatrolRoute](#)

The vehicle patrol route to use.

Returns

bool ↗

True if the patrol was started successfully, false otherwise.

Remarks

If no officers are available at the nearest police station, the patrol will not be created. Use [FindVehiclePatrolRoute\(string\)](#) to locate existing patrol routes in the scene. The assigned officer will automatically patrol the route with a police vehicle.

Class PatrolGroup

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Represents a group of police officers on foot patrol. Manages patrol group movement, waypoints, and member coordination.

```
public sealed class PatrolGroup
```

Inheritance

[object](#) ← PatrolGroup

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

CurrentWaypoint

Gets the current waypoint index the patrol group is moving toward.

```
public int CurrentWaypoint { get; }
```

Property Value

[int](#)

MemberCount

Gets the number of officers in this patrol group.

```
public int MemberCount { get; }
```

Property Value

[int ↗](#)

Route

Gets the patrol route this group is following.

```
public FootPatrolRoute Route { get; }
```

Property Value

[FootPatrolRoute](#)

Methods

AdvanceGroup()

Advances the patrol group to the next waypoint.

```
public void AdvanceGroup()
```

DisbandGroup()

Disbands the patrol group, releasing all officers back to their normal duties.

```
public void DisbandGroup()
```

IsGroupReadyToAdvance()

Checks if all members of the patrol group are ready to advance to the next waypoint.

```
public bool IsGroupReadyToAdvance()
```

Returns

[bool](#)

True if all members are ready to advance.

IsPaused()

Checks if the patrol group is currently paused.

```
public bool IsPaused()
```

Returns

[bool](#)

True if the patrol is paused.

Class PlayerCrimeData

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Manages a player's criminal record, wanted level, and police pursuit state.

```
public sealed class PlayerCrimeData
```

Inheritance

[object](#) ← PlayerCrimeData

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

BodySearchPending

Whether the player is subject to a body search by police.

```
public bool BodySearchPending { get; }
```

Property Value

[bool](#)

CurrentPursuitLevel

The current pursuit level (wanted level) of the player.

```
public PursuitLevel CurrentPursuitLevel { get; }
```

Property Value

[PursuitLevel](#)

EvadedArrest

Whether the player has evaded arrest (escalated from Arresting to NonLethal). This flag affects the severity of charges.

```
public bool EvadedArrest { get; }
```

Property Value

[bool](#)

LastKnownPosition

The player's last known position to police. This is where officers will search if they lose sight of the player.

```
public Vector3 LastKnownPosition { get; }
```

Property Value

[Vector3](#)

TimeSinceSighted

Time in seconds since police last saw the player. When this exceeds the search time for the current pursuit level, the pursuit ends.

```
public float TimeSinceSighted { get; }
```

Property Value

[float](#) ↗

Methods

ClearCrimes()

Clears all crimes from the player's record.

```
public void ClearCrimes()
```

Remarks

This does not automatically end the pursuit or change the pursuit level. To fully clear wanted status, use [SetPursuitLevel\(PursuitLevel\)](#) with [None](#).

Deescalate()

De-escalates the pursuit level to the next lower level.

```
public void Deescalate()
```

Remarks

Progression: Lethal → NonLethal → Arresting → Investigating → None. De-escalating to None will clear all crimes.

Escalate()

Escalates the pursuit level to the next higher level.

```
public void Escalate()
```

Remarks

Progression: None → Investigating → Arresting → NonLethal → Lethal. Escalating from Arresting to NonLethal marks the player as having evaded arrest.

GetSearchTime()

How long police will search for the player after losing sight, based on current pursuit level.

```
public float GetSearchTime()
```

Returns

float ↗

Search time in seconds.

RecordLastKnownPosition(bool)

Records the player's current position as their last known position to police.

```
public void RecordLastKnownPosition(bool resetTimeSinceSighted = true)
```

Parameters

resetTimeSinceSighted bool ↗

If true, resets the time since sighted to 0 (police just saw the player).

SetPursuitLevel(PursuitLevel)

Sets the player's pursuit level (wanted level).

```
public void SetPursuitLevel(PursuitLevel level)
```

Parameters

level [PursuitLevel](#)

The pursuit level to set.

Remarks

Setting to None will clear all crimes and end the pursuit. This method will not work during tutorial mode.

Enum PursuitLevel

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Represents the intensity level of a police pursuit. Higher levels indicate more aggressive police response.

```
public enum PursuitLevel
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Arresting = 2

Police are attempting to arrest the player. Officers will attempt non-lethal takedown. Search time: 25 seconds after losing sight. Escalates to NonLethal after 25 seconds if player remains visible.

Investigating = 1

Police are investigating the player's location. Officers will search for the player but won't use force. Search time: 60 seconds after losing sight.

Lethal = 4

Police are authorized to use lethal force. Maximum pursuit intensity - officers will shoot on sight. Search time: 40 seconds after losing sight. Does not escalate further.

NonLethal = 3

Police are authorized to use non-lethal force (tasers, batons). Officers will be more aggressive in pursuit. Search time: 30 seconds after losing sight. Escalates to Lethal after 120 seconds if player remains visible.

None = 0

No active pursuit. Player is not wanted.

Class VehiclePatrolRoute

Namespace: [S1API.Law](#)

Assembly: S1API.dll

Represents a vehicle patrol route with waypoints for police vehicles to follow.

```
public sealed class VehiclePatrolRoute
```

Inheritance

[object](#) ← VehiclePatrolRoute

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Position

Gets the position of the route object itself.

```
public Vector3 Position { get; }
```

Property Value

Vector3

RouteName

Gets the name of this patrol route.

```
public string RouteName { get; }
```

Property Value

[string](#)

StartWaypointIndex

Gets the starting waypoint index for this route.

```
public int StartWaypointIndex { get; }
```

Property Value

[int](#)

WaypointCount

Gets the number of waypoints in this route.

```
public int WaypointCount { get; }
```

Property Value

[int](#)

Methods

GetWaypointPosition(int)

Gets the position of a waypoint at the specified index.

```
public Vector3 GetWaypointPosition(int index)
```

Parameters

index [int](#)

The waypoint index.

Returns

Vector3

The waypoint position, or Vector3.zero if invalid index.

Namespace S1API.Leveling

Classes

[LevelManager](#)

Exposes the player progression system, including ranks, tiers, XP, unlockables, and related events.

[Unlockable](#)

Represents an entry that becomes available when a rank threshold is met.

Structs

[FullRank](#)

Represents a rank/tier combination.

Enums

[Rank](#)

Represents all ranks available in the game.

Struct FullRank

Namespace: [S1API.Leveling](#)

Assembly: S1API.dll

Represents a rank/tier combination.

```
public readonly struct FullRank : IEquatable<FullRank>, IComparable<FullRank>
```

Implements

[IEquatable](#)<[FullRank](#)>, [IComparable](#)<[FullRank](#)>

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

FullRank(Rank, int)

Creates a rank/tier combination.

```
public FullRank(Rank rank, int tier)
```

Parameters

rank [Rank](#)

Player rank.

tier [int](#)

Tier within the rank (1-5 for most ranks).

Properties

Rank

Rank component.

```
public Rank Rank { get; }
```

Property Value

[Rank](#)

Tier

Tier component (1-5 for most ranks, unlimited for Kingpin).

```
public int Tier { get; }
```

Property Value

[int](#)

Methods

CompareTo(FullRank)

```
public int CompareTo(FullRank other)
```

Parameters

other [FullRank](#)

Returns

[int](#)

Equals(FullRank)

```
public bool Equals(FullRank other)
```

Parameters

[other](#) [FullRank](#)

Returns

[bool](#)

Equals(object?)

```
public override bool Equals(object? obj)
```

Parameters

[obj](#) [object](#)

Returns

[bool](#)

GetHashCode()

```
public override int GetHashCode()
```

Returns

[int ↗](#)

GetRankIndex()

Returns an index useful for UI progress (0..N).

```
public int GetRankIndex()
```

Returns

[int ↗](#)

NextRank()

Returns the next rank/tier (rolls into the next rank after tier 5).

```
public FullRank NextRank()
```

Returns

[FullRank](#)

ToFloat()

Converts the rank to a float for interpolation logic.

```
public float ToFloat()
```

Returns

[float](#)

ToString()

```
public override string ToString()
```

Returns

[string](#)

Operators

operator ==(FullRank, FullRank)

Equality check.

```
public static bool operator ==(FullRank a, FullRank b)
```

Parameters

a [FullRank](#)

b [FullRank](#)

Returns

[bool](#)

operator >(FullRank, FullRank)

Greater-than comparison helper.

```
public static bool operator >(FullRank a, FullRank b)
```

Parameters

a [FullRank](#)

b [FullRank](#)

Returns

bool 

operator >=(FullRank, FullRank)

Greater-or-equal comparison helper.

```
public static bool operator >=(FullRank a, FullRank b)
```

Parameters

a [FullRank](#)

b [FullRank](#)

Returns

bool 

operator !=(FullRank, FullRank)

Inequality check.

```
public static bool operator !=(FullRank a, FullRank b)
```

Parameters

a [FullRank](#)

b [FullRank](#)

Returns

[bool](#) ↗

operator <(FullRank, FullRank)

Less-than comparison helper.

```
public static bool operator <(FullRank a, FullRank b)
```

Parameters

a [FullRank](#)

b [FullRank](#)

Returns

[bool](#) ↗

operator <=(FullRank, FullRank)

Less-or-equal comparison helper.

```
public static bool operator <=(FullRank a, FullRank b)
```

Parameters

a [FullRank](#)

b [FullRank](#)

Returns

[bool](#) ↗

Class LevelManager

Namespace: [S1API.Leveling](#)

Assembly: S1API.dll

Exposes the player progression system, including ranks, tiers, XP, unlockables, and related events.

```
public static class LevelManager
```

Inheritance

[object](#) ← LevelManager

Inherited Members

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

Properties

CurrentRank

The player's current rank and tier combined.

```
public static FullRank CurrentRank { get; }
```

Property Value

[FullRank](#)

Exists

Returns true when the underlying levelling manager has been instantiated.

```
public static bool Exists { get; }
```

Property Value

[bool](#)

Rank

The player's current rank.

```
public static Rank Rank { get; }
```

Property Value

[Rank](#)

Tier

The player's current tier within their rank.

```
public static int Tier { get; }
```

Property Value

[int](#)

TotalXP

Total XP accumulated across all ranks.

```
public static int TotalXP { get; }
```

Property Value

[int](#)

XP

XP progress within the current tier.

```
public static int XP { get; }
```

Property Value

[int](#)

XPToNextTier

XP required to reach the next tier threshold.

```
public static float XPToNextTier { get; }
```

Property Value

[float](#)

Methods

AddUnlockable(Unlockable)

Registers an unlockable item for the specified rank.

```
public static void AddUnlockable(Unlockable unlockable)
```

Parameters

[unlockable](#) [Unlockable](#)

AddXP(int)

Adds XP to the player. Only works when invoked from the host/server.

```
public static void AddXP(int amount)
```

Parameters

[amount](#) [int](#)

How much XP to award.

GetFullRankForXP(int)

Converts a target XP value to the associated rank and tier.

```
public static FullRank GetFullRankForXP(int totalXp)
```

Parameters

[totalXp](#) [int](#)

Returns

[FullRank](#)

GetOrderLimitMultiplier(FullRank)

Returns the order limit multiplier used by customer systems for the specified rank.

```
public static float GetOrderLimitMultiplier(FullRank rank)
```

Parameters

rank [FullRank](#)

Returns

[float](#) ↗

GetTotalXPForRank(FullRank)

Gets the total XP required to reach a specific rank/tier combination.

```
public static int GetTotalXPForRank(FullRank rank)
```

Parameters

rank [FullRank](#)

Returns

[int](#) ↗

GetUnlockables(FullRank)

Enumerates unlockables for the provided rank.

```
public static IEnumerable<Unlockable> GetUnlockables(FullRank rank)
```

Parameters

rank [FullRank](#)

Returns

[IEnumerable](#)<[Unlockable](#)>

GetXPForTier(Rank)

Gets the XP required to complete a tier for the specified rank.

```
public static int GetXPForTier(Rank rank)
```

Parameters

rank [Rank](#)

Returns

[int](#)

Events

OnRankUp

Raised when the player's rank or tier actually increases (tier increases within a rank, or rank increases when tier exceeds 5). This only fires when the rank/tier value changes, not on every XP update. Provides the previous and new rank values.

```
public static event Action<FullRank, FullRank>? OnRankUp
```

Event Type

[Action](#)<[FullRank](#), [FullRank](#)>

OnXPChanged

Raised whenever XP or rank/tier data is updated, even if the rank/tier values haven't changed. This fires on every XP change, including when only XP increases without a rank/tier change. Provides the previous and new rank values (which may be the same).

```
public static event Action<FullRank, FullRank>? OnXPChanged
```

Event Type

[Action](#)<[FullRank](#), [FullRank](#)>

Enum Rank

Namespace: [S1API.Leveling](#)

Assembly: S1API.dll

Represents all ranks available in the game.

```
public enum Rank
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Bagman = 4

Represents the fifth rank for the player.

Baron = 9

Represents the tenth rank for the player.

BlockBoss = 7

Represents the eighth rank for the player.

Enforcer = 5

Represents the sixth rank for the player.

Hoodlum = 1

Represents the second rank for the player.

Hustler = 3

Represents the fourth rank for the player.

Kingpin = 10

Represents the eleventh rank for the player.

Peddler = 2

Represents the third rank for the player.

ShotCaller = 6

Represents the seventh rank for the player.

StreetRat = 0

Represents the first rank for the player.

Underlord = 8

Represents the ninth rank for the player.

Class Unlockable

Namespace: [S1API.Leveling](#)

Assembly: S1API.dll

Represents an entry that becomes available when a rank threshold is met.

```
public sealed class Unlockable
```

Inheritance

[object](#) ← Unlockable

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Unlockable(FullRank, string, Sprite)

Creates an unlockable tied to a specific rank and tier.

```
public Unlockable(FullRank rank, string title, Sprite icon)
```

Parameters

rank [FullRank](#)

Rank+tier requirement.

title [string](#)

Display title shown in UI.

icon Sprite

Display icon.

Properties

Icon

Icon associated with the unlockable entry.

```
public Sprite Icon { get; set; }
```

Property Value

Sprite

Rank

Rank requirement for this unlockable.

```
public FullRank Rank { get; set; }
```

Property Value

[FullRank](#)

Title

Display title used in UI elements.

```
public string Title { get; set; }
```

Property Value

[string](#) 

Namespace S1API.Lifecycle

Classes

[GameLifecycle](#)

Provides lifecycle events for game initialization and loading. Subscribe to these events to execute code at specific points in the game's lifecycle.

Class GameLifecycle

Namespace: [S1API.Lifecycle](#)

Assembly: S1API.dll

Provides lifecycle events for game initialization and loading. Subscribe to these events to execute code at specific points in the game's lifecycle.

```
public static class GameLifecycle
```

Inheritance

[object](#) ← GameLifecycle

Inherited Members

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

Examples

```
// Subscribe to PreLoad event to initialize items before game loads
GameLifecycle.OnPreLoad += () => {
    MelonLogger.Msg("Creating custom items...");
    CreateMyCustomItems();
};
```

Remarks

This API provides a cross-runtime abstraction over ScheduleOne's LoadManager events, eliminating the need for runtime-specific conditional compilation in mods.

Events

OnPreLoad

Fired before the game begins loading saved data. This is the ideal time to register custom items, as they need to exist before save data is loaded.

```
public static event Action OnPreLoad
```

Event Type

[Action](#) ↗

Remarks

Equivalent to LoadManager.onPreLoad but abstracted for cross-runtime compatibility.

Namespace S1API.Logging

Classes

[Log](#)

Centralized Logging class that handles both BepInEx and MelonLoader logging.

Class Log

Namespace: [S1API.Logging](#)

Assembly: S1API.dll

Centralized Logging class that handles both BepInEx and MelonLoader logging.

```
public class Log
```

Inheritance

[object](#) ← Log

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Log(string)

Default constructor for [Log](#) instance

```
public Log(string sourceName)
```

Parameters

sourceName [string](#)

The source name to use for logging

Methods

BigError(string)

Logs a message with Fatal level

```
public void BigError(string message)
```

Parameters

message string

Message to log

Error(string)

Logs a message with Error level

```
public void Error(string message)
```

Parameters

message string

Message to log

Msg(string)

Logs a message with Info level

```
public void Msg(string message)
```

Parameters

message string

Message to log

Warning(string)

Logs a message with Warning level

```
public void Warning(string message)
```

Parameters

message [string](#)

Message to log

Namespace S1API.Map

Namespaces

[S1API.Map.Buildings](#)

[S1API.Map.DeliveryLocations](#)

[S1API.Map.ParkingLots](#)

Classes

[Building](#)

Modder-facing wrapper for an enterable building in the world. Provides name-based lookup and basic metadata without exposing game types.

[DeliveryLocation](#)

Wrapper for a base-game delivery location.

[ParkingData](#)

Defines where to park

[ParkingLotNameAttribute](#)

Attribute to mark a parking lot identifier class with its GameObject name.

[ParkingLotRegistry](#)

Registry and utilities for parking-related queries.

[ParkingLotWrapper](#)

Modder-facing wrapper for a parking lot.

[ParkingSpotWrapper](#)

Modder-facing wrapper for a parking spot within a lot.

Interfaces

[IParkingLotIdentifier](#)

Marker interface for parking lot identifier classes. Use this with ParkingLotNameAttribute to create type-safe parking lot lookups.

Enums

[Region](#)

The regions available in the game.

Namespace S1API.MapBuildings

Classes

[ApartmentBuilding](#)

Identifier for the base-game building named "Apartment Building". Modders can use [Get<T>\(\)](#) to resolve it.

[Arcade](#)

Identifier for the base-game building named "Arcade". Modders can use [Get<T>\(\)](#) to resolve it.

[BenjisMotelRoom](#)

Identifier for the base-game building named "Benji's Motel Room". Modders can use [Get<T>\(\)](#) to resolve it.

[BethsRoom](#)

Identifier for the base-game building named "Beth's Room". Modders can use [Get<T>\(\)](#) to resolve it.

[BoutiqueStore](#)

Identifier for the base-game building named "Boutique Store". Modders can use [Get<T>\(\)](#) to resolve it.

[BradsTent](#)

Identifier for the base-game building named "Brad's Tent". Modders can use [Get<T>\(\)](#) to resolve it.

[BudsBar](#)

Identifier for the base-game building named "Bud's Bar". Modders can use [Get<T>\(\)](#) to resolve it.

[BuildingNameAttribute](#)

Annotate building identifier types with the display name of the building. Usage: [BuildingName("North apartments")]

[Cafe](#)

Identifier for the base-game building named "Cafe". Modders can use [Get<T>\(\)](#) to resolve it.

[Caravan](#)

Identifier for the base-game building named "Caravan". Modders can use [Get<T>\(\)](#) to resolve it.

[CarlsHouse](#)

Identifier for the base-game building named "Carl's House". Modders can use [Get<T>\(\)](#) to resolve it.

[Casino](#)

Identifier for the base-game building named "Casino". Modders can use [Get<T>\(\)](#) to resolve it.

[CharlesHouse](#)

Identifier for the base-game building named "Charles' House". Modders can use [Get<T>\(\)](#) to resolve it.

[ChemicalPlantA](#)

Identifier for the base-game building named "Chemical Plant A". Modders can use [Get<T>\(\)](#) to resolve it.

[ChineseRestaurant](#)

Identifier for the base-game building named "Chinese Restaurant". Modders can use [Get<T>\(\)](#) to resolve it.

[Church](#)

Identifier for the base-game building named "Church". Modders can use [Get<T>\(\)](#) to resolve it.

[CommunityCenter](#)

Identifier for the base-game building named "Community Center". Modders can use [Get<T>\(\)](#) to resolve it.

[CornerStore](#)

Identifier for the base-game building named "Corner Store". Modders can use [Get<T>\(\)](#) to resolve it.

[Courthouse](#)

Identifier for the base-game building named "Courthouse". Modders can use [Get<T>\(\)](#) to resolve it.

[DansHardwareUpstairs](#)

Identifier for the base-game building named "Dan's Hardware Upstairs". Modders can use [Get<T>\(\)](#) to resolve it.

[DocksIndustrialBuilding](#)

Identifier for the base-game building named "Docks Industrial Building". Modders can use [Get<T>\(\)](#) to resolve it.

[DocksShippingContainer](#)

Identifier for the base-game building named "Docks Shipping Container". Modders can use [Get<T>\(\)](#) to resolve it.

[FireStation](#)

Identifier for the base-game building named "Fire station". Modders can use [Get<T>\(\)](#) to resolve it.

[FishWarehouse](#)

Identifier for the base-game building named "Fish Warehouse". Modders can use [Get<T>\(\)](#) to resolve it.

[GeorgeAndMollysHouse](#)

Identifier for the base-game building named "George and Molly's House". Modders can use [Get<T>\(\)](#) to resolve it.

[GoblinHideBuilding](#)

Identifier for the base-game building named "Goblin Hide Building". Modders can use [Get<T>\(\)](#) to resolve it.

[HAMLegal](#)

Identifier for the base-game building named "HAM Legal". Modders can use [Get<T>\(\)](#) to resolve it.

[HoltHouse](#)

Identifier for the base-game building named "Holt House". Modders can use [Get<T>\(\)](#) to resolve it.

[HylandBank](#)

Identifier for the base-game building named "Hyland Bank". Modders can use [Get<T>\(\)](#) to resolve it.

[HylandMedical](#)

Identifier for the base-game building named "Hyland Medical". Modders can use [Get<T>\(\)](#) to resolve it.

[JanesCaravan](#)

Identifier for the base-game building named "Jane's Caravan". Modders can use [Get<T>\(\)](#) to resolve it.

[JerrysTent](#)

Identifier for the base-game building named "Jerry's Tent". Modders can use [Get<T>\(\)](#) to resolve it.

[JessisRoom](#)

Identifier for the base-game building named "Jessi's Room". Modders can use [Get<T>\(\)](#) to resolve it.

[KennedyHouse](#)

Identifier for the base-game building named "Kennedy House". Modders can use [Get<T>\(\)](#) to resolve it.

[KnightHouse](#)

Identifier for the base-game building named "Knight House". Modders can use [Get<T>\(\)](#) to resolve it.

[KyleAndAustinsHouse](#)

Identifier for the base-game building named "Kyle and Austin's House". Modders can use [Get<T>\(\)](#) to resolve it.

[LeosShippingContainer](#)

Identifier for the base-game building named "Leos' Shipping Container". Modders can use [Get<T>\(\)](#) to resolve it.

[LesOrdurePuantes](#)

Identifier for the base-game building named "Les Ordure Puantes". Modders can use [Get<T>\(\)](#) to resolve it.

[MayorsHouse](#)

Identifier for the base-game building named "Mayor's House". Modders can use [Get<T>\(\)](#) to resolve it.

[MicksHouse](#)

Identifier for the base-game building named "Mick's House". Modders can use [Get<T>\(\)](#) to resolve it.

[ModernMansion](#)

Identifier for the base-game building named "Modern Mansion". Modders can use [Get<T>\(\)](#) to resolve it.

[MotelOffice](#)

Identifier for the base-game building named "Motel Office". Modders can use [Get<T>\(\)](#) to resolve it.

[Nightclub](#)

Identifier for the base-game building named "Nightclub". Modders can use [Get<T>\(\)](#) to resolve it.

[NorthApartments](#)

Identifier for the base-game building named "North apartments". Modders can use [Get<T>\(\)](#) to resolve it.

[NorthIndustrialBuilding](#)

Identifier for the base-game building named "North Industrial Building". Modders can use [Get<T>\(\)](#) to resolve it.

[NorthWarehouse](#)

Identifier for the base-game building named "North Warehouse". Modders can use [Get<T>\(\)](#) to resolve it.

[OverpassBuilding](#)

Identifier for the base-game building named "Overpass Building". Modders can use [Get<T>\(\)](#) to resolve it.

[PawnShop](#)

Identifier for the base-game building named "Pawn Shop". Modders can use [Get<T>\(\)](#) to resolve it.

[PetersRoom](#)

Identifier for the base-game building named "Peter's Room". Modders can use [Get<T>\(\)](#) to resolve it.

[Pillville](#)

Identifier for the base-game building named "Pillville". Modders can use [Get<T>\(\)](#) to resolve it.

[PoliceStation](#)

Identifier for the base-game building named "Police Station". Modders can use [Get<T>\(\)](#) to resolve it.

[RandysBaitTackle](#)

Identifier for the base-game building named "Randy's Bait & Tackle". Modders can use [Get<T>\(\)](#) to resolve it.

[Room1](#)

Identifier for the base-game building named "Room 1". Modders can use [Get<T>\(\)](#) to resolve it.

Room2

Identifier for the base-game building named "Room 2". Modders can use [Get<T>\(\)](#) to resolve it.

Room3

Identifier for the base-game building named "Room 3". Modders can use [Get<T>\(\)](#) to resolve it.

SauerkrautSupreme

Identifier for the base-game building named "Sauerkraut Supreme". Modders can use [Get<T>\(\)](#) to resolve it.

Shack

Identifier for the base-game building named "Shack". Modders can use [Get<T>\(\)](#) to resolve it.

ShermanHouse

Identifier for the base-game building named "Sherman House". Modders can use [Get<T>\(\)](#) to resolve it.

ShootingRange

Identifier for the base-game building named "Shooting Range". Modders can use [Get<T>\(\)](#) to resolve it.

SlopShop

Identifier for the base-game building named "Slop Shop". Modders can use [Get<T>\(\)](#) to resolve it.

SouthOverpassBuilding

Identifier for the base-game building named "South Overpass Building". Modders can use [Get<T>\(\)](#) to resolve it.

StevensonHouse

Identifier for the base-game building named "Stevenson House". Modders can use [Get<T>\(\)](#) to resolve it.

StorageUnit2

Identifier for the base-game building named "Storage Unit #2". Modders can use [Get<T>\(\)](#) to resolve it.

StorageUnit3

Identifier for the base-game building named "Storage Unit #3". Modders can use [Get<T>\(\)](#) to resolve it.

Supermarket

Identifier for the base-game building named "Supermarket". Modders can use [Get<T>\(\)](#) to resolve it.

TallTower

Identifier for the base-game building named "Tall Tower". Modders can use [Get<T>\(\)](#) to resolve it.

TheCrimsonCanary

Identifier for the base-game building named "The Crimson Canary". Modders can use [Get<T>\(\)](#) to resolve it.

[ThePissHut](#)

Identifier for the base-game building named "The Piss Hut". Modders can use [Get<T>\(\)](#) to resolve it.

[ThompsonConstructionDemo](#)

Identifier for the base-game building named "Thompson Construction & Demo". Modders can use [Get<T>\(\)](#) to resolve it.

[ThompsonHouse](#)

Identifier for the base-game building named "Thompson House". Modders can use [Get<T>\(\)](#) to resolve it.

[TownHall](#)

Identifier for the base-game building named "Town hall". Modders can use [Get<T>\(\)](#) to resolve it.

[UpscaleApartments](#)

Identifier for the base-game building named "Upscale Apartments". Modders can use [Get<T>\(\)](#) to resolve it.

[WebsterHouse](#)

Identifier for the base-game building named "Webster House". Modders can use [Get<T>\(\)](#) to resolve it.

[WeisCabin](#)

Identifier for the base-game building named "Wei's Cabin". Modders can use [Get<T>\(\)](#) to resolve it.

[WestGasMart](#)

Identifier for the base-game building named "West Gas-Mart". Modders can use [Get<T>\(\)](#) to resolve it.

[WilkinsonHouse](#)

Identifier for the base-game building named "Wilkinson House". Modders can use [Get<T>\(\)](#) to resolve it.

Interfaces

[IBuildingIdentifier](#)

Marker interface for building identifier types used with Building.Get<T>(). Implement empty classes like 'public sealed class NorthApartments : IBuildingIdentifier {}' and optionally annotate with [BuildingName("North apartments")].

Class ApartmentBuilding

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Apartment Buiding". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Apartment Buiding")]
public sealed class ApartmentBuilding : IBuildingIdentifier
```

Inheritance

[object](#) ← ApartmentBuilding

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ApartmentBuilding()

```
public ApartmentBuilding()
```

Class Arcade

Namespace: [S1API.Map.Buildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Arcade". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Arcade")]
public sealed class Arcade : IBuildingIdentifier
```

Inheritance

[object](#) ← Arcade

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Arcade()

```
public Arcade()
```

Class BenjisMotelRoom

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Benji's Motel Room". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Benji's Motel Room")]
public sealed class BenjisMotelRoom : IBuildingIdentifier
```

Inheritance

[object](#) ← BenjisMotelRoom

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BenjisMotelRoom()

```
public BenjisMotelRoom()
```

Class BethsRoom

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Beth's Room". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Beth's Room")]
public sealed class BethsRoom : IBuildingIdentifier
```

Inheritance

[object](#) ← BethsRoom

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BethsRoom()

```
public BethsRoom()
```

Class BoutiqueStore

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Boutique Store". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Boutique Store")]
public sealed class BoutiqueStore : IBuildingIdentifier
```

Inheritance

[object](#) ← BoutiqueStore

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BoutiqueStore()

```
public BoutiqueStore()
```

Class BradsTent

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Brad's Tent". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Brad's Tent")]
public sealed class BradsTent : IBuildingIdentifier
```

Inheritance

[object](#) ← BradsTent

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BradsTent()

```
public BradsTent()
```

Class BudsBar

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Bud's Bar". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Bud's Bar")]
public sealed class BudsBar : IBuildingIdentifier
```

Inheritance

[object](#) ← BudsBar

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BudsBar()

```
public BudsBar()
```

Class BuildingNameAttribute

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Annotate building identifier types with the display name of the building. Usage: [BuildingName("North apartments")]

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false, Inherited = false)]
public sealed class BuildingNameAttribute : Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← BuildingNameAttribute

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BuildingNameAttribute(string)

```
public BuildingNameAttribute(string name)
```

Parameters

name [string](#)

Properties

Name

```
public string Name { get; }
```

Property Value

[string](#)

Class Cafe

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Cafe". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Cafe")]
public sealed class Cafe : IBuildingIdentifier
```

Inheritance

[object](#) ← Cafe

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Cafe()

```
public Cafe()
```

Class Caravan

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Caravan". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Caravan")]
public sealed class Caravan : IBuildingIdentifier
```

Inheritance

[object](#) ← Caravan

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Caravan()

```
public Caravan()
```

Class CarlsHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Carl's House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Carl's House")]
public sealed class CarlsHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← CarlsHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

CarlsHouse()

```
public CarlsHouse()
```

Class Casino

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Casino". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Casino")]
public sealed class Casino : IBuildingIdentifier
```

Inheritance

[object](#) ← Casino

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Casino()

```
public Casino()
```

Class CharlesHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Charles' House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Charles' House")]
public sealed class CharlesHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← CharlesHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

CharlesHouse()

```
public CharlesHouse()
```

Class ChemicalPlantA

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Chemical Plant A". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Chemical Plant A")]
public sealed class ChemicalPlantA : IBuildingIdentifier
```

Inheritance

[object](#) ← ChemicalPlantA

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ChemicalPlantA()

```
public ChemicalPlantA()
```

Class ChineseRestaurant

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Chinese Restaurant". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Chinese Restaurant")]
public sealed class ChineseRestaurant : IBuildingIdentifier
```

Inheritance

[object](#) ← ChineseRestaurant

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ChineseRestaurant()

```
public ChineseRestaurant()
```

Class Church

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Church". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Church")]
public sealed class Church : IBuildingIdentifier
```

Inheritance

[object](#) ← Church

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Church()

```
public Church()
```

Class CommunityCenter

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Community Center". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Community Center")]
public sealed class CommunityCenter : IBuildingIdentifier
```

Inheritance

[object](#) ← CommunityCenter

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

CommunityCenter()

```
public CommunityCenter()
```

Class CornerStore

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Corner Store". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Corner Store")]
public sealed class CornerStore : IBuildingIdentifier
```

Inheritance

[object](#) ← CornerStore

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

CornerStore()

```
public CornerStore()
```

Class Courthouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Courthouse". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Courthouse")]
public sealed class Courthouse : IBuildingIdentifier
```

Inheritance

[object](#) ← Courthouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Courthouse()

```
public Courthouse()
```

Class DansHardwareUpstairs

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Dan's Hardware Upstairs". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Dan's Hardware Upstairs")]
public sealed class DansHardwareUpstairs : IBuildingIdentifier
```

Inheritance

[object](#) ← DansHardwareUpstairs

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DansHardwareUpstairs()

```
public DansHardwareUpstairs()
```

Class DocksIndustrialBuilding

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Docks Industrial Building". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Docks Industrial Building")]
public sealed class DocksIndustrialBuilding : IBuildingIdentifier
```

Inheritance

[object](#) ← DocksIndustrialBuilding

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DocksIndustrialBuilding()

```
public DocksIndustrialBuilding()
```

Class DocksShippingContainer

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Docks Shipping Container". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Docks Shipping Container")]
public sealed class DocksShippingContainer : IBuildingIdentifier
```

Inheritance

[object](#) ← DocksShippingContainer

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DocksShippingContainer()

```
public DocksShippingContainer()
```

Class FireStation

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Fire station". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Fire station")]
public sealed class FireStation : IBuildingIdentifier
```

Inheritance

[object](#) ← FireStation

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

FireStation()

```
public FireStation()
```

Class FishWarehouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Fish Warehouse". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Fish Warehouse")]
public sealed class FishWarehouse : IBuildingIdentifier
```

Inheritance

[object](#) ← FishWarehouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

FishWarehouse()

```
public FishWarehouse()
```

Class GeorgeAndMollysHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "George and Molly's House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("George and Molly's House")]
public sealed class GeorgeAndMollysHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← GeorgeAndMollysHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

GeorgeAndMollysHouse()

```
public GeorgeAndMollysHouse()
```

Class GoblinHideBuilding

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Goblin Hide Building". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("NPC Hiding Building")]
public sealed class GoblinHideBuilding : IBuildingIdentifier
```

Inheritance

[object](#) ← GoblinHideBuilding

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

GoblinHideBuilding()

```
public GoblinHideBuilding()
```

Class HAMLegal

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "HAM Legal". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("HAM Legal")]
public sealed class HAMLegal : IBuildingIdentifier
```

Inheritance

[object](#) ← HAMLegal

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

HAMLegal()

```
public HAMLegal()
```

Class HoltHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Holt House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Holt House")]
public sealed class HoltHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← HoltHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

HoltHouse()

```
public HoltHouse()
```

Class HylandBank

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Hyland Bank". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Hyland Bank")]
public sealed class HylandBank : IBuildingIdentifier
```

Inheritance

[object](#) ← HylandBank

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

HylandBank()

```
public HylandBank()
```

Class HylandMedical

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Hyland Medical". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Hyland Medical")]
public sealed class HylandMedical : IBuildingIdentifier
```

Inheritance

[object](#) ← HylandMedical

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

HylandMedical()

```
public HylandMedical()
```

Interface IBuildingIdentifier

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Marker interface for building identifier types used with Building.Get<T>(). Implement empty classes like 'public sealed class NorthApartments : IBuildingIdentifier {}' and optionally annotate with [BuildingName("North apartments")].

```
public interface IBuildingIdentifier
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Class JanesCaravan

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Jane's Caravan". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Jane's Caravan")]
public sealed class JanesCaravan : IBuildingIdentifier
```

Inheritance

[object](#) ← JanesCaravan

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

JanesCaravan()

```
public JanesCaravan()
```

Class Jerry'sTent

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Jerry's Tent". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Jerry's Tent")]
public sealed class Jerry'sTent : IBuildingIdentifier
```

Inheritance

[object](#) ← Jerry'sTent

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Jerry'sTent()

```
public Jerry'sTent()
```

Class JessisRoom

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Jessi's Room". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Jessi's Room")]
public sealed class JessisRoom : IBuildingIdentifier
```

Inheritance

[object](#) ← JessisRoom

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

JessisRoom()

```
public JessisRoom()
```

Class KennedyHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Kennedy House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Kennedy House")]
public sealed class KennedyHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← KennedyHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

KennedyHouse()

```
public KennedyHouse()
```

Class KnightHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Knight House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Knight House")]
public sealed class KnightHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← KnightHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

KnightHouse()

```
public KnightHouse()
```

Class KyleAndAustinsHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Kyle and Austin's House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Kyle and Austin's House")]
public sealed class KyleAndAustinsHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← KyleAndAustinsHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

KyleAndAustinsHouse()

```
public KyleAndAustinsHouse()
```

Class LeosShippingContainer

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Leos' Shipping Container". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Leos' Shipping Container")]
public sealed class LeosShippingContainer : IBuildingIdentifier
```

Inheritance

[object](#) ← LeosShippingContainer

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

LeosShippingContainer()

```
public LeosShippingContainer()
```

Class LesOrdurePuantes

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Les Ordure Puantes". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Les Ordure Puantes")]
public sealed class LesOrdurePuantes : IBuildingIdentifier
```

Inheritance

[object](#) ← LesOrdurePuantes

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

LesOrdurePuantes()

```
public LesOrdurePuantes()
```

Class MayorsHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Mayor's House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Mayor's House")]
public sealed class MayorsHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← MayorsHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

MayorsHouse()

```
public MayorsHouse()
```

Class MicksHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Mick's House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Mick's House")]
public sealed class MicksHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← MicksHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

MicksHouse()

```
public MicksHouse()
```

Class ModernMansion

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Modern Mansion". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Modern Mansion")]
public sealed class ModernMansion : IBuildingIdentifier
```

Inheritance

[object](#) ← ModernMansion

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ModernMansion()

```
public ModernMansion()
```

Class MotelOffice

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Motel Office". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Motel Office")]
public sealed class MotelOffice : IBuildingIdentifier
```

Inheritance

[object](#) ← MotelOffice

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

MotelOffice()

```
public MotelOffice()
```

Class Nightclub

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Nightclub". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Nightclub")]
public sealed class Nightclub : IBuildingIdentifier
```

Inheritance

[object](#) ← Nightclub

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Nightclub()

```
public Nightclub()
```

Class NorthApartments

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "North apartments". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("North apartments")]
public sealed class NorthApartments : IBuildingIdentifier
```

Inheritance

[object](#) ← NorthApartments

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

NorthApartments()

```
public NorthApartments()
```

Class NorthIndustrialBuilding

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "North Industrial Building". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("North Industrial Building")]
public sealed class NorthIndustrialBuilding : IBuildingIdentifier
```

Inheritance

[object](#) ← NorthIndustrialBuilding

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NorthIndustrialBuilding()

```
public NorthIndustrialBuilding()
```

Class NorthWarehouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "North Warehouse". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("North Warehouse")]
public sealed class NorthWarehouse : IBuildingIdentifier
```

Inheritance

[object](#) ← NorthWarehouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

NorthWarehouse()

```
public NorthWarehouse()
```

Class OverpassBuilding

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Overpass Building". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Overpass Building")]
public sealed class OverpassBuilding : IBuildingIdentifier
```

Inheritance

[object](#) ← OverpassBuilding

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

OverpassBuilding()

```
public OverpassBuilding()
```

Class PawnShop

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Pawn Shop". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Pawn Shop")]
public sealed class PawnShop : IBuildingIdentifier
```

Inheritance

[object](#) ← PawnShop

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

PawnShop()

```
public PawnShop()
```

Class PetersRoom

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Peter's Room". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Peter's Room")]
public sealed class PetersRoom : IBuildingIdentifier
```

Inheritance

[object](#) ← PetersRoom

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

PetersRoom()

```
public PetersRoom()
```

Class Pillville

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Pillville". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Pillville")]
public sealed class Pillville : IBuildingIdentifier
```

Inheritance

[object](#) ← Pillville

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Pillville()

```
public Pillville()
```

Class PoliceStation

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Police Station". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Police Station")]
public sealed class PoliceStation : IBuildingIdentifier
```

Inheritance

[object](#) ← PoliceStation

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

PoliceStation()

```
public PoliceStation()
```

Class RandysBaitTackle

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Randy's Bait & Tackle". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Randy's Bait & Tackle")]
public sealed class RandysBaitTackle : IBuildingIdentifier
```

Inheritance

[object](#) ← RandysBaitTackle

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

RandysBaitTackle()

```
public RandysBaitTackle()
```

Class Room1

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Room 1". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Room 1")]
public sealed class Room1 : IBuildingIdentifier
```

Inheritance

[object](#) ← Room1

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Room1()

```
public Room1()
```

Class Room2

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Room 2". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Room 2")]
public sealed class Room2 : IBuildingIdentifier
```

Inheritance

[object](#) ← Room2

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Room2()

```
public Room2()
```

Class Room3

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Room 3". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Room 3")]
public sealed class Room3 : IBuildingIdentifier
```

Inheritance

[object](#) ← Room3

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Room3()

```
public Room3()
```

Class SauerkrautSupreme

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Sauerkraut Supreme". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Sauerkraut Supreme")]
public sealed class SauerkrautSupreme : IBuildingIdentifier
```

Inheritance

[object](#) ← SauerkrautSupreme

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

SauerkrautSupreme()

```
public SauerkrautSupreme()
```

Class Shack

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Shack". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Shack")]
public sealed class Shack : IBuildingIdentifier
```

Inheritance

[object](#) ← Shack

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Shack()

```
public Shack()
```

Class ShermanHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Sherman House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Sherman House")]
public sealed class ShermanHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← ShermanHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ShermanHouse()

```
public ShermanHouse()
```

Class ShootingRange

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Shooting Range". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Shooting Range")]
public sealed class ShootingRange : IBuildingIdentifier
```

Inheritance

[object](#) ← ShootingRange

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ShootingRange()

```
public ShootingRange()
```

Class SlopShop

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Slop Shop". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Slop Shop")]
public sealed class SlopShop : IBuildingIdentifier
```

Inheritance

[object](#) ← SlopShop

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

SlopShop()

```
public SlopShop()
```

Class SouthOverpassBuilding

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "South Overpass Building". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("South Overpass Building")]
public sealed class SouthOverpassBuilding : IBuildingIdentifier
```

Inheritance

[object](#) ← SouthOverpassBuilding

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

SouthOverpassBuilding()

```
public SouthOverpassBuilding()
```

Class StevensonHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Stevenson House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Stevenson House")]
public sealed class StevensonHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← StevensonHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

StevensonHouse()

```
public StevensonHouse()
```

Class StorageUnit2

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Storage Unit #2". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Storage Unit #2")]
public sealed class StorageUnit2 : IBuildingIdentifier
```

Inheritance

[object](#) ← StorageUnit2

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

StorageUnit2()

```
public StorageUnit2()
```

Class StorageUnit3

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Storage Unit #3". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Storage Unit #3")]
public sealed class StorageUnit3 : IBuildingIdentifier
```

Inheritance

[object](#) ← StorageUnit3

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

StorageUnit3()

```
public StorageUnit3()
```

Class Supermarket

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Supermarket". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Supermarket")]
public sealed class Supermarket : IBuildingIdentifier
```

Inheritance

[object](#) ← Supermarket

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Supermarket()

```
public Supermarket()
```

Class TallTower

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Tall Tower". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Tall Tower")]
public sealed class TallTower : IBuildingIdentifier
```

Inheritance

[object](#) ← TallTower

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

TallTower()

```
public TallTower()
```

Class TheCrimsonCanary

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "The Crimson Canary". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("The Crimson Canary")]
public sealed class TheCrimsonCanary : IBuildingIdentifier
```

Inheritance

[object](#) ← TheCrimsonCanary

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

TheCrimsonCanary()

```
public TheCrimsonCanary()
```

Class ThePissHut

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "The Piss Hut". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("The Piss Hut")]
public sealed class ThePissHut : IBuildingIdentifier
```

Inheritance

[object](#) ← ThePissHut

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ThePissHut()

```
public ThePissHut()
```

Class ThompsonConstructionDemo

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Thompson Construction & Demo". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Thompson Construction & Demo")]
public sealed class ThompsonConstructionDemo : IBuildingIdentifier
```

Inheritance

[object](#) ← ThompsonConstructionDemo

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ThompsonConstructionDemo()

```
public ThompsonConstructionDemo()
```

Class ThompsonHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Thompson House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Thompson House")]
public sealed class ThompsonHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← ThompsonHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ThompsonHouse()

```
public ThompsonHouse()
```

Class TownHall

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Town hall". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Town hall")]
public sealed class TownHall : IBuildingIdentifier
```

Inheritance

[object](#) ← TownHall

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

TownHall()

```
public TownHall()
```

Class UpscaleApartments

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Upscale Apartments". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Upscale Apartments")]
public sealed class UpscaleApartments : IBuildingIdentifier
```

Inheritance

[object](#) ← UpscaleApartments

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

UpscaleApartments()

```
public UpscaleApartments()
```

Class WebsterHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Webster House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Webster House")]
public sealed class WebsterHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← WebsterHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

WebsterHouse()

```
public WebsterHouse()
```

Class WeisCabin

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Wei's Cabin". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Wei's Cabin")]
public sealed class WeisCabin : IBuildingIdentifier
```

Inheritance

[object](#) ← WeisCabin

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

WeisCabin()

```
public WeisCabin()
```

Class WestGasMart

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "West Gas-Mart". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("West Gas-Mart")]
public sealed class WestGasMart : IBuildingIdentifier
```

Inheritance

[object](#) ← WestGasMart

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

WestGasMart()

```
public WestGasMart()
```

Class WilkinsonHouse

Namespace: [S1API.MapBuildings](#)

Assembly: S1API.dll

Identifier for the base-game building named "Wilkinson House". Modders can use [Get<T>\(\)](#) to resolve it.

```
[BuildingName("Wilkinson House")]
public sealed class WilkinsonHouse : IBuildingIdentifier
```

Inheritance

[object](#) ← WilkinsonHouse

Implements

[IBuildingIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

WilkinsonHouse()

```
public WilkinsonHouse()
```

Namespace S1API.Map.DeliveryLocations

Classes

[AlleywayHAMLegal](#)

[AlleywayTopTattoo](#)

[BasketballCourt](#)

[BehindAutoShop](#)

[BehindBank](#)

[BehindCasino](#)

[BehindConstructionSite](#)

[BehindCrimsonCanary](#)

[BehindDealership](#)

[BehindDiner](#)

[BehindGreenHouse](#)

[BehindHandyHanks](#)

[BehindHylandRange](#)

[BehindPawnShop](#)

[BehindPostOffice](#)

[BehindRandysBaitTackle](#)

[BehindRaysRealty](#)

[BehindSauerkrautSupreme](#)

[BehindShippingContainersDocks](#)

[BehindSuburbsBusStop](#)

[BehindTacoTicklers](#)

[BehindThompsonConstruction](#)

[BehindThriftyThreads](#)

[BehindWesternGasMart](#)

[BrickWarehouseDocks](#)

[CentralCanal](#)

[ConstructionSite](#)

[DeliveryLocationNameAttribute](#)

Annotate delivery location identifier types with the display name of the delivery location. Usage:
[DeliveryLocationName("Behind Bank")]

[DestroyedRV](#)

[Graveyard](#)

[InFrontOfMotel](#)

[NextToBarbershop](#)

[NextToBudsBar](#)

[NextToCourthouse](#)

[NextToMedicalCenter](#)

[NextToParkingGarage](#)

[NextToWestParkStatue](#)

[NorthWaterfront](#)

[OutsideBrownApartmentRoom2](#)

[ResidentialPark](#)

[ShippingContainersSauerkrautSupreme](#)

[Skatepark](#)

[SouthOverpass](#)

[UnderWestBridge](#)

[WestWarehouseWharf](#)

[WesternWharf](#)

[WharfFrenchRestaurant](#)

Interfaces

[IDeliveryLocationIdentifier](#)

Marker interface for delivery location identifier types used with DeliveryLocations.Get<T>(). Implement empty classes like 'public sealed class BehindBank : IDeliveryLocationIdentifier {}' and optionally annotate with [DeliveryLocationName("Behind Bank")].

Class AlleywayHAMLegal

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Alleyway next to HAM legal services")]
public sealed class AlleywayHAMLegal : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← AlleywayHAMLegal

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

AlleywayHAMLegal()

```
public AlleywayHAMLegal()
```

Class AlleywayTopTattoo

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Alleyway behind Top Tattoo")]
public sealed class AlleywayTopTattoo : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← AlleywayTopTattoo

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

AlleywayTopTattoo()

```
public AlleywayTopTattoo()
```

Class BasketballCourt

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Basketball court")]
public sealed class BasketballCourt : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BasketballCourt

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BasketballCourt()

```
public BasketballCourt()
```

Class BehindAutoShop

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the auto shop")]
public sealed class BehindAutoShop : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindAutoShop

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindAutoShop()

```
public BehindAutoShop()
```

Class BehindBank

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Bank")]
public sealed class BehindBank : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindBank

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindBank()

```
public BehindBank()
```

Class BehindCasino

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the casino")]
public sealed class BehindCasino : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindCasino

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindCasino()

```
public BehindCasino()
```

Class BehindConstructionSite

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the construction site")]
public sealed class BehindConstructionSite : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindConstructionSite

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindConstructionSite()

```
public BehindConstructionSite()
```

Class BehindCrimsonCanary

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the Crimson Canary")]
public sealed class BehindCrimsonCanary : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindCrimsonCanary

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindCrimsonCanary()

```
public BehindCrimsonCanary()
```

Class BehindDealership

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the dealership")]
public sealed class BehindDealership : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindDealership

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindDealership()

```
public BehindDealership()
```

Class BehindDiner

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind diner")]
public sealed class BehindDiner : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindDiner

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindDiner()

```
public BehindDiner()
```

Class BehindGreenHouse

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind green house")]
public sealed class BehindGreenHouse : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindGreenHouse

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindGreenHouse()

```
public BehindGreenHouse()
```

Class BehindHandyHanks

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Handy Hank's")]
public sealed class BehindHandyHanks : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindHandyHanks

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindHandyHanks()

```
public BehindHandyHanks()
```

Class BehindHylandRange

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Hyland Range")]
public sealed class BehindHylandRange : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindHylandRange

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindHylandRange()

```
public BehindHylandRange()
```

Class BehindPawnShop

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind pawn shop")]
public sealed class BehindPawnShop : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindPawnShop

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindPawnShop()

```
public BehindPawnShop()
```

Class BehindPostOffice

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the Post Office")]
public sealed class BehindPostOffice : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindPostOffice

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindPostOffice()

```
public BehindPostOffice()
```

Class BehindRandysBaitTackle

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Randy's bait & tackle")]
public sealed class BehindRandysBaitTackle : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindRandysBaitTackle

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindRandysBaitTackle()

```
public BehindRandysBaitTackle()
```

Class BehindRaysRealty

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Ray's Realty")]
public sealed class BehindRaysRealty : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindRaysRealty

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindRaysRealty()

```
public BehindRaysRealty()
```

Class BehindSauerkrautSupreme

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Sauerkraut Supreme Pizzeria")]
public sealed class BehindSauerkrautSupreme : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindSauerkrautSupreme

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindSauerkrautSupreme()

```
public BehindSauerkrautSupreme()
```

Class BehindShippingContainersDocks

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the shipping containers at the docks")]
public sealed class BehindShippingContainersDocks : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindShippingContainersDocks

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindShippingContainersDocks()

```
public BehindShippingContainersDocks()
```

Class BehindSuburbsBusStop

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the suburbs bus stop")]
public sealed class BehindSuburbsBusStop : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindSuburbsBusStop

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindSuburbsBusStop()

```
public BehindSuburbsBusStop()
```

Class BehindTacoTicklers

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Taco Ticklers")]
public sealed class BehindTacoTicklers : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindTacoTicklers

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindTacoTicklers()

```
public BehindTacoTicklers()
```

Class BehindThompsonConstruction

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Thompson construction and demolition")]
public sealed class BehindThompsonConstruction : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindThompsonConstruction

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindThompsonConstruction()

```
public BehindThompsonConstruction()
```

Class BehindThriftyThreads

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind Thrifty Threads")]
public sealed class BehindThriftyThreads : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindThriftyThreads

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindThriftyThreads()

```
public BehindThriftyThreads()
```

Class BehindWesternGasMart

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Behind the western Gas Mart")]
public sealed class BehindWesternGasMart : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BehindWesternGasMart

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BehindWesternGasMart()

```
public BehindWesternGasMart()
```

Class BrickWarehouseDocks

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Brick warehouse at the docks")]
public sealed class BrickWarehouseDocks : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← BrickWarehouseDocks

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BrickWarehouseDocks()

```
public BrickWarehouseDocks()
```

Class CentralCanal

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Central canal")]
public sealed class CentralCanal : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← CentralCanal

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

CentralCanal()

```
public CentralCanal()
```

Class ConstructionSite

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Construction site")]
public sealed class ConstructionSite : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← ConstructionSite

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ConstructionSite()

```
public ConstructionSite()
```

Class DeliveryLocationNameAttribute

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

Annotate delivery location identifier types with the display name of the delivery location. Usage:
[DeliveryLocationName("Behind Bank")]

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false, Inherited = false)]
public sealed class DeliveryLocationNameAttribute : Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← DeliveryLocationNameAttribute

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DeliveryLocationNameAttribute(string)

```
public DeliveryLocationNameAttribute(string name)
```

Parameters

name [string](#)

Properties

Name

```
public string Name { get; }
```

Property Value

[string](#)

Class DestroyedRV

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Destroyed RV")]
public sealed class DestroyedRV : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← DestroyedRV

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DestroyedRV()

```
public DestroyedRV()
```

Class Graveyard

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Graveyard")]
public sealed class Graveyard : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← Graveyard

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Graveyard()

```
public Graveyard()
```

Interface IDeliveryLocationIdentifier

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

Marker interface for delivery location identifier types used with DeliveryLocations.Get<T>(). Implement empty classes like 'public sealed class BehindBank : IDeliveryLocationIdentifier {}' and optionally annotate with [DeliveryLocationName("Behind Bank")].

```
public interface IDeliveryLocationIdentifier
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Class InFrontOfMotel

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("In front of motel")]
public sealed class InFrontOfMotel : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← InFrontOfMotel

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

InFrontOfMotel()

```
public InFrontOfMotel()
```

Class NextToBarbershop

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Next to barbershop")]
public sealed class NextToBarbershop : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NextToBarbershop

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NextToBarbershop()

```
public NextToBarbershop()
```

Class NextToBudsBar

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Next to Bud's Bar")]
public sealed class NextToBudsBar : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NextToBudsBar

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NextToBudsBar()

```
public NextToBudsBar()
```

Class NextToCourthouse

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Next to the courthouse")]
public sealed class NextToCourthouse : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NextToCourthouse

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NextToCourthouse()

```
public NextToCourthouse()
```

Class NextToMedicalCenter

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Next to Medical Center")]
public sealed class NextToMedicalCenter : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NextToMedicalCenter

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NextToMedicalCenter()

```
public NextToMedicalCenter()
```

Class NextToParkingGarage

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Next to parking garage")]
public sealed class NextToParkingGarage : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NextToParkingGarage

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NextToParkingGarage()

```
public NextToParkingGarage()
```

Class NextToWestParkStatue

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Next to the statue in the west park")]
public sealed class NextToWestParkStatue : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NextToWestParkStatue

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NextToWestParkStatue()

```
public NextToWestParkStatue()
```

Class NorthWaterfront

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("North waterfront")]
public sealed class NorthWaterfront : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← NorthWaterfront

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NorthWaterfront()

```
public NorthWaterfront()
```

Class OutsideBrownApartmentRoom2

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Outside brown apartment block room #2")]
public sealed class OutsideBrownApartmentRoom2 : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← OutsideBrownApartmentRoom2

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

OutsideBrownApartmentRoom2()

```
public OutsideBrownApartmentRoom2()
```

Class ResidentialPark

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Residential park")]
public sealed class ResidentialPark : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← ResidentialPark

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ResidentialPark()

```
public ResidentialPark()
```

Class ShippingContainersSauerkrautSupreme

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Shipping containers near Sauerkraut Supreme")]
public sealed class ShippingContainersSauerkrautSupreme
: IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← ShippingContainersSauerkrautSupreme

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ShippingContainersSauerkrautSupreme()

```
public ShippingContainersSauerkrautSupreme()
```

Class Skatepark

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Skatepark")]
public sealed class Skatepark : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← Skatepark

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Skatepark()

```
public Skatepark()
```

Class SouthOverpass

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("South Overpass")]
public sealed class SouthOverpass : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← SouthOverpass

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

SouthOverpass()

```
public SouthOverpass()
```

Class UnderWestBridge

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Under west bridge")]
public sealed class UnderWestBridge : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← UnderWestBridge

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

UnderWestBridge()

```
public UnderWestBridge()
```

Class WestWarehouseWharf

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("West Warehouse Wharf")]
public sealed class WestWarehouseWharf : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← WestWarehouseWharf

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WestWarehouseWharf()

```
public WestWarehouseWharf()
```

Class WesternWharf

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Western wharf")]
public sealed class WesternWharf : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← WesternWharf

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WesternWharf()

```
public WesternWharf()
```

Class WharfFrenchRestaurant

Namespace: [S1API.Map.DeliveryLocations](#)

Assembly: S1API.dll

```
[DeliveryLocationName("Wharf near french restaurant")]
public sealed class WharfFrenchRestaurant : IDeliveryLocationIdentifier
```

Inheritance

[object](#) ← WharfFrenchRestaurant

Implements

[IDeliveryLocationIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WharfFrenchRestaurant()

```
public WharfFrenchRestaurant()
```

Namespace S1API.Map.ParkingLots

Classes

[BillyParking](#)

Identifier for the Billy parking lot.

[ChemicalPlantBase](#)

Identifier for the Chemical Plant Base parking lot.

[ConstructionParking](#)

Identifier for the Construction parking lot.

[CorporatePlazaParkingLot](#)

Identifier for the Corporate plaza parking lot.

[DealershipParkingLot](#)

Identifier for the Dealership Parking Lot.

[FastFoodParkingLot](#)

Identifier for the Fast food parking lot.

[HansonHouseParking](#)

Identifier for the Hanson house parking lot.

[HoltHouseParking](#)

Identifier for the Holt house parking lot.

[KennedyHouseParking](#)

Identifier for the Kennedy House Parking lot.

[KnightHouseParking](#)

Identifier for the Knight house parking lot.

[ManorParking](#)

Identifier for the Manor parking lot.

[MansionParking](#)

Identifier for the Mansion parking lot.

[MedicalCenterParkingLot](#)

Identifier for the Medical center parking lot.

[MeetingPoint](#)

Identifier for the Meeting point parking lot.

[MicksHouseParking](#)

Identifier for Mick's house parking lot.

[ParkingGarage](#)

Identifier for the Parking Garage.

[ParkingLot](#)

Identifier for the generic "Parking lot".

[ParkingLot2](#)

Identifier for the second "Parking lot" (duplicate name in the list). Note: This is the second instance of "Parking lot" in the game.

[PizzeriaParkingLot](#)

Identifier for the Pizzeria parking lot.

[ThompsonHouseParking](#)

Identifier for the Thompson house parking lot.

[WestParkingLot](#)

Identifier for the West parking lot.

[WilkinsonHouseParking](#)

Identifier for the Wilkinson house parking lot.

Class BillyParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Billy parking lot.

```
[ParkingLotName("Billy parking")]
public sealed class BillyParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← BillyParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

BillyParking()

```
public BillyParking()
```

Class ChemicalPlantBase

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Chemical Plant Base parking lot.

```
[ParkingLotName("chemicalplantbase")]
public sealed class ChemicalPlantBase : IParkingLotIdentifier
```

Inheritance

[object](#) ← ChemicalPlantBase

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ChemicalPlantBase()

```
public ChemicalPlantBase()
```

Class ConstructionParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Construction parking lot.

```
[ParkingLotName("Construction parking")]
public sealed class ConstructionParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← ConstructionParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ConstructionParking()

```
public ConstructionParking()
```

Class CorporatePlazaParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Corporate plaza parking lot.

```
[ParkingLotName("Corporate plaza parking lot")]
public sealed class CorporatePlazaParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← CorporatePlazaParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

CorporatePlazaParkingLot()

```
public CorporatePlazaParkingLot()
```

Class DealershipParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Dealership Parking Lot.

```
[ParkingLotName("Dealership Parking Lot")]
public sealed class DealershipParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← DealershipParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DealershipParkingLot()

```
public DealershipParkingLot()
```

Class FastFoodParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Fast food parking lot.

```
[ParkingLotName("Fast food parking lot")]
public sealed class FastFoodParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← FastFoodParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

FastFoodParkingLot()

```
public FastFoodParkingLot()
```

Class HansonHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Hanson house parking lot.

```
[ParkingLotName("Hanson house parking")]
public sealed class HansonHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← HansonHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

HansonHouseParking()

```
public HansonHouseParking()
```

Class HoltHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Holt house parking lot.

```
[ParkingLotName("Holt house parking")]
public sealed class HoltHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← HoltHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

HoltHouseParking()

```
public HoltHouseParking()
```

Class KennedyHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Kennedy House Parking lot.

```
[ParkingLotName("Kennedy House Parking")]
public sealed class KennedyHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← KennedyHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

KennedyHouseParking()

```
public KennedyHouseParking()
```

Class KnightHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Knight house parking lot.

```
[ParkingLotName("Knight house parking")]
public sealed class KnightHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← KnightHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

KnightHouseParking()

```
public KnightHouseParking()
```

Class ManorParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Manor parking lot.

```
[ParkingLotName("Manor parking")]
public sealed class ManorParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← ManorParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ManorParking()

```
public ManorParking()
```

Class MansionParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Mansion parking lot.

```
[ParkingLotName("Mansion parking")]
public sealed class MansionParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← MansionParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MansionParking()

```
public MansionParking()
```

Class MedicalCenterParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Medical center parking lot.

```
[ParkingLotName("Medical center parking lot")]
public sealed class MedicalCenterParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← MedicalCenterParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MedicalCenterParkingLot()

```
public MedicalCenterParkingLot()
```

Class MeetingPoint

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Meeting point parking lot.

```
[ParkingLotName("Meeting point")]
public sealed class MeetingPoint : IParkingLotIdentifier
```

Inheritance

[object](#) ← MeetingPoint

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MeetingPoint()

```
public MeetingPoint()
```

Class MicksHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for Mick's house parking lot.

```
[ParkingLotName("Mick's house parking")]
public sealed class MicksHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← MicksHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MicksHouseParking()

```
public MicksHouseParking()
```

Class ParkingGarage

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Parking Garage.

```
[ParkingLotName("ParkingGarage")]
public sealed class ParkingGarage : IParkingLotIdentifier
```

Inheritance

[object](#) ← ParkingGarage

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ParkingGarage()

```
public ParkingGarage()
```

Class ParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the generic "Parking lot".

```
[ParkingLotName("Parking lot")]
public sealed class ParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← ParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ParkingLot()

```
public ParkingLot()
```

Class ParkingLot2

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the second "Parking lot" (duplicate name in the list). Note: This is the second instance of "Parking lot" in the game.

```
[ParkingLotName("Parking lot")]
public sealed class ParkingLot2 : IParkingLotIdentifier
```

Inheritance

[object](#) ← ParkingLot2

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ParkingLot2()

```
public ParkingLot2()
```

Class PizzeriaParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Pizzeria parking lot.

```
[ParkingLotName("Pizzeria parking lot")]
public sealed class PizzeriaParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← PizzeriaParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

PizzeriaParkingLot()

```
public PizzeriaParkingLot()
```

Class ThompsonHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Thompson house parking lot.

```
[ParkingLotName("Thompson house parking")]
public sealed class ThompsonHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← ThompsonHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ThompsonHouseParking()

```
public ThompsonHouseParking()
```

Class WestParkingLot

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the West parking lot.

```
[ParkingLotName("West parking lot")]
public sealed class WestParkingLot : IParkingLotIdentifier
```

Inheritance

[object](#) ← WestParkingLot

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WestParkingLot()

```
public WestParkingLot()
```

Class WilkinsonHouseParking

Namespace: [S1API.Map.ParkingLots](#)

Assembly: S1API.dll

Identifier for the Wilkinson house parking lot.

```
[ParkingLotName("Wilkinson house parking")]
public sealed class WilkinsonHouseParking : IParkingLotIdentifier
```

Inheritance

[object](#) ← WilkinsonHouseParking

Implements

[IParkingLotIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WilkinsonHouseParking()

```
public WilkinsonHouseParking()
```

Class Building

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Modder-facing wrapper for an enterable building in the world. Provides name-based lookup and basic metadata without exposing game types.

```
public sealed class Building
```

Inheritance

[object](#) ← Building

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

All

```
public static readonly List<Building> All
```

Field Value

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

Properties

Name

Display name of the building.

```
public string Name { get; }
```

Property Value

[string](#)

Methods

GetAll()

Returns all known buildings (sorted by name).

```
public static Building[] GetAll()
```

Returns

[Building](#)

GetByName(string)

Returns the first building with the provided display name, or null.

```
public static Building GetByName(string name)
```

Parameters

name [string](#)

Returns

[Building](#)

Get<T>()

Resolves a building using a typed identifier T. Declare an identifier class annotated with [Buildings.BuildingName("...")].

```
public static Building Get<T>() where T : IBuildingIdentifier
```

Returns

[Building](#)

Type Parameters

T

Class DeliveryLocation

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Wrapper for a base-game delivery location.

```
public sealed class DeliveryLocation
```

Inheritance

[object](#) ← DeliveryLocation

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

CustomerStandPoint

Customer standing position.

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

Property Value

Transform

Description

Human-readable description for UI.

```
public string Description { get; }
```

Property Value

[string](#)

GUID

GUID string identifier used by the base game.

```
public string GUID { get; }
```

Property Value

[string](#)

Name

Location display name.

```
public string Name { get; }
```

Property Value

[string](#)

TeleportPoint

Teleport target point near the location.

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

Property Value

Transform

Methods

GetAll()

Returns all known delivery locations (sorted by name).

```
public static DeliveryLocation[] GetAll()
```

Returns

[DeliveryLocation\[\]](#)

GetByGuid(string)

Returns a delivery location by GUID string.

```
public static DeliveryLocation GetByGuid(string guid)
```

Parameters

guid [string](#)

Returns

[DeliveryLocation](#)

GetByName(string)

Returns a delivery location by case-insensitive name match.

```
public static DeliveryLocation GetByName(string name)
```

Parameters

name [string](#) ↗

Returns

[DeliveryLocation](#)

Get<T>()

Resolves a delivery location using a typed identifier T. Declare an identifier class annotated with [DeliveryLocations.DeliveryLocationName("...")].

```
public static DeliveryLocation Get<T>() where T : IDeliveryLocationIdentifier
```

Returns

[DeliveryLocation](#)

Type Parameters

T

Interface IParkingLotIdentifier

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Marker interface for parking lot identifier classes. Use this with ParkingLotNameAttribute to create type-safe parking lot lookups.

```
public interface IParkingLotIdentifier
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Class ParkingData

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Defines where to park

```
public class ParkingData
```

Inheritance

[object](#) ← ParkingData

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

ParkingData(string, int, ParkingAlignment)

Defines where to park

```
public ParkingData(string _lotId, int _index, ParkingAlignment _align)
```

Parameters

_lotId [string](#)

_index [int](#)

_align [ParkingAlignment](#)

Properties

Alignment

```
public ParkingAlignment Alignment { get; }
```

Property Value

[ParkingAlignment](#)

Index

```
public int Index { get; }
```

Property Value

[int](#)

LotId

```
public string LotId { get; }
```

Property Value

[string](#)

Class ParkingLotNameAttribute

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Attribute to mark a parking lot identifier class with its GameObject name.

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false)]
public sealed class ParkingLotNameAttribute : Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← [ParkingLotNameAttribute](#)

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

Constructors

ParkingLotNameAttribute(string)

Initializes a new instance of the ParkingLotNameAttribute.

```
public ParkingLotNameAttribute(string name)
```

Parameters

name [string](#)

The GameObject name of the parking lot.

Properties

Name

The GameObject name of the parking lot.

```
public string Name { get; }
```

Property Value

[string](#)

Class ParkingLotRegistry

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Registry and utilities for parking-related queries.

```
public static class ParkingLotRegistry
```

Inheritance

[object](#) ← ParkingLotRegistry

Inherited Members

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

Methods

GetAll()

Returns all lots currently registered.

```
public static ParkingLotWrapper[] GetAll()
```

Returns

[ParkingLotWrapper](#)[]

GetByGUID(string)

Finds a lot by GUID string. Returns null if not found.

```
public static ParkingLotWrapper GetByGUID(string guid)
```

Parameters

guid [string](#)

Returns

[ParkingLotWrapper](#)

GetByName(string)

Finds a parking lot by GameObject name.

```
public static ParkingLotWrapper GetByName(string gameObjectName)
```

Parameters

gameObjectname [string](#)

The name of the GameObject containing the ParkingLot component.

Returns

[ParkingLotWrapper](#)

A parking lot wrapper, or null if not found.

GetFreeSpots(string)

Finds free spots in a lot, by GUID.

```
public static ParkingSpotWrapper[] GetFreeSpots(string lotGuid)
```

Parameters

lotGuid [string](#)

Returns

[ParkingSpotWrapper\[\]](#)

GetFreeSpotsByName(string)

Finds free spots in a lot, by GameObject name.

```
public static ParkingSpotWrapper[] GetFreeSpotsByName(string lotGameObjectName)
```

Parameters

lotGameObjectName [string](#)

Returns

[ParkingSpotWrapper\[\]](#)

Get<T>()

Resolves a parking lot using a typed identifier T. Declare an identifier class annotated with [ParkingLotName("...")].

```
public static ParkingLotWrapper Get<T>() where T : IParkingLotIdentifier
```

Returns

[ParkingLotWrapper](#)

The parking lot wrapper, or null if not found.

Type Parameters

A type implementing IParkingLotIdentifier with ParkingLotNameAttribute

Class ParkingLotWrapper

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Modder-facing wrapper for a parking lot.

```
public sealed class ParkingLotWrapper
```

Inheritance

[object](#) ← ParkingLotWrapper

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

EntryPointPosition

Optional entry point world position, if configured.

```
public Vector3? EntryPointPosition { get; }
```

Property Value

Vector3?

GUID

Unique GUID string identifier for this lot.

```
public string GUID { get; }
```

Property Value

[string](#)

GameObjectName

GameObject name of the parking lot.

```
public string GameObjectName { get; }
```

Property Value

[string](#)

ParkingSpotsCount

Total amount of parking spots are in this lot, available or otherwise

```
public int ParkingSpotsCount { get; }
```

Property Value

[int](#)

Methods

GetSpot(int)

Gets the parking spot at the specified index, does not validate the index

```
public ParkingSpotWrapper GetSpot(int index)
```

Parameters

[index](#) [int](#)

Returns

[ParkingSpotWrapper](#)

Class ParkingSpotWrapper

Namespace: [S1API.Map](#)

Assembly: S1API.dll

Modder-facing wrapper for a parking spot within a lot.

```
public sealed class ParkingSpotWrapper
```

Inheritance

[object](#) ← ParkingSpotWrapper

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Alignment

Alignment enum for this spot.

```
public ParkingAlignment Alignment { get; }
```

Property Value

[ParkingAlignment](#)

AlignmentPoint

Alignment Point for Occupant Vehicle

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

Property Value

Transform

IsFree

True if the spot currently has no vehicle occupant.

```
public bool IsFree { get; }
```

Property Value

[bool](#) ↗

OccupantVehicle

LandVehicle currently occupying this spot

```
public LandVehicle? OccupantVehicle { get; }
```

Property Value

[LandVehicle](#)

Methods

SetOccupant(LandVehicle)

Sets the vehicle currently occupying this spot

```
public void SetOccupant(LandVehicle vehicle)
```

Parameters

vehicle [LandVehicle](#)

Enum Region

Namespace: [S1API.Map](#)

Assembly: S1API.dll

The regions available in the game.

```
public enum Region
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Docks = 3

The fourth region in the game.

Downtown = 2

The third region in the game.

Northtown = 0

The first region in the game.

Suburbia = 4

The fifth region in the game.

Uptown = 5

The sixth region in the game.

Westville = 1

The second region in the game.

Namespace S1API.Messaging

Classes

[Response](#)

Represents a message response displayed for the player.

Class Response

Namespace: [S1API.Messaging](#)

Assembly: S1API.dll

Represents a message response displayed for the player.

```
public class Response
```

Inheritance

[object](#) ← Response

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Response()

Creates a new response for displaying in-game.

```
public Response()
```

Properties

Label

The unique identifier for this response.

```
public string Label { get; set; }
```

Property Value

[string](#)

OnTriggered

A callback for when the response is triggered.

```
public Action? OnTriggered { get; set; }
```

Property Value

[Action](#)

Text

The text displayed in-game for the player.

```
public string Text { get; set; }
```

Property Value

[string](#)

Namespace S1API.Money

Classes

[CashDefinition](#)

Represents the definition of a cash type. NOTE: This is not the instance of cash, but the definition itself.

[CashInstance](#)

Represents an instance of cash within the game.

[Money](#)

Provides static access to financial operations, including methods for managing cash balance, creating online transactions, and calculating net worth.

Class CashDefinition

Namespace: [S1API.Money](#)

Assembly: S1API.dll

Represents the definition of a cash type. NOTE: This is not the instance of cash, but the definition itself.

```
public class CashDefinition : ItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← CashDefinition

Inherited Members

[ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) , [ItemDefinition.ID](#) , [ItemDefinition.Name](#) ,
[ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) , [ItemDefinition.Category](#) , [ItemDefinition.Icon](#) ,
[ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) , [ItemDefinition.LabelDisplayColor](#) ,
[ItemDefinition.Keywords](#) , [ItemDefinition.GUID](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ToString\(\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

CreateInstance(int)

Creates an instance of the cash definition in-game.

```
public override ItemInstance CreateInstance(int quantity = 1)
```

Parameters

quantity [int](#)

Returns

ItemInstance

Class CashInstance

Namespace: [S1API.Money](#)

Assembly: S1API.dll

Represents an instance of cash within the game.

```
public class CashInstance : ItemInstance
```

Inheritance

[object](#) ← [ItemInstance](#) ← CashInstance

Inherited Members

[ItemInstance.Definition](#) , [ItemInstance.Quantity](#) , [ItemInstance.IsStackable](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Methods

AddQuantity(float)

Adds to the quantity of cash for this instance. NOTE: Supports negative numbers to remove.

```
public void AddQuantity(float amount)
```

Parameters

amount [float](#)

Quantity to set the cash to.

SetQuantity(float)

Sets the quantity of cash for this instance.

```
public void SetQuantity(float newQuantity)
```

Parameters

newQuantity [float](#)

Quantity to set the cash to.

Class Money

Namespace: [S1API.Money](#)

Assembly: S1API.dll

Provides static access to financial operations, including methods for managing cash balance, creating online transactions, and calculating net worth.

```
public static class Money
```

Inheritance

[object](#) ← Money

Inherited Members

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

Methods

AddNetworthCalculation(Action<object>)

Registers a callback to be invoked during net worth calculation.

```
public static void AddNetworthCalculation(Action<object> callback)
```

Parameters

callback [Action<object>](#)

The callback to be executed when net worth is calculated. It receives an object as its parameter.

ChangeCashBalance(float, bool, bool)

Adjusts the cash balance by the specified amount.

```
public static void ChangeCashBalance(float amount, bool visualizeChange = true, bool playCashSound = false)
```

Parameters

amount [float](#)

The amount to modify the cash balance by. Positive values increase the balance, and negative values decrease it.

visualizeChange [bool](#)

Indicates whether the cash change should be visualized on the HUD.

playCashSound [bool](#)

Indicates whether a sound effect should be played to signify the cash adjustment.

CreateCashInstance(float)

Creates a new cash instance with the specified balance.

```
public static CashInstance CreateCashInstance(float amount)
```

Parameters

amount [float](#)

The initial amount of cash to set in the instance.

Returns

[CashInstance](#)

A newly created instance of cash with the specified balance.

CreateOnlineTransaction(string, float, float, string)

Creates an online transaction.

```
public static void CreateOnlineTransaction(string transactionName, float unitAmount,  
float quantity, string transactionNote)
```

Parameters

transactionName [string](#)

The name of the transaction.

unitAmount [float](#)

The monetary amount per unit involved in the transaction.

quantity [float](#)

The number of units in the transaction.

transactionNote [string](#)

An optional note or description for the transaction.

GetCashBalance()

Retrieves the current cash balance.

```
public static float GetCashBalance()
```

Returns

[float](#)

The current cash balance as a floating-point value.

GetNetWorth()

Retrieves the total net worth, including all cash and online balances combined.

```
public static float GetNetWorth()
```

Returns

[float](#)

The total net worth as a floating-point value.

GetOnlineBalance()

Retrieves the current online balance.

```
public static float GetOnlineBalance()
```

Returns

[float](#)

The current amount of online balance.

RemoveNetworthCalculation(Action<object>)

Removes a previously registered networth calculation callback.

```
public static void RemoveNetworthCalculation(Action<object> callback)
```

Parameters

callback [Action](#)<[object](#)>

The callback to be removed from the networth calculation updates.

Events

OnBalanceChanged

Event triggered whenever there is a change in the balance, including cash balance or online transactions.

```
public static event Action? OnBalanceChanged
```

Event Type

[Action ↗](#)

Namespace S1API.PhoneApp

Classes

[PhoneApp](#)

Abstract base class for creating custom applications to be used within an in-game phone system.

Enums

[PhoneApp.EOrientation](#)

Represents the orientation settings for phone applications.

Class PhoneApp

Namespace: [S1API.PhoneApp](#)

Assembly: S1API.dll

Abstract base class for creating custom applications to be used within an in-game phone system.

```
public abstract class PhoneApp : Registerable
```

Inheritance

[object](#) ← [Registerable](#) ← PhoneApp

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This class provides an extensible framework for defining application behaviors, user interface elements, and registration mechanics for integration into the phone's ecosystem.

Constructors

PhoneApp()

```
protected PhoneApp()
```

Fields

Logger

Logger instance used for logging messages, warnings, or errors related to the functionality of in-game phone applications.

```
protected static readonly Log Logger
```

Field Value

[Log](#)

Properties

AppName

Gets the unique identifier for the application within the phone system.

```
protected abstract string AppName { get; }
```

Property Value

[string](#) ↗

Remarks

This property is used as a key to identify the application when creating UI elements or interacting with other components of the in-game phone system. It must be implemented in derived classes to provide a consistent and unique name for the application.

AppTitle

Gets the display title of the application as it appears in the in-game phone system.

```
protected abstract string AppTitle { get; }
```

Property Value

[string](#)

Remarks

This property specifies the human-readable name of the application, different from the internal [AppName](#) that uniquely identifies the app within the system. It is displayed to the user on the application icon or within the application UI.

IconFileName

Specifies the file name of the icon used to represent the phone application in the in-game phone system.

```
protected abstract string IconFileName { get; }
```

Property Value

[string](#)

Remarks

The value of this property is typically a string containing the file name of the icon asset, such as "icon-name.png". It is used to identify and load the appropriate icon for the application.

IconLabel

Gets the label text displayed on the application's icon.

```
protected abstract string IconLabel { get; }
```

Property Value

[string](#)

A string representing the label text displayed under the app icon, which explains or identifies the app to the user.

Remarks

The `IconLabel` property is an abstract member that must be overridden by each implementation of the `PhoneApp` class. It specifies the label text shown directly below the application's icon on the in-game phone's home screen. This property is utilized when creating or modifying the app's icon, as part of the `SpawnIcon` method, to ensure that the label represents the application's name or a relevant description. The value should be concise and contextually meaningful to the user.

IconSprite

Optional direct icon sprite. If provided, it takes precedence over `IconFileName`.

```
protected virtual Sprite? IconSprite { get; }
```

Property Value

Sprite

Orientation

Gets the orientation of the phone app (Horizontal or Vertical). Determines how the phone is rotated when the app is opened.

```
protected virtual PhoneApp.EOrientation Orientation { get; }
```

Property Value

[PhoneApp.EOrientation](#)

Methods

CloseApp()

Closes this phone application, cleaning up its state.

```
public void CloseApp()
```

Exit(ExitAction)

Handles exit/home button functionality. Called when user presses escape or home.

```
public virtual void Exit(ExitAction exit)
```

Parameters

`exit` ExitAction

IsOpen()

Determines if this phone app is currently open.

```
public bool IsOpen()
```

Returns

[bool](#)

True if the app is open, false otherwise

OnCreated()

Invoked when the PhoneApp instance is created. Responsible for registering the app with the PhoneAppRegistry, integrating it into the in-game phone system.

```
protected override void OnCreated()
```

OnCreatedUI(GameObject)

Invoked to define the user interface layout when the application panel is created. The method is used to populate the provided container with custom UI elements specific to the application.

```
protected abstract void OnCreatedUI(GameObject container)
```

Parameters

container GameObject

The GameObject container where the application's UI elements will be added.

OnDestroyed()

Cleans up resources and resets state when the app is destroyed. This method ensures any associated UI elements and resources are properly disposed of and variables tracking the app state are reset.

```
protected override void OnDestroyed()
```

OnPhoneClosed()

Called when the in-game phone is closed. Override in derived apps to reset state.

```
protected virtual void OnPhoneClosed()
```

OpenApp()

Opens this phone application, managing proper app state transitions.

```
public void OpenApp()
```

SetIconSprite(Sprite)

Sets the app icon directly from a sprite, bypassing file loading. If the icon is not yet spawned, stores the sprite to apply later.

```
public bool SetIconSprite(Sprite sprite)
```

Parameters

sprite Sprite

Sprite to use for the app icon.

Returns

bool ↗

True if applied immediately or stored for later application.

SetIconTexture(Texture2D)

Sets the app icon directly from a texture by creating a sprite.

```
public bool SetIconTexture(Texture2D texture)
```

Parameters

texture Texture2D

Texture to convert into a sprite for the app icon.

Returns

[bool](#)

True if applied or stored; false if texture is null.

Enum PhoneApp.EOrientation

Namespace: [S1API.PhoneApp](#)

Assembly: S1API.dll

Represents the orientation settings for phone applications.

```
public enum PhoneApp.EOrientation
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Horizontal = 0

Vertical = 1

Namespace S1API.PhoneCalls

Namespaces

[S1API.PhoneCalls.Constants](#)

Classes

[CallManager](#)

Provides a safe queuing layer on top of the game's CallManager. Multiple pending calls are stored and dispatched to the game one at a time.

[CallStageEntry](#)

Represents a Stage entry in a call.

[CallerDefinition](#)

Represents the definition of a ScheduleOne.ScriptableObjects.CallerID instance.

[PhoneCallDefinition](#)

Abstract base class for defining phone calls in Schedule One. Provides methods to create and configure phone call data including caller information and call stages.

Namespace S1API.PhoneCalls.Constants

Enums

[EvaluationType](#)

Defines how conditional triggers should evaluate their conditions.

[SystemTriggerType](#)

Defines the timing when system triggers should execute during a phone call stage.

Enum EvaluationType

Namespace: [S1API.PhoneCalls.Constants](#)

Assembly: S1API.dll

Defines how conditional triggers should evaluate their conditions.

```
public enum EvaluationType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`PassOnFalse = 1`

The trigger passes and executes when the condition evaluates to false.

`PassOnTrue = 0`

The trigger passes and executes when the condition evaluates to true.

Enum SystemTriggerType

Namespace: [S1API.PhoneCalls.Constants](#)

Assembly: S1API.dll

Defines the timing when system triggers should execute during a phone call stage.

```
public enum SystemTriggerType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`DoneTrigger = 1`

Trigger executes when the call stage completes, after the player has finished reading the text.

`StartTrigger = 0`

Trigger executes when the call stage begins, before the text is displayed to the player.

Class CallManager

Namespace: [S1API.PhoneCalls](#)

Assembly: S1API.dll

Provides a safe queuing layer on top of the game's CallManager. Multiple pending calls are stored and dispatched to the game one at a time.

```
public static class CallManager
```

Inheritance

[object](#) ← CallManager

Inherited Members

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

Properties

PendingCount

Number of calls currently pending in the S1API queue (excluding the game's active/queued call).

```
public static int PendingCount { get; }
```

Property Value

[int](#)

Methods

ClearPendingQueue()

Clears all pending S1API-queued calls. Does not affect the game's current queued/active call.

```
public static void ClearPendingQueue()
```

QueueCall(PhoneCallDefinition)

Enqueue a phone call to be played. If the game's queue is empty, this will be forwarded immediately; otherwise it will be held until the current call completes.

```
public static void QueueCall(PhoneCallDefinition phoneCallDefinition)
```

Parameters

phoneCallDefinition [PhoneCallDefinition](#)

Class CallStageEntry

Namespace: [S1API.PhoneCalls](#)

Assembly: S1API.dll

Represents a Stage entry in a call.

```
public class CallStageEntry
```

Inheritance

[object](#) ← CallStageEntry

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

DoneTriggerEntries

INTERNAL: List of system triggers that execute when this call stage completes. These triggers are executed after the player has finished reading the stage text.

```
protected readonly List<SystemTriggerEntry> DoneTriggerEntries
```

Field Value

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

StartTriggerEntries

INTERNAL: List of system triggers that execute when this call stage starts. These triggers are executed before the stage text is displayed to the player.

```
protected readonly List<SystemTriggerEntry> StartTriggerEntries
```

Field Value

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

Properties

Text

The text to display in this Stage

```
public string Text { get; set; }
```

Property Value

[string](#)

Methods

AddSystemTrigger(SystemTriggerType)

Adds a start trigger to the [CallStageEntry](#) instance.

```
public SystemTriggerEntry AddSystemTrigger(SystemTriggerType triggerType)
```

Parameters

triggerType [SystemTriggerType](#)

The [SystemTriggerType](#) this trigger has to be added to

Returns

[SystemTriggerEntry](#)

Class CallerDefinition

Namespace: [S1API.PhoneCalls](#)

Assembly: S1API.dll

Represents the definition of a ScheduleOne.ScriptableObjects.CallerID instance.

```
public class CallerDefinition
```

Inheritance

[object](#) ← CallerDefinition

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Name

The name of the caller.

```
public string Name { get; set; }
```

Property Value

[string](#)

ProfilePicture

The profile picture of the caller.

```
public Sprite? ProfilePicture { get; set; }
```

Property Value

Sprite

Class PhoneCallDefinition

Namespace: [S1API.PhoneCalls](#)

Assembly: S1API.dll

Abstract base class for defining phone calls in Schedule One. Provides methods to create and configure phone call data including caller information and call stages.

```
public abstract class PhoneCallDefinition
```

Inheritance

[object](#) ← PhoneCallDefinition

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

PhoneCallDefinition(NPC?)

Creates a new phone call definition using an NPC as the caller. Initializes the underlying PhoneCallData and sets up the caller information from the NPC.

```
protected PhoneCallDefinition(NPC? npcCallerID)
```

Parameters

npcCallerID [NPC](#)

The [NPC](#) instance to use for the caller information.

PhoneCallDefinition(string, Sprite?)

Public constructor used for instancing a new ScheduleOne.ScriptableObjects.PhoneCallData

```
protected PhoneCallDefinition(string name, Sprite? profilePicture = null)
```

Parameters

`name` [string](#)

The name of the caller

`profilePicture` [Sprite](#)

The sprite of the caller

Fields

Caller

The caller of the [PhoneCallDefinition](#) instance

```
public CallerDefinition? Caller
```

Field Value

[CallerDefinition](#)

S1PhoneCallData

The original ScheduleOne.ScriptableObjects.PhoneCallData instance

```
public readonly PhoneCallData S1PhoneCallData
```

Field Value

PhoneCallData

StageEntries

A list of all stage entries added to this phone call.

```
protected readonly List<CallStageEntry> StageEntries
```

Field Value

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

Methods

AddCallerID(NPC?)

Set's a new CallerID definition based of an existing [NPC](#) instance.

```
protected CallerDefinition AddCallerID(NPC? npc)
```

Parameters

npc [NPC](#)

The [NPC](#) instance to use for the caller

Returns

[CallerDefinition](#)

A reference to the CallerID definition

AddCallerID(string, Sprite?)

Set's a new CallerID definition to the PhoneCall

```
protected CallerDefinition AddCallerID(string name, Sprite? profilePicture = null)
```

Parameters

name [string](#)

The name of the caller

profilePicture [Sprite](#)

The sprite of the caller

Returns

[CallerDefinition](#)

A reference to the CallerID definition

AddStage(string)

Add's a [CallStageEntry](#) instance to the [S1PhoneCallData](#)

```
protected CallStageEntry AddStage(string text)
```

Parameters

text [string](#)

The text to display in this stage

Returns

[CallStageEntry](#)

A reference to the Stage entry

Completed()

Completes the ScheduleOne.ScriptableObjects.PhoneCallData instance.

```
public void Completed()
```

Namespace S1API.Products

Classes

[CocaineDefinition](#)

Defines the characteristics and behaviors of a cocaine product within the system.

[MethDefinition](#)

Represents the definition of a meth product within the ScheduleOne product framework.

[PackagingDefinition](#)

Represents a type of packaging in-game.

[ProductDefinition](#)

Represents a product definition in the game.

[ProductDefinitionWrapper](#)

Provides functionality to wrap and convert generic product definitions into their specific type-derived definitions.

[ProductInstance](#)

Represents an instance of a product in the game.

[ProductManager](#)

Provides management over all products in the game.

[ShroomAppearanceSettings](#)

Represents the visual appearance settings for mushroom products, including colors and spot patterns.

[ShroomDefinition](#)

Represents a specific type of mushroom (shroom) product definition.

[ShroomInstance](#)

Represents an instance of a mushroom (shroom) product.

[WeedDefinition](#)

Represents a specific type of weed product definition.

Enums

[DrugType](#)

API-safe product type enum. Mirrors base game drug types.

[Quality](#)

Represents the quality levels for items.

Class CocaineDefinition

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Defines the characteristics and behaviors of a cocaine product within the system.

```
public class CocaineDefinition : ProductDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← [ProductDefinition](#) ← CocaineDefinition

Inherited Members

[ProductDefinition.Price](#) , [ProductDefinition.BasePrice](#) , [ProductDefinition.MarketValue](#) ,
[ProductDefinition.Icon](#) , [ProductDefinition.Properties](#) , [ProductDefinition.DrugTypes](#) ,
[ProductDefinition.DrugType](#) , [ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) ,
[ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) ,
[ItemDefinition.Category](#) , [ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) ,
[ItemDefinition.LabelDisplayColor](#) , [ItemDefinition.Keywords](#) , [ItemDefinition.GUID](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

CreateInstance(int)

Creates an instance of this product definition with the specified quantity.

```
public override ItemInstance CreateInstance(int quantity = 1)
```

Parameters

quantity [int](#)

The quantity of the product to instantiate. Defaults to 1 if not specified.

Returns

[ItemInstance](#)

An [ItemInstance](#) representing the instantiated product with the specified quantity.

GetProperties()

Retrieves a list of properties associated with this product definition.

```
public List<PropertyBase> GetProperties()
```

Returns

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

A list of runtime-agnostic property wrappers that work on both Mono and IL2CPP.

GetRawProperties()

DEPRECATED: Use GetProperties() which returns IProperty wrappers. Retrieves the raw Schedule One properties (may cause assembly reference issues in IL2CPP).

```
[Obsolete("Use GetProperties() instead for IL2CPP compatibility")]
public List<Effect> GetRawProperties()
```

Returns

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

A list of raw Schedule One properties.

Enum DrugType

Namespace: [S1API.Products](#)

Assembly: S1API.dll

API-safe product type enum. Mirrors base game drug types.

```
public enum DrugType
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Cocaine = 2

Heroin = 5

MDMA = 3

Marijuana = 0

Methamphetamine = 1

Shrooms = 4

Class MethDefinition

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents the definition of a meth product within the ScheduleOne product framework.

```
public class MethDefinition : ProductDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← [ProductDefinition](#) ← MethDefinition

Inherited Members

[ProductDefinition.Price](#) , [ProductDefinition.BasePrice](#) , [ProductDefinition.MarketValue](#) ,
[ProductDefinition.Icon](#) , [ProductDefinition.Properties](#) , [ProductDefinition.DrugTypes](#) ,
[ProductDefinition.DrugType](#) , [ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) ,
[ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) ,
[ItemDefinition.Category](#) , [ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) ,
[ItemDefinition.LabelDisplayColor](#) , [ItemDefinition.Keywords](#) , [ItemDefinition.GUID](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

CreateInstance(int)

Creates an instance of this meth product with the specified quantity.

```
public override ItemInstance CreateInstance(int quantity = 1)
```

Parameters

quantity [int](#)

The quantity of the product instance to create. Defaults to 1 if not specified.

Returns

[ItemInstance](#)

An instance of [ItemInstance](#) representing the created meth product.

GetProperties()

Retrieves the list of properties associated with the meth product definition.

```
public List<PropertyBase> GetProperties()
```

Returns

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

A list of runtime-agnostic property wrappers that work on both Mono and IL2CPP.

GetRawProperties()

DEPRECATED: Use GetProperties() which returns IProperty wrappers. Retrieves the raw Schedule One properties (may cause assembly reference issues in IL2CPP).

```
[Obsolete("Use GetProperties() instead for IL2CPP compatibility")]
public List<Effect> GetRawProperties()
```

Returns

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

A list of raw Schedule One properties.

Class PackagingDefinition

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents a type of packaging in-game.

```
public class PackagingDefinition : ItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← PackagingDefinition

Inherited Members

[ItemDefinition.CreateInstance\(int\)](#) , [ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) ,
[ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) ,
[ItemDefinition.Category](#) , [ItemDefinition.Icon](#) , [ItemDefinition.AvailableInDemo](#) ,
[ItemDefinition.LegalStatus](#) , [ItemDefinition.LabelDisplayColor](#) , [ItemDefinition.Keywords](#) ,
[ItemDefinition.GUID](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Quantity

The quantity that this packaging can hold.

```
public int Quantity { get; }
```

Property Value

[int](#)

Class ProductDefinition

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents a product definition in the game.

```
public class ProductDefinition : ItemDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← ProductDefinition

Derived

[CocaineDefinition](#), [MethDefinition](#), [ShroomDefinition](#), [WeedDefinition](#)

Inherited Members

[ItemDefinition.Equals\(object\)](#), [ItemDefinition.GetHashCode\(\)](#), [ItemDefinition.ID](#), [ItemDefinition.Name](#),
[ItemDefinition.Description](#), [ItemDefinition.StackLimit](#), [ItemDefinition.Category](#),
[ItemDefinition.AvailableInDemo](#), [ItemDefinition.LegalStatus](#), [ItemDefinition.LabelDisplayColor](#),
[ItemDefinition.Keywords](#), [ItemDefinition.GUID](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),
[object.ToString\(\)](#), [object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#), [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

BasePrice

The base price associated with this product.

```
public float BasePrice { get; }
```

Property Value

[float](#)

DrugType

The primary drug type for this product (convenience property).

```
public EDrugType DrugType { get; }
```

Property Value

EDrugType

DrugTypes

The list of drug types associated with this product definition. Returns a C# list for IL2CPP builds to avoid type mismatches.

```
public List<DrugTypeContainer> DrugTypes { get; }
```

Property Value

[List](#)<DrugTypeContainer>

Icon

Gets the in-game icon associated with the product.

```
public Sprite Icon { get; }
```

Property Value

Sprite

MarketValue

The market value associated with this product.

```
public float MarketValue { get; }
```

Property Value

[float](#)

Price

The price associated with this product.

```
public float Price { get; }
```

Property Value

[float](#)

Properties

The list of product properties for this definition. Returns runtime-agnostic property wrappers that work on both Mono and IL2CPP.

```
public IReadOnlyList<PropertyBase> Properties { get; }
```

Property Value

[IReadOnlyList](#)<[PropertyBase](#)>

Methods

CreateInstance(int)

Creates an instance of this product in-game.

```
public override ItemInstance CreateInstance(int quantity = 1)
```

Parameters

quantity [int](#)

The quantity of product.

Returns

[ItemInstance](#)

An instance of the product.

Class ProductDefinitionWrapper

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Provides functionality to wrap and convert generic product definitions into their specific type-derived definitions.

```
public static class ProductDefinitionWrapper
```

Inheritance

[object](#) ← ProductDefinitionWrapper

Inherited Members

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

Methods

Wrap(ProductDefinition)

Converts a generic [ProductDefinition](#) into its corresponding typed wrapper.

```
public static ProductDefinition Wrap(ProductDefinition def)
```

Parameters

def [ProductDefinition](#)

The raw product definition to be processed and converted.

Returns

[ProductDefinition](#)

A wrapped instance of [ProductDefinition](#) with type-specific methods and properties, or the input definition if no specific wrapper applies.

Class ProductInstance

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents an instance of a product in the game.

```
public class ProductInstance : ItemInstance
```

Inheritance

[object](#) ← [ItemInstance](#) ← ProductInstance

Derived

[ShroomInstance](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

This class defines specific properties and behaviors for a product instance, such as quality, packaging, and definition, derived from the S1API's item instance structure.

Properties

AppliedPackaging

Provides access to the packaging information applied to the product, represented as a specific packaging definition instance.

```
public PackagingDefinition AppliedPackaging { get; }
```

Property Value

[PackagingDefinition](#)

Definition

Gets the definition of the product associated with this instance.

```
public ProductDefinition Definition { get; }
```

Property Value

[ProductDefinition](#)

IsPackaged

Indicates whether the product instance has applied packaging.

```
public bool IsPackaged { get; }
```

Property Value

[bool](#) ↗

Properties

Gets the list of properties associated with the product definition.

```
public IReadOnlyList<PropertyBase> Properties { get; }
```

Property Value

[IReadOnlyList](#)<[PropertyBase](#)>

Remarks

This property provides an unmodifiable list of properties associated with the underlying product definition. Each property represents a specific characteristic or behavior of the corresponding product.

Quality

Represents the quality level of the product instance.

```
public Quality Quality { get; }
```

Property Value

[Quality](#)

Remarks

Quality levels provide a measure of the product's grading, ranging from "Trash" to "Heavenly".

Class ProductManager

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Provides management over all products in the game.

```
public static class ProductManager
```

Inheritance

[object](#) ← ProductManager

Inherited Members

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

Properties

DiscoveredProducts

A list of product definitions discovered on this save.

```
public static ProductDefinition[] DiscoveredProducts { get; }
```

Property Value

[ProductDefinition](#)

Enum Quality

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents the quality levels for items.

```
public enum Quality
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Heavenly = 4

Represents "Heavenly" quality

Poor = 1

Represents "Poor" Quality

Premium = 3

Represents "Premium" quality

Standard = 2

Represents "Standard" Quality

Trash = 0

Represents "Trash" Quality

Remarks

This enumeration defines various quality tiers that items can belong to. Each tier represents a specific standard or grade, ranging from the lowest to the highest.

Class ShroomAppearanceSettings

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents the visual appearance settings for mushroom products, including colors and spot patterns.

```
public sealed class ShroomAppearanceSettings
```

Inheritance

[object](#) ← ShroomAppearanceSettings

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

DefaultPrimaryColor

The default primary color used for mushrooms.

```
public static Color32 DefaultPrimaryColor { get; }
```

Property Value

Color32

DefaultSecondaryColor

The default secondary color used for mushrooms.

```
public static Color32 DefaultSecondaryColor { get; }
```

Property Value

Color32

DefaultSpotsColor

The default spots color used for mushrooms.

```
public static Color32 DefaultSpotsColor { get; }
```

Property Value

Color32

HasSpots

Whether this mushroom has spots on its cap.

```
public bool HasSpots { get; }
```

Property Value

[bool](#) ↗

PrimaryColor

The primary color of the mushroom.

```
public Color32 PrimaryColor { get; }
```

PropertyValue

Color32

SecondaryColor

The secondary color of the mushroom.

```
public Color32 SecondaryColor { get; }
```

PropertyValue

Color32

SpotsColor

The color of the spots on the mushroom cap (if HasSpots is true).

```
public Color32 SpotsColor { get; }
```

PropertyValue

Color32

Methods

IsUninitialized()

Checks if the appearance settings are uninitialized (have default/clear colors).

```
public bool IsUninitialized()
```

Returns

[bool](#) ↗

True if the appearance settings are uninitialized, false otherwise.

Class ShroomDefinition

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents a specific type of mushroom (shroom) product definition.

```
public sealed class ShroomDefinition : ProductDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← [ProductDefinition](#) ← ShroomDefinition

Inherited Members

[ProductDefinition.Price](#) , [ProductDefinition.BasePrice](#) , [ProductDefinition.MarketValue](#) ,
[ProductDefinition.Icon](#) , [ProductDefinition.Properties](#) , [ProductDefinition.DrugTypes](#) ,
[ProductDefinition.DrugType](#) , [ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) ,
[ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) ,
[ItemDefinition.Category](#) , [ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) ,
[ItemDefinition.LabelDisplayColor](#) , [ItemDefinition.Keywords](#) , [ItemDefinition.GUID](#) , [object.GetType\(\)](#) ,
[object.ToString\(\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

AppearanceSettings

The appearance settings for this mushroom definition, including colors and visual properties.

```
public ShroomAppearanceSettings AppearanceSettings { get; }
```

Property Value

[ShroomAppearanceSettings](#)

BulkMaterial

The material used for rendering bulk mushrooms.

```
public Material BulkMaterial { get; }
```

Property Value

Material

EyeballMaterial

The material applied to character eyeballs when consuming mushrooms.

```
public Material EyeballMaterial { get; }
```

Property Value

Material

ShroomMaterial

The material used for rendering individual mushrooms.

```
public Material ShroomMaterial { get; }
```

Property Value

Material

Methods

CreateInstance(int)

Creates an instance of this mushroom product with the specified quantity.

```
public override ItemInstance CreateInstance(int quantity = 1)
```

Parameters

quantity [int](#)

The quantity of the product instance to create. Defaults to 1 if not specified.

Returns

[ItemInstance](#)

An [ItemInstance](#) representing the created mushroom product.

GetProperties()

Retrieves the list of properties associated with the mushroom product definition.

```
public List<PropertyBase> GetProperties()
```

Returns

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

A list of runtime-agnostic property wrappers that work on both Mono and IL2CPP.

Class ShroomInstance

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents an instance of a mushroom (shroom) product.

```
public sealed class ShroomInstance : ProductInstance
```

Inheritance

[object](#) ← [ItemInstance](#) ← [ProductInstance](#) ← ShroomInstance

Inherited Members

[ProductInstance.IsPackaged](#) , [ProductInstance.AppliedPackaging](#) , [ProductInstance.Quality](#) ,
[ProductInstance.Properties](#) , [ItemInstance.Quantity](#) , [ItemInstance.IsStackable](#) , [object.GetType\(\)](#) ,
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Remarks

Mushroom products can have psychedelic visual effects when consumed by players. This class extends `ProductInstance` to provide shroom-specific functionality.

Properties

Definition

Gets the shroom-specific definition for this instance.

```
public ShroomDefinition Definition { get; }
```

Property Value

Name

Gets the display name of the shroom instance. Automatically pluralizes "Shroom" to "Shrooms" when quantity is greater than 1.

```
public string Name { get; }
```

Property Value

[string](#)

Class WeedDefinition

Namespace: [S1API.Products](#)

Assembly: S1API.dll

Represents a specific type of weed product definition.

```
public class WeedDefinition : ProductDefinition
```

Inheritance

[object](#) ← [ItemDefinition](#) ← [ProductDefinition](#) ← WeedDefinition

Inherited Members

[ProductDefinition.Price](#) , [ProductDefinition.BasePrice](#) , [ProductDefinition.MarketValue](#) ,
[ProductDefinition.Icon](#) , [ProductDefinition.Properties](#) , [ProductDefinition.DrugTypes](#) ,
[ProductDefinition.DrugType](#) , [ItemDefinition.Equals\(object\)](#) , [ItemDefinition.GetHashCode\(\)](#) ,
[ItemDefinition.ID](#) , [ItemDefinition.Name](#) , [ItemDefinition.Description](#) , [ItemDefinition.StackLimit](#) ,
[ItemDefinition.Category](#) , [ItemDefinition.AvailableInDemo](#) , [ItemDefinition.LegalStatus](#) ,
[ItemDefinition.LabelDisplayColor](#) , [ItemDefinition.Keywords](#) , [ItemDefinition.GUID](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Methods

CreateInstance(int)

Creates an instance of the product with the specified quantity.

```
public override ItemInstance CreateInstance(int quantity = 1)
```

Parameters

quantity [int](#)

The quantity of the product to create. Defaults to 1 if not specified.

Returns

[ItemInstance](#)

An [ItemInstance](#) representing the created product instance with the specified quantity.

GetProperties()

Retrieves a list of properties associated with this weed definition.

```
public List<PropertyBase> GetProperties()
```

Returns

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

A list of runtime-agnostic property wrappers that work on both Mono and IL2CPP.

GetRawProperties()

DEPRECATED: Use GetProperties() which returns IProperty wrappers. Retrieves the raw Schedule One properties (may cause assembly reference issues in IL2CPP).

```
[Obsolete("Use GetProperties() instead for IL2CPP compatibility")]
public List<Effect> GetRawProperties()
```

Returns

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

A list of raw Schedule One properties.

Namespace S1API.Properties

Namespaces

[S1API.Properties.Interfaces](#)

[S1API.Properties.Tokens](#)

Classes

[ProductPropertyWrapper](#)

Runtime-agnostic wrapper for Schedule One product properties. Provides consistent access whether running on Mono or IL2CPP.

[Property](#)

Static accessors for common product properties. Avoids exposing game types while providing convenient tokens.

Namespace S1API.Properties.Interfaces

Classes

[PropertyBase](#)

Runtime-agnostic base class for product properties. Provides a consistent API regardless of whether running on Mono or IL2CPP. Uses concrete class instead of interface to avoid IL2CPP TypeLoadException issues.

Class PropertyBase

Namespace: [S1API.Properties.Interfaces](#)

Assembly: S1API.dll

Runtime-agnostic base class for product properties. Provides a consistent API regardless of whether running on Mono or IL2CPP. Uses concrete class instead of interface to avoid IL2CPP TypeLoadException issues.

```
public abstract class PropertyBase
```

Inheritance

[object](#) ← PropertyBase

Derived

[ProductPropertyWrapper](#), [PropertyToken](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

PropertyBase()

```
protected PropertyBase()
```

Properties

Addictiveness

How addictive this property makes the product (0-1).

```
public abstract float Addictiveness { get; }
```

Property Value

[float](#) ↗

Description

The description of what this property does.

```
public abstract string Description { get; }
```

Property Value

[string](#) ↗

ID

The unique identifier for this property.

```
public abstract string ID { get; }
```

Property Value

[string](#) ↗

Name

The display name of this property.

```
public abstract string Name { get; }
```

Property Value

[string ↗](#)

Tier

The tier/level of this property (1-5).

```
public abstract int Tier { get; }
```

Property Value

[int ↗](#)

name

The Unity name of this property (lowercase for compatibility). This corresponds to the Unity ScriptableObject.name property.

```
public abstract string name { get; }
```

Property Value

[string ↗](#)

Namespace S1API.Properties.Tokens

Classes

[AntiGravity](#)

Token for the AntiGravity property.

[Athletic](#)

Token for the Athletic property.

[Balding](#)

Token for the Balding property.

[BrightEyed](#)

Token for the BrightEyed property.

[Calming](#)

Token for the Calming property.

[CalorieDense](#)

Token for the CalorieDense property.

[Cyclopean](#)

Token for the Cyclopean property.

[Disorienting](#)

Token for the Disorienting property.

[Electrifying](#)

Token for the Electrifying property.

[Energizing](#)

Token for the Energizing property.

[Euphoric](#)

Token for the Euphoric property.

[Explosive](#)

Token for the Explosive property.

[Focused](#)

Token for the Focused property.

[Foggy](#)

Token for the Foggy property.

[Gingeritis](#)

Token for the Gingeritis property.

[Glowie](#)

Token for the Glowie property.

[Jennerising](#)

Token for the Jennerising property.

[Laxative](#)

Token for the Laxative property.

[Lethal](#)

Token for the Lethal property.

[LongFaced](#)

Token for the LongFaced property.

[Munchies](#)

Token for the Munchies property.

[Paranoia](#)

Token for the Paranoia property.

[PropertyToken](#)

Base token that represents a product property without referencing runtime game types.

[Refreshing](#)

Token for the Refreshing property.

[Schizophrenic](#)

Token for the Schizophrenic property.

[Sedating](#)

Token for the Sedating property.

[Seizure](#)

Token for the Seizure property.

[Shrinking](#)

Token for the Shrinking property.

[Slippery](#)

Token for the Slippery property.

[Smelly](#)

Token for the Smelly property.

[Sneaky](#)

Token for the Sneaky property.

[Spicy](#)

Token for the Spicy property.

[ThoughtProvoking](#)

Token for the ThoughtProvoking property.

[Toxic](#)

Token for the Toxic property.

[TropicThunder](#)

Token for the TropicThunder property.

[Zombifying](#)

Token for the Zombifying property.

Class AntiGravity

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the AntiGravity property.

```
public sealed class AntiGravity : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← AntiGravity

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

AntiGravity()

```
public AntiGravity()
```

Class Athletic

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Athletic property.

```
public sealed class Athletic : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Athletic

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Athletic()

```
public Athletic()
```

Class Balding

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Balding property.

```
public sealed class Balding : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Balding

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Balding()

```
public Balding()
```

Class BrightEyed

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the BrightEyed property.

```
public sealed class BrightEyed : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← BrightEyed

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BrightEyed()

```
public BrightEyed()
```

Class Calming

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Calming property.

```
public sealed class Calming : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Calming

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Calming()

```
public Calming()
```

Class CalorieDense

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the CalorieDense property.

```
public sealed class CalorieDense : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← CalorieDense

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

CalorieDense()

```
public CalorieDense()
```

Class Cyclopean

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Cyclopean property.

```
public sealed class Cyclopean : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Cyclopean

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Cyclopean()

```
public Cyclopean()
```

Class Disorienting

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Disorienting property.

```
public sealed class Disorienting : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Disorienting

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Disorienting()

```
public Disorienting()
```

Class Electrifying

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Electrifying property.

```
public sealed class Electrifying : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Electrifying

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Electrifying()

```
public Electrifying()
```

Class Energizing

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Energizing property.

```
public sealed class Energizing : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Energizing

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Energizing()

```
public Energizing()
```

Class Euphoric

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Euphoric property.

```
public sealed class Euphoric : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Euphoric

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Euphoric()

```
public Euphoric()
```

Class Explosive

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Explosive property.

```
public sealed class Explosive : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Explosive

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Explosive()

```
public Explosive()
```

Class Focused

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Focused property.

```
public sealed class Focused : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Focused

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Focused()

```
public Focused()
```

Class Foggy

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Foggy property.

```
public sealed class Foggy : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Foggy

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Foggy()

```
public Foggy()
```

Class Gingeritis

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Gingeritis property.

```
public sealed class Gingeritis : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← [Gingeritis](#)

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

[Gingeritis\(\)](#)

```
public Gingeritis()
```

Class Glowie

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Glowie property.

```
public sealed class Glowie : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Glowie

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Glowie()

```
public Glowie()
```

Class Jennerising

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Jennerising property.

```
public sealed class Jennerising : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Jennerising

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Jennerising()

```
public Jennerising()
```

Class Laxative

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Laxative property.

```
public sealed class Laxative : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Laxative

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Laxative()

```
public Laxative()
```

Class Lethal

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Lethal property.

```
public sealed class Lethal : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Lethal

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Lethal()

```
public Lethal()
```

Class LongFaced

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the LongFaced property.

```
public sealed class LongFaced : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← LongFaced

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

LongFaced()

```
public LongFaced()
```

Class Munchies

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Munchies property.

```
public sealed class Munchies : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Munchies

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Munchies()

```
public Munchies()
```

Class Paranoia

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Paranoia property.

```
public sealed class Paranoia : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Paranoia

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Paranoia()

```
public Paranoia()
```

Class PropertyToken

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Base token that represents a product property without referencing runtime game types.

```
public abstract class PropertyToken : PropertyBase
```

Inheritance

[object](#) ← [PropertyBase](#) ← PropertyToken

Derived

[AntiGravity](#), [Athletic](#), [Balding](#), [BrightEyed](#), [Calming](#), [CalorieDense](#), [Cyclopean](#), [Disorienting](#), [Electrifying](#), [Energizing](#), [Euphoric](#), [Explosive](#), [Focused](#), [Foggy](#), [Gingeritis](#), [Glowie](#), [Jennerising](#), [Laxative](#), [Lethal](#), [LongFaced](#), [Munchies](#), [Paranoia](#), [Refreshing](#), [Schizophrenic](#), [Sedating](#), [Seizure](#), [Shrinking](#), [Slippery](#), [Smelly](#), [Sneaky](#), [Spicy](#), [ThoughtProvoking](#), [Toxic](#), [TropicThunder](#), [Zombifying](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

PropertyToken(string, string, string, string, int, float)

```
protected PropertyToken(string id, string unityName = null, string displayName = null, string description = null, int tier = 1, float addictiveness = 0)
```

Parameters

id [string](#)

unityName [string](#)

displayName [string](#)

description [string](#)

tier [int](#)

addictiveness [float](#)

Properties

Addictiveness

How addictive this property makes the product (0-1).

```
public override float Addictiveness { get; }
```

Property Value

[float](#)

Description

The description of what this property does.

```
public override string Description { get; }
```

Property Value

[string](#)

ID

The unique identifier for this property.

```
public override string ID { get; }
```

Property Value

[string](#) ↗

Name

The display name of this property.

```
public override string Name { get; }
```

Property Value

[string](#) ↗

Tier

The tier/level of this property (1-5).

```
public override int Tier { get; }
```

Property Value

[int](#) ↗

name

The Unity name of this property (lowercase for compatibility). This corresponds to the Unity ScriptableObject.name property.

```
public override string name { get; }
```

Property Value

[string ↗](#)

Class Refreshing

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Refreshing property.

```
public sealed class Refreshing : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Refreshing

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Refreshing()

```
public Refreshing()
```

Class Schizophrenic

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Schizophrenic property.

```
public sealed class Schizophrenic : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Schizophrenic

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Schizophrenic()

```
public Schizophrenic()
```

Class Sedating

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Sedating property.

```
public sealed class Sedating : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Sedating

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Sedating()

```
public Sedating()
```

Class Seizure

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Seizure property.

```
public sealed class Seizure : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Seizure

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Seizure()

```
public Seizure()
```

Class Shrinking

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Shrinking property.

```
public sealed class Shrinking : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Shrinking

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Shrinking()

```
public Shrinking()
```

Class Slippery

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Slippery property.

```
public sealed class Slippery : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Slippery

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Slippery()

```
public Slippery()
```

Class Smelly

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Smelly property.

```
public sealed class Smelly : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Smelly

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Smelly()

```
public Smelly()
```

Class Sneaky

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Sneaky property.

```
public sealed class Sneaky : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← [Sneaky](#)

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

[Sneaky\(\)](#)

```
public Sneaky()
```

Class Spicy

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Spicy property.

```
public sealed class Spicy : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Spicy

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Spicy()

```
public Spicy()
```

Class ThoughtProvoking

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the ThoughtProvoking property.

```
public sealed class ThoughtProvoking : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← ThoughtProvoking

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

ThoughtProvoking()

```
public ThoughtProvoking()
```

Class Toxic

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Toxic property.

```
public sealed class Toxic : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Toxic

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Toxic()

```
public Toxic()
```

Class TropicThunder

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the TropicThunder property.

```
public sealed class TropicThunder : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← TropicThunder

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

TropicThunder()

```
public TropicThunder()
```

Class Zombifying

Namespace: [S1API.Properties.Tokens](#)

Assembly: S1API.dll

Token for the Zombifying property.

```
public sealed class Zombifying : PropertyToken
```

Inheritance

[object](#) ← [PropertyBase](#) ← [PropertyToken](#) ← Zombifying

Inherited Members

[PropertyToken.ID](#) , [PropertyToken.Name](#) , [PropertyToken.name](#) , [PropertyToken.Description](#) ,
[PropertyToken.Tier](#) , [PropertyToken.Addictiveness](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) ,
[object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Zombifying()

```
public Zombifying()
```

Class ProductPropertyWrapper

Namespace: [S1API.Properties](#)

Assembly: S1API.dll

Runtime-agnostic wrapper for Schedule One product properties. Provides consistent access whether running on Mono or IL2CPP.

```
public class ProductPropertyWrapper : PropertyBase
```

Inheritance

[object](#) ← [PropertyBase](#) ← ProductPropertyWrapper

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Addictiveness

How addictive this property makes the product (0-1).

```
public override float Addictiveness { get; }
```

Property Value

[float](#)

Description

The description of what this property does.

```
public override string Description { get; }
```

Property Value

[string](#)

ID

The unique identifier for this property.

```
public override string ID { get; }
```

Property Value

[string](#)

Name

The display name of this property.

```
public override string Name { get; }
```

Property Value

[string](#)

Tier

The tier/level of this property (1-5).

```
public override int Tier { get; }
```

Property Value

[int](#)

name

The Unity name of this property. This corresponds to the Unity ScriptableObject.name property.

```
public override string name { get; }
```

Property Value

[string](#)

Class Property

Namespace: [S1API.Properties](#)

Assembly: S1API.dll

Static accessors for common product properties. Avoids exposing game types while providing convenient tokens.

```
public static class Property
```

Inheritance

[object](#) ← Property

Inherited Members

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

Fields

AntiGravity

```
public static readonly PropertyBase AntiGravity
```

Field Value

[PropertyBase](#)

Athletic

```
public static readonly PropertyBase Athletic
```

Field Value

[PropertyBase](#)

Balding

```
public static readonly PropertyBase Balding
```

Field Value

[PropertyBase](#)

BrightEyed

```
public static readonly PropertyBase BrightEyed
```

Field Value

[PropertyBase](#)

Calming

```
public static readonly PropertyBase Calming
```

Field Value

[PropertyBase](#)

CalorieDense

```
public static readonly PropertyBase CalorieDense
```

Field Value

[PropertyBase](#)

Cyclopean

```
public static readonly PropertyBase Cyclopean
```

Field Value

[PropertyBase](#)

Disorienting

```
public static readonly PropertyBase Disorienting
```

Field Value

[PropertyBase](#)

Electrifying

```
public static readonly PropertyBase Electrifying
```

Field Value

[PropertyBase](#)

Energizing

```
public static readonly PropertyBase Energizing
```

Field Value

[PropertyBase](#)

Euphoric

```
public static readonly PropertyBase Euphoric
```

Field Value

[PropertyBase](#)

Explosive

```
public static readonly PropertyBase Explosive
```

Field Value

[PropertyBase](#)

Focused

```
public static readonly PropertyBase Focused
```

Field Value

[PropertyBase](#)

Foggy

```
public static readonly PropertyBase Foggy
```

Field Value

[PropertyBase](#)

Gingeritis

```
public static readonly PropertyBase Gingeritis
```

Field Value

[PropertyBase](#)

Glowie

```
public static readonly PropertyBase Glowie
```

Field Value

[PropertyBase](#)

Jennerising

```
public static readonly PropertyBase Jennerising
```

Field Value

[PropertyBase](#)

Laxative

```
public static readonly PropertyBase Laxative
```

Field Value

[PropertyBase](#)

Lethal

```
public static readonly PropertyBase Lethal
```

Field Value

[PropertyBase](#)

LongFaced

```
public static readonly PropertyBase LongFaced
```

Field Value

[PropertyBase](#)

Munchies

```
public static readonly PropertyBase Munchies
```

Field Value

[PropertyBase](#)

Paranoia

```
public static readonly PropertyBase Paranoia
```

Field Value

[PropertyBase](#)

Refreshing

```
public static readonly PropertyBase Refreshing
```

Field Value

[PropertyBase](#)

Schizophrenic

```
public static readonly PropertyBase Schizophrenic
```

Field Value

[PropertyBase](#)

Sedating

```
public static readonly PropertyBase Sedating
```

Field Value

[PropertyBase](#)

Seizure

```
public static readonly PropertyBase Seizure
```

Field Value

[PropertyBase](#)

Shrinking

```
public static readonly PropertyBase Shrinking
```

Field Value

[PropertyBase](#)

Slippery

```
public static readonly PropertyBase Slippery
```

Field Value

[PropertyBase](#)

Smelly

```
public static readonly PropertyBase Smelly
```

Field Value

[PropertyBase](#)

Sneaky

```
public static readonly PropertyBase Sneaky
```

Field Value

[PropertyBase](#)

Spicy

```
public static readonly PropertyBase Spicy
```

Field Value

[PropertyBase](#)

ThoughtProvoking

```
public static readonly PropertyBase ThoughtProvoking
```

Field Value

[PropertyBase](#)

Toxic

```
public static readonly PropertyBase Toxic
```

Field Value

[PropertyBase](#)

TropicThunder

```
public static readonly PropertyBase TropicThunder
```

Field Value

[PropertyBase](#)

Zombifying

```
public static readonly PropertyBase Zombifying
```

Field Value

[PropertyBase](#)

Namespace S1API.Property

Classes

[BaseProperty](#)

Represents an abstract base class for properties in the system.

[BusinessManager](#)

Provides methods for managing and retrieving business property data within the application.

[BusinessWrapper](#)

Represents a wrapper class for handling business properties derived from the ScheduleOne.Property.Business class. Provides an abstraction for interacting with business property details and operations in Unity.

[LaunderingOperation](#)

Represents a laundering operation associated with a business property.

[PropertyManager](#)

Provides methods for managing and retrieving property data within the application.

[PropertyWrapper](#)

Provides an abstraction for interacting with in game player properties.

Class BaseProperty

Namespace: [S1API.Property](#)

Assembly: S1API.dll

Represents an abstract base class for properties in the system.

```
public abstract class BaseProperty
```

Inheritance

[object](#) ← BaseProperty

Derived

[PropertyWrapper](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BaseProperty()

```
protected BaseProperty()
```

Properties

EmployeeCapacity

Gets or sets the maximum number of employees that can be assigned to the property.

```
public abstract int EmployeeCapacity { get; set; }
```

Property Value

[int ↗](#)

Remarks

This property represents the capacity for employees within a given property. Modifying this value impacts the operations and resource management of the property. Suitable for scenarios where resource allocation and workforce planning are essential.

IsOwned

```
public abstract bool IsOwned { get; }
```

Property Value

[bool ↗](#)

Price

Represents the cost or monetary value associated with a property.

```
public abstract float Price { get; set; }
```

Property Value

[float ↗](#)

A [float](#) representing the monetary price of the property.

Remarks

The **Price** property is a floating-point value that indicates the price for the property. It provides a read-only mechanism to access this value. This value is essential in determining the economic aspect of the property.

PropertyCode

Gets the unique code that identifies the property.

```
public abstract string PropertyCode { get; }
```

Property Value

[string](#)

Remarks

This code is typically used to differentiate between various properties within the system. It is unique to each property and can be leveraged in operations like identification, filtering, or querying.

PropertyName

Gets the name of the property.

```
public abstract stringPropertyName { get; }
```

Property Value

[string](#)

Remarks

This property represents the name or title of the property entity. It retrieves the value associated with the property from the underlying system or data structure.

Methods

IsPointInside(Vector3)

Determines whether a specified point lies within the boundary of the property.

```
public abstract bool IsPointInside(Vector3 point)
```

Parameters

point Vector3

The point to check, specified as a Vector3 coordinate.

Returns

[bool](#)

true if the point is within the property's boundary; otherwise, false.

SetOwned()

Marks the property as owned. This method updates the ownership status of the property by interacting with the underlying property implementation. Typically used to signify that the property has been acquired or purchased.

```
public abstract void SetOwned()
```

Class BusinessManager

Namespace: [S1API.Property](#)

Assembly: S1API.dll

Provides methods for managing and retrieving business property data within the application.

```
public static class BusinessManager
```

Inheritance

[object](#) ← BusinessManager

Inherited Members

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

Methods

FindBusinessByName(string)

Finds a business property with the given name from the list of available business properties.

```
public static BusinessWrapper? FindBusinessByName(string name)
```

Parameters

name [string](#)

The name of the business property to search for.

Returns

[BusinessWrapper](#)

A [BusinessWrapper](#) representing the business property with the specified name if found; otherwise, null.

GetAllBusinesses()

Retrieves a list of all business properties available.

```
public static List<BusinessWrapper> GetAllBusinesses()
```

Returns

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

A list of [BusinessWrapper](#) objects representing all available business properties.

GetOwnedBusinesses()

Retrieves a list of all business properties that are currently owned.

```
public static List<BusinessWrapper> GetOwnedBusinesses()
```

Returns

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

A list of [BusinessWrapper](#) objects representing the owned business properties.

Class BusinessWrapper

Namespace: [S1API.Property](#)

Assembly: S1API.dll

Represents a wrapper class for handling business properties derived from the ScheduleOne.Property.Business class. Provides an abstraction for interacting with business property details and operations in Unity.

```
public class BusinessWrapper : PropertyWrapper
```

Inheritance

[object](#) ← [BaseProperty](#) ← [PropertyWrapper](#) ← BusinessWrapper

Inherited Members

[PropertyWrapper.SetOwned\(\)](#) , [PropertyWrapper.IsPointInside\(Vector3\)](#) ,
[PropertyWrapper.GetEmployeeIdlePointPosition\(int\)](#) , [PropertyWrapper.GetUnassignedBedCount\(\)](#) ,
[PropertyWrapper.PropertyName](#) , [PropertyWrapper.PropertyCode](#) , [PropertyWrapper.Price](#) ,
[PropertyWrapper.IsOwned](#) , [PropertyWrapper.EmployeeCapacity](#) ,
[PropertyWrapper.ExteriorSpawnPosition](#) , [PropertyWrapper.InteriorSpawnPosition](#) ,
[PropertyWrapper>LoadingDockCount](#) , [PropertyWrapper.DefaultRotation](#) ,
[PropertyWrapper.AvailableInDemo](#) , [PropertyWrapper.IsContentCulled](#) , [PropertyWrapper.EmployeeCount](#) ,
[PropertyWrapper.BuildableItemCount](#) , [PropertyWrapper.NPCSpawnPosition](#) ,
[PropertyWrapper.EmployeeIdlePointCount](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) ,
[object.ToString\(\)](#) , [object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

BusinessWrapper(Business)

A wrapper class that extends the functionality of [PropertyWrapper](#) and acts as a bridge to interact with an inner business property implementation from the II2CppScheduleOne.Property namespace.

```
public BusinessWrapper(Business business)
```

Parameters

business Business

Properties

AppliedLaunderLimit

Gets the applied launder limit for the business.

```
public float AppliedLaunderLimit { get; }
```

Property Value

[float](#)

CurrentLaunderTotal

Gets the current total amount of money being laundered by the business.

```
public float CurrentLaunderTotal { get; }
```

Property Value

[float](#)

IsAtLaunderingCapacity

Gets a value indicating whether the business has reached its laundering capacity.

```
public bool IsAtLaunderingCapacity { get; }
```

Property Value

[bool](#) ↗

LaunderCapacity

Gets or sets the laundering capacity of the business.

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

Property Value

[float](#) ↗

LaunderingCapacityUsagePercent

Gets the percentage of laundering capacity currently in use. Returns a value between 0 and 1, where 1 represents 100% capacity.

```
public float LaunderingCapacityUsagePercent { get; }
```

Property Value

[float](#) ↗

LaunderingOperationCount

Gets the number of active laundering operations currently in progress.

```
public int LaunderingOperationCount { get; }
```

Property Value

[int](#)

LaunderingOperations

Gets or sets the laundering operations associated with the business.

```
public List<LaunderingOperation> LaunderingOperations { get; }
```

Property Value

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

Methods

AddLaunderingOperation(float, int)

Adds a new laundering operation to the business with the specified amount and minutes since started.
Added LaunderingOperation will be reflected in the [LaunderingOperations](#) list after the client receives it.

```
public void AddLaunderingOperation(float amount, int minutesSinceStarted)
```

Parameters

amount [float](#)

The amount of money to be laundered.

minutesSinceStarted [int](#)

The number of minutes since the laundering operation started.

Class LaunderingOperation

Namespace: [S1API.Property](#)

Assembly: S1API.dll

Represents a laundering operation associated with a business property.

```
public class LaunderingOperation
```

Inheritance

[object](#) ← LaunderingOperation

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

LaunderingOperation(BusinessWrapper, float, int, int)

Initializes a new instance of the [LaunderingOperation](#) class with the specified business, amount, minutes since started, and optional completion time.

```
public LaunderingOperation(BusinessWrapper business, float amount, int
minutesSinceStarted, int completionTimeInMinutes = 1140)
```

Parameters

business [BusinessWrapper](#)

The business associated with the laundering operation.

amount [float](#)

The amount of money to be laundered.

minutesSinceStarted [int](#)

The number of minutes since the laundering operation started.

completionTimeInMinutes [int](#)

The total time in minutes required to complete the laundering operation. Default is 1140 minutes.

LaunderingOperation(LaunderingOperation)

A wrapper class that acts as a bridge to interact with an inner laundering operation implementation from the II2CppScheduleOne.Property namespace.

```
public LaunderingOperation(LaunderingOperation launderingOperation)
```

Parameters

launderingOperation LaunderingOperation

Properties

Amount

Gets or sets the amount of money to be laundered in this operation.

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

Property Value

[float](#)

Business

Gets or sets the business associated with the laundering operation.

```
public BusinessWrapper Business { get; set; }
```

Property Value

[BusinessWrapper](#)

CompletionTimeInMinutes

Gets or sets the total time in minutes required to complete the laundering operation.

```
public int CompletionTimeInMinutes { get; set; }
```

Property Value

[int](#)

MinutesSinceStarted

Gets or sets the number of minutes that have passed since the laundering operation started.

```
public int MinutesSinceStarted { get; set; }
```

Property Value

[int](#)

Class PropertyManager

Namespace: [S1API.Property](#)

Assembly: S1API.dll

Provides methods for managing and retrieving property data within the application.

```
public static class PropertyManager
```

Inheritance

[object](#) ← PropertyManager

Inherited Members

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

Methods

FindPropertyByName(string)

Finds a property with the given name from the list of available properties.

```
public static PropertyWrapper FindPropertyByName(string name)
```

Parameters

name [string](#)

The name of the property to search for.

Returns

[PropertyWrapper](#)

A [PropertyWrapper](#) representing the property with the specified name if found; otherwise, null.

GetAllProperties()

Retrieves a list of all properties available.

```
public static List<PropertyWrapper> GetAllProperties()
```

Returns

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

A list of [PropertyWrapper](#) objects representing all available properties.

GetOwnedProperties()

```
public static List<PropertyWrapper> GetOwnedProperties()
```

Returns

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

A list of PropertyWrapper objects representing the owned properties.

Class PropertyWrapper

Namespace: [S1API.Property](#)

Assembly: S1API.dll

Provides an abstraction for interacting with in game player properties.

```
public class PropertyWrapper : BaseProperty
```

Inheritance

[object](#) ← [BaseProperty](#) ← PropertyWrapper

Derived

[BusinessWrapper](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

PropertyWrapper(Property)

A wrapper class that extends the functionality of [BaseProperty](#) and acts as a bridge to interact with an inner property implementation from the ScheduleOne.Property namespace.

```
public PropertyWrapper(Property property)
```

Parameters

property Property

Properties

AvailableInDemo

Gets a value indicating whether the property is available in the demo version of the game.

```
public bool AvailableInDemo { get; }
```

Property Value

[bool](#) ↗

BuildableItemCount

Gets the number of buildable items currently placed in this property.

```
public int BuildableItemCount { get; }
```

Property Value

[int](#) ↗

DefaultRotation

Gets the default rotation value for the property.

```
public float DefaultRotation { get; }
```

Property Value

[float](#) ↗

EmployeeCapacity

Represents the maximum number of employees that can be allocated to the property. This property is both readable and writable, allowing for dynamic configuration of employee capacity based on the property's current requirements or constraints.

```
public override int EmployeeCapacity { get; set; }
```

Property Value

[int↗](#)

EmployeeCount

Gets the number of employees currently assigned to this property.

```
public int EmployeeCount { get; }
```

Property Value

[int↗](#)

EmployeeIdlePointCount

Gets the number of employee idle points configured for this property.

```
public int EmployeeIdlePointCount { get; }
```

Property Value

[int↗](#)

ExteriorSpawnPosition

Gets the exterior spawn point position of the property. This is typically used for spawning outside the property.

```
public Vector3 ExteriorSpawnPosition { get; }
```

Property Value

Vector3

InteriorSpawnPosition

Gets the interior spawn point position of the property. This is typically used for spawning inside the property.

```
public Vector3 InteriorSpawnPosition { get; }
```

Property Value

Vector3

IsContentCulled

Gets or sets a value indicating whether the property's content is currently culled. Content culling is used to optimize performance by hiding property contents when far away.

```
public bool IsContentCulled { get; set; }
```

Property Value

[bool](#) ↗

IsOwned

Gets a value indicating whether the property is currently owned.

```
public override bool IsOwned { get; }
```

Property Value

[bool](#)

Remarks

This property reflects the ownership status of the property. Returns true if the property is owned and false otherwise. The ownership status is based on the internal state of the wrapped property implementation.

LoadingDockCount

Gets the number of loading docks available at this property.

```
public int LoadingDockCount { get; }
```

Property Value

[int](#)

NPCSpawnPosition

Gets the position of the NPC spawn point for this property. Returns Vector3.zero if no NPC spawn point is configured.

```
public Vector3 NPCSpawnPosition { get; }
```

Property Value

Price

Gets the price of the property.

```
public override float Price { get; set; }
```

Property Value

[float](#)

Remarks

The price represents a floating-point value that denotes the monetary value or cost associated with the property. This property is both readable and writable, allowing for dynamic adjustments to the property's price.

PropertyCode

Gets the unique code representing this property. This code serves as an identifier for distinguishing the property in the system and is typically defined in the internal implementation of the property.

```
public override string PropertyCode { get; }
```

Property Value

[string](#)

PropertyName

Gets the name of the property. Represents the underlying property name as defined by its implementation.

```
public override string PropertyName { get; }
```

Property Value

[string](#)

Methods

GetEmployeeIdlePointPosition(int)

Gets the position of a specific employee idle point by index.

```
public Vector3 GetEmployeeIdlePointPosition(int index)
```

Parameters

[index](#) [int](#)

The zero-based index of the idle point.

Returns

Vector3

The position of the idle point, or Vector3.zero if the index is invalid.

GetUnassignedBedCount()

Gets the number of unassigned beds currently available in this property.

```
public int GetUnassignedBedCount()
```

Returns

[int](#)

The count of beds that do not have an assigned employee.

IsPointInside(Vector3)

Determines whether a specified point lies within the boundary of the property.

```
public override bool IsPointInside(Vector3 point)
```

Parameters

point Vector3

The point to check, specified as a Vector3 coordinate.

Returns

[bool](#)

true if the point is within the property's boundary; otherwise, false.

SetOwned()

Marks the property as owned within the PropertyWrapper implementation. Updates the ownership status by delegating the operation to the underlying ScheduleOne.Property.Property instance. This is typically used to signify that the property has been acquired or purchased.

```
public override void SetOwned()
```

Namespace S1API.Quests

Namespaces

[S1API.Quests.Constants](#)

[S1API.Quests.Identifiers](#)

Classes

[Quest](#)

An abstract class intended to be derived from for creating custom quests in the game.

[QuestData](#)

Generic data applied for all custom quests.

[QuestEntry](#)

Represents a quest entry on a quest. These are the individual `tasks` on a quest.

[QuestManager](#)

Provides management of quests across the game.

[QuestWrapper](#)

Wrapper for accessing quests that can handle both custom mod quests and base game quests.

Provides unified access to quest events and properties.

Namespace S1API.Quests.Constants

Enums

[QuestAction](#)

A wrapper around EQuestAction

[QuestState](#)

Represents all states a quest can be. Applicable to quest entries as well.

Enum QuestAction

Namespace: [S1API.Quests.Constants](#)

Assembly: S1API.dll

A wrapper around EQuestAction

```
public enum QuestAction
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Begin = 0

Begin the quest

Cancel = 4

Cancel the quest

Expire = 3

Expire the quest

Fail = 2

Fail the quest

Success = 1

Succeed (complete) the quest

Enum QuestState

Namespace: [S1API.Quests.Constants](#)

Assembly: S1API.dll

Represents all states a quest can be. Applicable to quest entries as well.

```
public enum QuestState
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Active = 1

Represents a quest / quest entry that has been started but not ended.

Cancelled = 5

Represents a quest / quest entry that has been cancelled.

Completed = 2

Represents a quest / quest entry that has been completed successfully by the player.

Expired = 4

Represents a quest / quest entry that has been expired.

Failed = 3

Represents a quest / quest entry that has been failed by the player.

Inactive = 0

Represents a quest / quest entry that has not been started yet.

Namespace S1API.Quests.Identifiers

Classes

Botanists

Identifier for the base-game quest "Botanists" (Quest_Botanists). Modders can use [Get<T>\(\)](#) to resolve it.

Chemists

Identifier for the base-game quest "Chemists" (Quest_Chemists). Modders can use [Get<T>\(\)](#) to resolve it.

CleanCash

Identifier for the base-game quest "Clean Cash" (Quest_CleanCash). Modders can use [Get<T>\(\)](#) to resolve it.

Cleaners

Identifier for the base-game quest "Cleaners" (Quest_Cleaners). Modders can use [Get<T>\(\)](#) to resolve it.

DealForCartel

Identifier for the base-game quest "Deal for the Benzies Family" (Quest_DealForCartel). Modders can use [Get<T>\(\)](#) to resolve it.

DefeatCartel

Identifier for the base-game quest "Finishing the Job" (Quest_DefeatCartel). Modders can use [Get<T>\(\)](#) to resolve it.

DodgyDealing

Identifier for the base-game quest "Dodgy Dealing" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

GearingUp

Identifier for the base-game quest "Gearing Up" (Quest_GearingUp). Modders can use [Get<T>\(\)](#) to resolve it.

GettingStarted

Identifier for the base-game quest "Getting Started" (Quest_GettingStarted). Modders can use [Get<T>\(\)](#) to resolve it.

KeepingItFresh

Identifier for the base-game quest "Keeping it Fresh" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

MakingTheRounds

Identifier for the base-game quest "Making the Rounds" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

[MixingMania](#)

Identifier for the base-game quest "Mixing Mania" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

[MoneyManagement](#)

Identifier for the base-game quest "Money Management" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

[MovingUp](#)

Identifier for the base-game quest "Moving Up" (Quest_MovingUp). Modders can use [Get<T>\(\)](#) to resolve it.

[NeedingTheGreen](#)

Identifier for the base-game quest "Needin' the Green" (Quest_NeedingTheGreen). Modders can use [Get<T>\(\)](#) to resolve it.

[OnTheGrind](#)

Identifier for the base-game quest "On the Grind" (Quest_OnTheGrind). Modders can use [Get<T>\(\)](#) to resolve it.

[Packagers](#)

Identifier for the base-game quest "Handlers" (Quest_Packagers). Modders can use [Get<T>\(\)](#) to resolve it.

[Packin](#)

Identifier for the base-game quest "Packin" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

[QuestNameAttribute](#)

Annotate quest identifier types with the display title of the quest. Usage: [QuestName("Finishing the Job")]

[UnfavourableAgreements](#)

Identifier for the base-game quest "Unfavourable Agreements" (Quest_UnfavourableAgreements). Modders can use [Get<T>\(\)](#) to resolve it.

[Warehouse](#)

Identifier for the base-game quest "Wretched Hive of Scum and Villainy" (Quest_Warehouse). Modders can use [Get<T>\(\)](#) to resolve it.

[WeNeedToCook](#)

Identifier for the base-game quest "We Need To Cook" (Quest_WeNeedToCook). Modders can use [Get<T>\(\)](#) to resolve it.

[WelcomeToHylandPoint](#)

Identifier for the base-game quest "Welcome to Hyland Point" (Quest_WelcomeToHylandPoint).
Modders can use [Get<T>\(\)](#) to resolve it.

Interfaces

[IQuestIdentifier](#)

Marker interface for quest identifier types used with QuestManager.Get<T>(). Implement empty classes like 'public sealed class DefeatCartel : IQuestIdentifier {}' and optionally annotate with [QuestName("Finishing the Job")].

Class Botanists

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Botanists" (Quest_Botanists). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Botanists")]
public sealed class Botanists : IQuestIdentifier
```

Inheritance

[object](#) ← Botanists

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Botanists()

```
public Botanists()
```

Class Chemists

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Chemists" (Quest_Chemists). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Chemists")]
public sealed class Chemists : IQuestIdentifier
```

Inheritance

[object](#) ← Chemists

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

Chemists()

```
public Chemists()
```

Class CleanCash

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Clean Cash" (Quest_CleanCash). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Clean Cash")]
public sealed class CleanCash : IQuestIdentifier
```

Inheritance

[object](#) ← CleanCash

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

CleanCash()

```
public CleanCash()
```

Class Cleaners

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Cleaners" (Quest_Cleaners). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Cleaners")]
public sealed class Cleaners : IQuestIdentifier
```

Inheritance

[object](#) ← Cleaners

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Cleaners()

```
public Cleaners()
```

Class DealForCartel

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Deal for the Benzies Family" (Quest_DealForCartel). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Deal for the Benzies Family")]
public sealed class DealForCartel : IQuestIdentifier
```

Inheritance

[object](#) ← DealForCartel

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DealForCartel()

```
public DealForCartel()
```

Class DefeatCartel

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Finishing the Job" (Quest_DefeatCartel). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Finishing the Job")]
public sealed class DefeatCartel : IQuestIdentifier
```

Inheritance

[object](#) ← DefeatCartel

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DefeatCartel()

```
public DefeatCartel()
```

Class DodgyDealing

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Dodgy Dealing" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Dodgy Dealing")]
public sealed class DodgyDealing : IQuestIdentifier
```

Inheritance

[object](#) ← DodgyDealing

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

DodgyDealing()

```
public DodgyDealing()
```

Class GearingUp

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Gearing Up" (Quest_GearingUp). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Gearing Up")]
public sealed class GearingUp : IQuestIdentifier
```

Inheritance

[object](#) ← GearingUp

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

GearingUp()

```
public GearingUp()
```

Class GettingStarted

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Getting Started" (Quest_GettingStarted). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Getting Started")]
public sealed class GettingStarted : IQuestIdentifier
```

Inheritance

[object](#) ← GettingStarted

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

GettingStarted()

```
public GettingStarted()
```

Interface IQuestIdentifier

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Marker interface for quest identifier types used with QuestManager.Get<T>(). Implement empty classes like 'public sealed class DefeatCartel : IQuestIdentifier {}' and optionally annotate with [QuestName("Finishing the Job")].

```
public interface IQuestIdentifier
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Class KeepingItFresh

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Keeping it Fresh" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Keeping it Fresh")]
public sealed class KeepingItFresh : IQuestIdentifier
```

Inheritance

[object](#) ← KeepingItFresh

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

KeepingItFresh()

```
public KeepingItFresh()
```

Class MakingTheCounts

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Making the Rounds" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Making the Rounds")]
public sealed class MakingTheCounts : IQuestIdentifier
```

Inheritance

[object](#) ← MakingTheCounts

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MakingTheCounts()

```
public MakingTheCounts()
```

Class MixingMania

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Mixing Mania" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Mixing Mania")]
public sealed class MixingMania : IQuestIdentifier
```

Inheritance

[object](#) ← MixingMania

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MixingMania()

```
public MixingMania()
```

Class MoneyManagement

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Money Management" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Money Management")]
public sealed class MoneyManagement : IQuestIdentifier
```

Inheritance

[object](#) ← MoneyManagement

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Constructors

MoneyManagement()

```
public MoneyManagement()
```

Class MovingUp

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Moving Up" (Quest_MovingUp). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Moving Up")]
public sealed class MovingUp : IQuestIdentifier
```

Inheritance

[object](#) ← MovingUp

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

MovingUp()

```
public MovingUp()
```

Class NeedingTheGreen

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Needin' the Green" (Quest_NeedingTheGreen). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Needin' the Green")]
public sealed class NeedingTheGreen : IQuestIdentifier
```

Inheritance

[object](#) ← NeedingTheGreen

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

NeedingTheGreen()

```
public NeedingTheGreen()
```

Class OnTheGrind

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "On the Grind" (Quest_OnTheGrind). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("On the Grind")]
public sealed class OnTheGrind : IQuestIdentifier
```

Inheritance

[object](#) ← OnTheGrind

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

OnTheGrind()

```
public OnTheGrind()
```

Class Packagers

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Handlers" (Quest_Packagers). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Handlers")]
public sealed class Packagers : IQuestIdentifier
```

Inheritance

[object](#) ← Packagers

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Packagers()

```
public Packagers()
```

Class Packin

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Packin" (Quest). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Packin'")]
public sealed class Packin : IQuestIdentifier
```

Inheritance

[object](#) ← Packin

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Packin()

```
public Packin()
```

Class QuestNameAttribute

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Annotate quest identifier types with the display title of the quest. Usage: [QuestName("Finishing the Job")]

```
[AttributeUsage(AttributeTargets.Class, AllowMultiple = false, Inherited = false)]
public sealed class QuestNameAttribute : Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← QuestNameAttribute

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.GetType\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object, object\)](#) ,
[object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

QuestNameAttribute(string)

```
public QuestNameAttribute(string name)
```

Parameters

name [string](#)

Properties

Name

```
public string Name { get; }
```

Property Value

[string](#)

Class UnfavourableAgreements

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Unfavourable Agreements" (Quest_UnfavourableAgreements).
Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Unfavourable Agreements")]
public sealed class UnfavourableAgreements : IQuestIdentifier
```

Inheritance

[object](#) ← UnfavourableAgreements

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

UnfavourableAgreements()

```
public UnfavourableAgreements()
```

Class Warehouse

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Wretched Hive of Scum and Villainy" (Quest_Warehouse). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Wretched Hive of Scum and Villainy")]
public sealed class Warehouse : IQuestIdentifier
```

Inheritance

[object](#) ← Warehouse

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

Warehouse()

```
public Warehouse()
```

Class WeNeedToCook

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "We Need To Cook" (Quest_WeNeedToCook). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("We Need To Cook")]
public sealed class WeNeedToCook : IQuestIdentifier
```

Inheritance

[object](#) ← WeNeedToCook

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WeNeedToCook()

```
public WeNeedToCook()
```

Class WelcomeToHylandPoint

Namespace: [S1API.Quests.Identifiers](#)

Assembly: S1API.dll

Identifier for the base-game quest "Welcome to Hyland Point" (Quest_WelcomeToHylandPoint). Modders can use [Get<T>\(\)](#) to resolve it.

```
[QuestName("Welcome to Hyland Point")]
public sealed class WelcomeToHylandPoint : IQuestIdentifier
```

Inheritance

[object](#) ← WelcomeToHylandPoint

Implements

[IQuestIdentifier](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

WelcomeToHylandPoint()

```
public WelcomeToHylandPoint()
```

Class Quest

Namespace: [S1API.Quests](#)

Assembly: S1API.dll

An abstract class intended to be derived from for creating custom quests in the game.

```
public abstract class Quest : Saveable
```

Inheritance

[object](#) ← [Registerable](#) ← [Saveable](#) ← Quest

Inherited Members

[Saveable.RequestGameSave\(bool\)](#) , [Saveable.OnLoaded\(\)](#) , [Saveable.OnSaved\(\)](#) , [Saveable.LoadOrder](#) ,
[Registerable.OnCreated\(\)](#) , [Registerable.OnDestroyed\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

To create a quest, you must use [CreateQuest<T>\(string?\)](#) to instantiate your quest.

Constructors

Quest()

INTERNAL: Public constructor used for instancing the quest.

```
public Quest()
```

Fields

QuestEntries

A list of all quest entries added to this quest.

```
public readonly List<QuestEntry> QuestEntries
```

Field Value

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

Properties

AutoBegin

Whether to automatically begin the quest once instanced. NOTE: If this is false, you must manually [.Begin\(\)](#) this quest.

```
protected virtual bool AutoBegin { get; }
```

Property Value

[bool](#)

Description

The description provided to the player.

```
protected abstract string Description { get; }
```

Property Value

[string](#)

QuestIcon

Optional icon sprite to display for the quest. Override to use a custom icon loaded at runtime (e.g., from a file). `protected override Sprite? QuestIcon => ImageUtils.LoadImage("icon.png");`

```
protected virtual Sprite? QuestIcon { get; }
```

Property Value

Sprite

QuestState

The current quest state for this quest

```
protected QuestState QuestState { get; }
```

Property Value

[QuestState](#)

Title

The title of the quest to display for the player.

```
protected abstract string Title { get; }
```

Property Value

[string](#) ↗

Methods

AddEntry(string, NPC)

Adds a new quest entry to the quest with an NPC as the POI location. The POI marker will automatically update when the NPC moves.

```
protected QuestEntry AddEntry(string title, NPC npc)
```

Parameters

title [string](#)

The title for the quest entry.

npc [NPC](#)

The NPC to use as the POI location.

Returns

[QuestEntry](#)

A reference to the quest entry

AddEntry(string, Vector3?)

Adds a new quest entry to the quest.

```
protected QuestEntry AddEntry(string title, Vector3? poiPosition = null)
```

Parameters

title [string](#)

The title for the quest entry.

poiPosition [Vector3?](#)

A position for the point-of-interest, if applicable.

Returns

[QuestEntry](#)

A reference to the quest entry

Begin()

Starts the quest for the save file.

```
public void Begin()
```

Cancel()

Cancels the quest for the save file.

```
public void Cancel()
```

Complete()

Completes the quest for the save file.

```
public void Complete()
```

End()

Ends the quest for the save file. NOTE: This is done upon completion of the entries by default.

```
public void End()
```

Expire()

Expires the quest for the save file.

```
public void Expire()
```

Fail()

Fails the quest for the save file.

```
public void Fail()
```

Events

OnComplete

An action called once a quest has been completed.

```
public event Action OnComplete
```

Event Type

[Action](#)

OnFail

An action called once a quest has been failed.

```
public event Action? OnFail
```

Event Type

[Action ↗](#)

See Also

[RequestGameSave\(bool ↗\)](#)

Class QuestData

Namespace: [S1API.Quests](#)

Assembly: S1API.dll

Generic data applied for all custom quests.

```
public class QuestData
```

Inheritance

[object](#) ← QuestData

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

QuestData(string)

Creates a new quest data descriptor.

```
public QuestData(string className)
```

Parameters

className [string](#)

Fully qualified quest type name (e.g., "MyMod.Quests.IntroQuest").

Fields

ClassName

Fully qualified quest class name used by the game to instantiate or resolve the custom quest.

```
public readonly string ClassName
```

Field Value

[string](#) ↗

Class QuestEntry

Namespace: [S1API.Quests](#)

Assembly: S1API.dll

Represents a quest entry on a quest. These are the individual [tasks](#) on a quest.

```
public class QuestEntry
```

Inheritance

[object](#) ← QuestEntry

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

POIPosition

The point-of-interest world position. Returns Vector3.zero if no location is set. Setting a position will create/update the PolLocation transform if it doesn't exist.

```
public Vector3 POIPosition { get; set; }
```

Property Value

Vector3

State

The current state of this quest entry.

```
public QuestState State { get; }
```

Property Value

[QuestState](#)

Title

The title displayed for the quest entry.

```
public string Title { get; set; }
```

Property Value

[string](#) ↗

Methods

Begin()

Marks the quest entry as started and transitions its [State](#) to the in-progress state in-game.

```
public void Begin()
```

Complete()

Marks the quest entry as completed.

```
public void Complete()
```

SetPOIToNPC(NPC)

Sets the POI location to follow an NPC instance. The POI marker will automatically update when the NPC moves. If the POI hasn't been created yet, it will be created automatically.

```
public bool SetPOIToNPC(NPC npc)
```

Parameters

npc [NPC](#)

The NPC instance to follow.

Returns

[bool](#)

True if the NPC was valid and POI location was set, false otherwise.

SetPOIToNPC<T>()

Sets the POI location to follow an NPC by type. The POI marker will automatically update when the NPC moves.

```
public bool SetPOIToNPC<T>() where T : NPC
```

Returns

[bool](#)

True if the NPC was found and POI location was set, false otherwise.

Type Parameters

T

The NPC type to follow.

SetPOIToSpraySurface(SpraySurface)

Sets the POI location to a spray surface's position.

```
public bool SetPOIToSpraySurface(SpraySurface spraySurface)
```

Parameters

spraySurface [SpraySurface](#)

The spray surface to set POI to.

Returns

[bool](#) ↗

True if the spray surface was valid and POI location was set, false otherwise.

SetState(QuestState)

Manually sets the state of the quest entry.

```
public void SetState(QuestState questState)
```

Parameters

questState [QuestState](#)

The state you want the entry to be.

Events

OnComplete

An action called once a quest has been completed.

```
public event Action OnComplete
```

Event Type

[Action](#) ↗

Class QuestManager

Namespace: [S1API.Quests](#)

Assembly: S1API.dll

Provided management of quests across the game.

```
public static class QuestManager
```

Inheritance

[object](#) ← QuestManager

Inherited Members

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

Methods

CreateQuest(Type, string?)

Creates a new quest for the player to complete from your custom quest class.

```
public static Quest CreateQuest(Type questType, string? guid = null)
```

Parameters

questType [Type](#)

Your custom quest class that derived from [Quest](#).

guid [string](#)

The unique identifier for this quest. By default, assigned a random GUID.

Returns

[Quest](#)

CreateQuest<T>(string?)

Creates a new quest for the player to complete from your custom quest class.

```
public static Quest CreateQuest<T>(string? guid = null) where T : Quest
```

Parameters

guid [string](#)

The unique identifier for this quest. By default, assigned a random GUID.

Returns

[Quest](#)

The instance of your quest.

Type Parameters

T

Your custom quest class that derived from [Quest](#).

GetQuestByGuid(string)

Returns a [Quest](#) instance by the Quest GUID

```
public static Quest? GetQuestByGuid(string guid)
```

Parameters

guid [string](#)

The unique identifier to use for searching this quest

Returns

[Quest](#)

The quest instance

GetQuestByName(string)

Returns a [Quest](#) instance by the Quest Name. Searches custom mod quests only. For base game quests, use S1API.Quests.QuestManager.GetBaseGameQuestByName(System.String).

```
public static Quest? GetQuestByName(string questName)
```

Parameters

questName [string](#)

The quest title to use for searching this quest

Returns

[Quest](#)

The quest instance, or null if not found

Get<T>()

Returns a [QuestWrapper](#) instance using a typed identifier. Declare an identifier class annotated with [Identifiers.QuestName("...")]. Works for both custom mod quests and base game quests.

```
public static QuestWrapper? Get<T>() where T : IQuestIdentifier
```

Returns

[QuestWrapper](#)

The quest wrapper instance, or null if not found.

Type Parameters

T

A quest identifier type implementing [IQuestIdentifier](#).

Class QuestWrapper

Namespace: [S1API.Quests](#)

Assembly: S1API.dll

Wrapper for accessing quests that can handle both custom mod quests and base game quests. Provides unified access to quest events and properties.

```
public sealed class QuestWrapper
```

Inheritance

[object](#) ← QuestWrapper

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

QuestEntries

Gets the quest entries for this quest.

```
public List<QuestEntry> QuestEntries { get; }
```

Property Value

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

Title

The quest title.

```
public string Title { get; }
```

Property Value

[string ↗](#)

Events

OnComplete

An action called once a quest has been completed.

```
public event Action OnComplete
```

Event Type

[Action ↗](#)

OnFail

An action called once a quest has been failed.

```
public event Action OnFail
```

Event Type

[Action ↗](#)

Namespace S1API.Rendering

Classes

[AccessoryFactory](#)

Factory for creating custom accessory prefabs at runtime by cloning and modifying existing ones.

[IconFactory](#)

Factory for generating item icons using the game's IconGenerator and MugshotGenerator.

Item Icon Generation (Experimental): Direct item icon generation using IconGenerator is experimental.

Use [GenerateIcon\(Transform, int, bool\)](#) for static mesh item models.

Accessory Icon Generation (Confirmed Working): Accessory icon generation using MugshotGenerator is confirmed to work. Use [GenerateAccessoryIcon\(string, Action<Texture2D>, Color?, int\)](#) or [GenerateAccessoryIconSprite\(string, Action<Sprite>, Color?, int\)](#) for clothing accessories (hats, glasses, etc.).

[MaterialHelper](#)

Utility methods for working with Unity materials and shaders. Provides safe, convenient methods for common material operations.

[RuntimeResourceRegistry](#)

Provides runtime registration of Unity Resources that can be loaded via Resources.Load. This allows mods to inject custom assets without modifying Unity asset bundles.

[TextureUtils](#)

Utility class for loading and creating textures at runtime.

Class AccessoryFactory

Namespace: [S1API.Rendering](#)

Assembly: S1API.dll

Factory for creating custom accessory prefabs at runtime by cloning and modifying existing ones.

```
public static class AccessoryFactory
```

Inheritance

[object](#) ← AccessoryFactory

Inherited Members

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

Methods

**CloneAccessoryWithCustomTextures(string, string, Dictionary<string,
Texture2D>, Color?, string)**

Clones an accessory prefab and applies custom textures/materials.

```
public static GameObject CloneAccessoryWithCustomTextures(string sourceResourcePath,  
string newName, Dictionary<string, Texture2D> textureReplacements = null, Color?  
colorTint = null, string targetResourcePath = null)
```

Parameters

sourceResourcePath [string](#)

The Resources path to the source accessory prefab.

newName [string](#)

Name for the cloned accessory.

`textureReplacements` [Dictionary<string, Texture2D>](#)

Optional dictionary of shader texture names and replacement textures.

`colorTint` [Color?](#)

Optional color tint to apply.

`targetResourcePath` [string](#)

Optional target resource path for the cloned accessory. If provided, sets Accessory.AssetPath to this value instead of sourceResourcePath. This ensures PlayerClothing validation can match the accessory correctly.

Returns

[GameObject](#)

The cloned and customized accessory [GameObject](#), or null if cloning failed.

`CreateAndRegisterAccessory(string, string, string, Dictionary<string, Texture2D>, Color?)`

Creates a custom accessory by cloning a source and applies custom textures, then registers it. This is a convenience method that combines cloning, customization, and registration.

```
public static bool CreateAndRegisterAccessory(string sourceResourcePath, string targetResourcePath, string newName, Dictionary<string, Texture2D> textureReplacements = null, Color? colorTint = null)
```

Parameters

`sourceResourcePath` [string](#)

The Resources path to the source accessory prefab.

`targetResourcePath` [string](#)

The Resources path where the custom accessory will be registered.

newName [string](#)

Name for the cloned accessory.

textureReplacements [Dictionary](#)<[string](#), Texture2D>

Optional dictionary of shader texture names and replacement textures.

colorTint Color?

Optional color tint to apply.

Returns

[bool](#)

True if the accessory was successfully created and registered.

RegisterAccessory(string, GameObject)

Registers a cloned accessory with the RuntimeResourceRegistry so it can be loaded via Resources.Load.

```
public static bool RegisterAccessory(string resourcePath, GameObject accessory)
```

Parameters

resourcePath [string](#)

The Resources path to register the accessory at.

accessory GameObject

The accessory GameObject to register.

Returns

[bool](#)

True if registration was successful.

Class IconFactory

Namespace: [S1API.Rendering](#)

Assembly: S1API.dll

Factory for generating item icons using the game's IconGenerator and MugshotGenerator.

Item Icon Generation (Experimental): Direct item icon generation using IconGenerator is experimental. Use [GenerateIcon\(Transform, int, bool\)](#) for static mesh item models.

Accessory Icon Generation (Confirmed Working): Accessory icon generation using MugshotGenerator is confirmed to work. Use [GenerateAccessoryIcon\(string, Action<Texture2D>, Color?, int\)](#) or [GenerateAccessoryIconSprite\(string, Action<Sprite>, Color?, int\)](#) for clothing accessories (hats, glasses, etc.).

```
public static class IconFactory
```

Inheritance

[object](#) ← IconFactory

Inherited Members

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

Methods

GenerateAccessoryIcon(string, Action<Texture2D>, Color?, int)

Generates an icon for an accessory by rendering it on an avatar using MugshotGenerator. This is an asynchronous operation that will invoke the callback when complete.

```
public static void GenerateAccessoryIcon(string accessoryPath, Action<Texture2D>
callback, Color? accessoryColor = null, int size = 512)
```

Parameters

accessoryPath [string](#)

The resource path to the accessory (e.g., from Appearances.AccessoryFields.Head)

callback [Action](#)<Texture2D>

Callback invoked with the generated texture when complete

accessoryColor [Color?](#)

Optional tint color for the accessory (defaults to white)

size [int](#)

The size of the square icon (default 512)

Remarks

This method attempts to use the local player's current avatar settings for a personalized icon. However, if called during mod initialization (e.g., in OnSceneWasLoaded), the player may not have spawned yet, causing it to fall back to a generic avatar with neutral face, eyes, and basic clothing. To use the player's actual appearance, consider deferring icon generation until after player spawn by subscribing to [LocalPlayerSpawned](#).

GenerateAccessoryIconSprite(string, Action<Sprite>, Color?, int)

Generates an icon for an accessory as a Sprite by rendering it on an avatar using MugshotGenerator. This is an asynchronous operation that will invoke the callback when complete.

```
public static void GenerateAccessoryIconSprite(string accessoryPath, Action<Sprite> callback, Color? accessoryColor = null, int size = 512)
```

Parameters

accessoryPath [string](#)

The resource path to the accessory (e.g., from Appearances.AccessoryFields.Head)

callback [Action](#)<Sprite>

Callback invoked with the generated sprite when complete

accessoryColor Color?

Optional tint color for the accessory (defaults to white)

size [int](#)

The size of the square icon (default 512)

GenerateIcon(Transform, int, bool)

Generates a preview texture for the specified model.

```
public static Texture2D? GenerateIcon(Transform model, int size = 512, bool  
bakeSkinnedMeshes = true)
```

Parameters

model Transform

The model to generate an icon for.

size [int](#)

The size of the square icon (default 512).

bakeSkinnedMeshes [bool](#)

If true, bakes SkinnedMeshRenderers to static MeshRenderers to ensure correct bounds (default true).

Returns

Texture2D

A Texture2D containing the icon, or null if generation failed.

GenerateIconSprite(Transform, int, bool)

Generates a preview icon as a Sprite for the specified model.

```
public static Sprite? GenerateIconSprite(Transform model, int size = 512, bool bakeSkinnedMeshes = true)
```

Parameters

model Transform

The model to generate an icon for.

size [int](#)

The size of the square icon (default 512).

bakeSkinnedMeshes [bool](#)

If true, bakes SkinnedMeshRenderers to static MeshRenderers to ensure correct bounds (default true).

Returns

Sprite

A Sprite containing the icon, or null if generation failed.

GeneratePackagingIcon(string, string)

Generates an icon for a packaging ID and product ID.

```
public static Texture2D? GeneratePackagingIcon(string packagingID, string productID)
```

Parameters

packagingID [string](#)

The ID of the packaging visuals to use.

productID [string](#)

The ID of the product to display in the packaging.

Returns

Texture2D

A Texture2D containing the packaging icon, or null if generation failed.

GeneratePackagingIconSprite(string, string)

Generates an icon as a Sprite for a packaging ID and product ID.

```
public static Sprite? GeneratePackagingIconSprite(string packagingID,  
string productID)
```

Parameters

packagingID [string](#)

The ID of the packaging visuals to use.

productID [string](#)

The ID of the product to display in the packaging.

Returns

Sprite

A Sprite containing the packaging icon, or null if generation failed.

Class MaterialHelper

Namespace: [S1API.Rendering](#)

Assembly: S1API.dll

Utility methods for working with Unity materials and shaders. Provides safe, convenient methods for common material operations.

```
public static class MaterialHelper
```

Inheritance

[object](#) ← MaterialHelper

Inherited Members

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

Methods

ConfigureMaterial(Material, Action<Material>)

Sets multiple shader properties at once using a configuration action.

```
public static void ConfigureMaterial(Material material,  
Action<Material> configurator)
```

Parameters

material Material

The material to modify.

configurator [Action](#)<Material>

Action to configure material properties.

Examples

```
MaterialHelper.ConfigureMaterial(material, mat => {
    MaterialHelper.SetColor(mat, "_BaseColor", Color.red);
    MaterialHelper.SetFloat(mat, "_Metallic", 0.8f);
    MaterialHelper.SetFloat(mat, "_Smoothness", 0.5f);
});
```

CreateMetallicVariant(Material, Color, float, float)

Creates a metallic material variant from an existing material. Removes all textures and applies metallic properties.

```
public static Material CreateMetallicVariant(Material baseMaterial, Color
metalColor, float metallic = 0.8, float smoothness = 0.5)
```

Parameters

baseMaterial Material

The material to clone and modify.

metalColor Color

The color to apply to the metal.

metallic [float](#)

Metallic value (0.0 to 1.0, default 0.8).

smoothness [float](#)

Smoothness value (0.0 to 1.0, default 0.5).

Returns

Material

A new material with metallic properties.

Examples

```
var metalMaterial = MaterialHelper.CreateMetallicVariant(  
    originalMaterial,  
    new Color(0.5f, 0.5f, 0.55f), // Gray metal  
    metallic: 0.8f,  
    smoothness: 0.5f  
);
```

RemoveAllTextures(Material)

Removes all textures from a material. Iterates through all shader properties and sets texture properties to null.

```
public static void RemoveAllTextures(Material material)
```

Parameters

material Material

The material to modify.

ReplaceMaterials(GameObject, Func<Material, bool>, Action<Material>)

Replaces materials on a GameObject and all its children based on a predicate. Creates new material instances to avoid modifying shared materials.

```
public static void ReplaceMaterials(GameObject gameObject, Func<Material, bool>  
    predicate, Action<Material> materialModifier)
```

Parameters

gameObject GameObject

The GameObject to process (will include all children).

predicate [Func](#)<Material, [bool](#)>

Function that returns true for materials to replace.

materialModifier [Action](#)<Material>

Action to modify the matched materials.

Examples

```
// Replace all materials with "wood" in the name
MaterialHelper.ReplaceMaterials(
    myObject,
    mat => mat.name.ToLower().Contains("wood"),
    mat => {
        MaterialHelper.SetColor(mat, "_BaseColor", Color.red);
        MaterialHelper.SetFloat(mat, "_Metallic", 0.8f);
    }
);
```

SetColor(Material, string, Color)

Sets a color property on a material, checking if the property exists first.

```
public static bool SetColor(Material material, string propertyName, Color color)
```

Parameters

material Material

The material to modify.

propertyName [string](#)

The shader property name (e.g., "_BaseColor").

color Color

The color value to set.

Returns

[bool](#)

True if the property was set, false if it doesn't exist.

SetFloat(Material, string, float)

Sets a float property on a material, checking if the property exists first.

```
public static bool SetFloat(Material material, string propertyName, float value)
```

Parameters

[material](#) Material

The material to modify.

[propertyName](#) [string](#)

The shader property name (e.g., "_Metallic").

[value](#) [float](#)

The float value to set.

Returns

[bool](#)

True if the property was set, false if it doesn't exist.

SetTexture(Material, string, Texture)

Sets a texture property on a material, checking if the property exists first.

```
public static bool SetTexture(Material material, string propertyName,
```

```
Texture texture)
```

Parameters

material Material

The material to modify.

propertyName [string](#)

The shader property name (e.g., "_MainTex").

texture Texture

The texture to set (null to clear).

Returns

[bool](#)

True if the property was set, false if it doesn't exist.

Class RuntimeResourceRegistry

Namespace: [S1API.Rendering](#)

Assembly: S1API.dll

Provides runtime registration of Unity Resources that can be loaded via Resources.Load. This allows mods to inject custom assets without modifying Unity asset bundles.

```
public static class RuntimeResourceRegistry
```

Inheritance

[object](#) ← RuntimeResourceRegistry

Inherited Members

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

Methods

GetRegisteredAsset(string)

Gets a registered asset.

```
public static Object GetRegisteredAsset(string resourcePath)
```

Parameters

resourcePath [string](#)

The Resources path.

Returns

Object

The registered asset, or null if not found.

GetRegisteredAsset<T>(string)

Gets a registered asset of a specific type.

```
public static T GetRegisteredAsset<T>(string resourcePath) where T : Object
```

Parameters

resourcePath [string](#)

The Resources path.

Returns

T

The registered asset cast to type T, or null if not found or wrong type.

Type Parameters

T

The type of asset to retrieve.

IsRegistered(string)

Checks if an asset is registered at the given path.

```
public static bool IsRegistered(string resourcePath)
```

Parameters

resourcePath [string](#)

The Resources path to check.

Returns

[bool](#)

True if registered, false otherwise.

RegisterAsset(string, Object)

Registers an asset with a Resources path. After registration, the asset can be loaded via Resources.Load using the provided path.

```
public static bool RegisterAsset(string resourcePath, Object asset)
```

Parameters

[resourcePath](#) [string](#)

The Resources path (e.g., "MyMod/Accessories/CustomHat").

[asset](#) [Object](#)

The Unity Object to register.

Returns

[bool](#)

True if registration was successful.

RegisterGameObject(string, GameObject)

Registers a GameObject asset with a Resources path.

```
public static bool RegisterGameObject(string resourcePath, GameObject gameObject)
```

Parameters

resourcePath [string](#)

The Resources path.

gameObject [GameObject](#)

The GameObject to register.

Returns

[bool](#)

True if registration was successful.

Class TextureUtils

Namespace: [S1API.Rendering](#)

Assembly: S1API.dll

Utility class for loading and creating textures at runtime.

```
public static class TextureUtils
```

Inheritance

[object](#) ← TextureUtils

Inherited Members

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

Methods

LoadTextureFromBytes(byte[], FilterMode, TextureWrapMode)

Loads a texture from a byte array.

```
public static Texture2D LoadTextureFromBytes(byte[] imageData, FilterMode filterMode  
= 1, TextureWrapMode wrapMode = 1)
```

Parameters

imageData [byte](#)[]

The image data bytes.

filterMode FilterMode

The filter mode for the texture.

wrapMode TextureWrapMode

The wrap mode for the texture.

Returns

Texture2D

The loaded texture, or null if loading failed.

LoadTextureFromFile(string, FilterMode, TextureWrapMode)

Loads a texture from a file path.

```
public static Texture2D LoadTextureFromFile(string filePath, FilterMode filterMode = 1, TextureWrapMode wrapMode = 1)
```

Parameters

filePath string

The path to the image file.

filterMode FilterMode

The filter mode for the texture.

wrapMode TextureWrapMode

The wrap mode for the texture.

Returns

Texture2D

The loaded texture, or null if loading failed.

LoadTextureFromResource(Assembly, string, FilterMode, TextureWrapMode)

Loads a texture from an embedded resource.

```
public static Texture2D LoadTextureFromResource(Assembly assembly, string  
resourceName, FilterMode filterMode = 1, TextureWrapMode wrapMode = 1)
```

Parameters

assembly [Assembly](#)

The assembly containing the embedded resource.

resourceName [string](#)

The fully qualified name of the embedded resource.

filterMode [FilterMode](#)

The filter mode for the texture.

wrapMode [TextureWrapMode](#)

The wrap mode for the texture.

Returns

[Texture2D](#)

The loaded texture, or null if loading failed.

Namespace S1API.Saveables

Classes

[SaveableField](#)

Marks a field to be saved alongside the class instance. This attribute is intended to work across all custom game elements. (For example, custom NPCs, quests, etc.) DO NOT NAME THE FIELD "QuestData" AS THIS WILL CONFLICT WITH THE API.

Enums

[SaveableLoadOrder](#)

Defines when a modded saveable should load relative to base game saveables.

Override the S1API.Internal.Abstraction.Saveable.LoadOrder property in your S1API.Internal.Abstraction.Saveable class to control load timing.

Class SaveableField

Namespace: [S1API.Saveables](#)

Assembly: S1API.dll

Marks a field to be saved alongside the class instance. This attribute is intended to work across all custom game elements. (For example, custom NPCs, quests, etc.) DO NOT NAME THE FIELD "QuestData" AS THIS WILL CONFLICT WITH THE API.

```
[AttributeUsage(AttributeTargets.Field)]
public class SaveableField : Attribute
```

Inheritance

[object](#) ← [Attribute](#) ← SaveableField

Inherited Members

[Attribute.GetCustomAttributes\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(MemberInfo\)](#) , [Attribute.GetCustomAttributes\(MemberInfo, bool\)](#) ,
[Attribute.IsDefined\(MemberInfo, Type\)](#) , [Attribute.IsDefined\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(MemberInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo\)](#) , [Attribute.GetCustomAttributes\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(ParameterInfo, bool\)](#) , [Attribute.IsDefined\(ParameterInfo, Type\)](#) ,
[Attribute.IsDefined\(ParameterInfo, Type, bool\)](#) , [Attribute.GetCustomAttribute\(ParameterInfo, Type\)](#) ,
[Attribute.GetCustomAttribute\(ParameterInfo, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Module, Type\)](#) , [Attribute.GetCustomAttributes\(Module\)](#) ,
[Attribute.GetCustomAttributes\(Module, bool\)](#) , [Attribute.GetCustomAttributes\(Module, Type, bool\)](#) ,
[Attribute.IsDefined\(Module, Type\)](#) , [Attribute.IsDefined\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttribute\(Module, Type\)](#) , [Attribute.GetCustomAttribute\(Module, Type, bool\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttributes\(Assembly\)](#) ,
[Attribute.GetCustomAttributes\(Assembly, bool\)](#) , [Attribute.IsDefined\(Assembly, Type\)](#) ,
[Attribute.IsDefined\(Assembly, Type, bool\)](#) , [Attribute.GetCustomAttribute\(Assembly, Type\)](#) ,
[Attribute.GetCustomAttribute\(Assembly, Type, bool\)](#) , [Attribute.Equals\(object\)](#) ,
[Attribute.GetHashCode\(\)](#) , [Attribute.Match\(object\)](#) , [Attribute.IsDefaultAttribute\(\)](#) ,
[Attribute.TypeId](#) , [object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

SaveableField(string)

Base constructor for initializing a SaveableField.

```
public SaveableField(string saveName)
```

Parameters

saveName [string](#)

Enum SaveableLoadOrder

Namespace: [S1API.Saveables](#)

Assembly: S1API.dll

Defines when a modded saveable should load relative to base game saveables.

Override the S1API.Internal.Abstraction.Saveable.LoadOrder property in your S1API.Internal.Abstraction.Saveable class to control load timing.

```
public enum SaveableLoadOrder
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Fields

AfterBaseGame = 1

Load after base game saveables (default). Runs as a postfix after NPCsLoader.Load (one of the last loaders).

Use this when your mod data depends on base game entities being loaded first, such as:

- Storing references to NPCs, buildings, or vehicles
- Modifying base game entity states after they're loaded
- Most general-purpose mod saveables

This is the **default behavior**. If you don't override LoadOrder, your saveable will use AfterBaseGame. Base game entities **are loaded** when S1API.Internal.Abstraction.Saveable.OnLoaded is called.

BeforeBaseGame = 0

Load before base game saveables. Runs as a prefix before BuildingsLoader.Load (one of the earliest loaders).

Use this when your mod data needs to be available before base game ISaveables load, such as:

- Setting up hooks that intercept base game load events
- Initializing global state that base game loaders depend on
- Advanced modding scenarios requiring early initialization

Important: When using BeforeBaseGame, base game entities (NPCs, buildings, vehicles) are **not yet loaded** when S1API.Internal.Abstraction.Saveable.OnLoaded is called.

Examples

```
public class MyCustomSaveable : Saveable
{
    // Override to load BEFORE base game entities
    public override SaveableLoadOrder LoadOrder => SaveableLoadOrder.BeforeBaseGame;

    [SaveableField("my_data")]
    private MyDataClass _myData = new MyDataClass();
}
```

Namespace S1API.Shops

Classes

[Shop](#)

Represents an in-game shop where items can be purchased. Provides high-level operations without exposing game types.

[ShopManager](#)

Provides access to all shops in the game and convenience methods for shop integration.

Class Shop

Namespace: [S1API.Shops](#)

Assembly: S1API.dll

Represents an in-game shop where items can be purchased. Provides high-level operations without exposing game types.

```
public sealed class Shop
```

Inheritance

[object](#) ← Shop

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Name

The display name of this shop.

```
public string Name { get; }
```

Property Value

[string](#)

Methods

AddItem(ItemDefinition, float?)

Adds an item to this shop's inventory with automatic UI creation and event hookup. The item will appear in the shop's listing and be purchasable by the player.

```
public bool AddItem(ItemDefinition item, float? customPrice = null)
```

Parameters

item [ItemDefinition](#)

The item definition to add to the shop.

customPrice [float](#)?

Optional custom price override. If null, uses item's BasePurchasePrice.

Returns

[bool](#)

True if the item was added successfully, false if it already exists or addition failed.

Examples

```
var shop = ShopManager.GetShopByName("Hardware Store");
var customItem = ItemManager.GetItemDefinition("my_custom_tool");
shop.AddItem(customItem);
```

GetItemIds()

Gets all item IDs currently sold by this shop.

```
public string[] GetItemIds()
```

Returns

[string](#) []

Array of item IDs.

HasItem(string)

Checks if this shop currently sells an item with the specified ID.

```
public bool HasItem(string itemId)
```

Parameters

[itemId](#) [string](#) []

The unique ID of the item to check.

Returns

[bool](#) []

True if the shop sells this item, false otherwise.

RemoveItem(string)

Removes an item from this shop's inventory.

```
public bool RemoveItem(string itemId)
```

Parameters

[itemId](#) [string](#) []

The ID of the item to remove.

Returns

[bool](#)

True if the item was removed, false if it wasn't found.

SellsCategory(ItemCategory)

Checks if this shop sells items in the specified category.

```
public bool SellsCategory(ItemCategory category)
```

Parameters

[category](#) [ItemCategory](#)

The item category to check.

Returns

[bool](#)

True if the shop sells at least one item in this category.

Class ShopManager

Namespace: [S1API.Shops](#)

Assembly: S1API.dll

Provides access to all shops in the game and convenience methods for shop integration.

```
public static class ShopManager
```

Inheritance

[object](#) ← ShopManager

Inherited Members

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

Methods

AddToCompatibleShops(ItemDefinition, float?)

Adds an item to all shops that sell items in the same category. This is a convenience method for making custom items available wherever similar items are sold.

```
public static int AddToCompatibleShops(ItemDefinition item, float? customPrice  
= null)
```

Parameters

item [ItemDefinition](#)

The item to add to compatible shops.

customPrice [float](#)?

Optional custom price override.

Returns

[int ↗](#)

The number of shops the item was added to.

Examples

```
var metalRack = ItemManager.GetItemDefinition("metalstoragerack");
int shopsUpdated = ShopManager.AddToCompatibleShops(metalRack);
Logger.Msg($"Added metal rack to {shopsUpdated} shops");
```

AddToShops(ItemDefinition, float?, params string[])

Adds an item to specific shops by name with a custom price.

```
public static int AddToShops(ItemDefinition item, float? customPrice, params
string[] shopNames)
```

Parameters

[item](#) [ItemDefinition](#)

The item to add.

[customPrice](#) [float ↗?](#)

Optional custom price override.

[shopNames](#) [string ↗\[\]](#)

Names of shops to add the item to.

Returns

[int ↗](#)

The number of shops the item was added to.

AddToShops(ItemDefinition, params string[])

Adds an item to specific shops by name.

```
public static int AddToShops(ItemDefinition item, params string[] shopNames)
```

Parameters

item [ItemDefinition](#)

The item to add.

shopNames [string\[\]](#)

Names of shops to add the item to.

Returns

[int](#)

The number of shops the item was added to.

Examples

```
ShopManager.AddToShops(myItem, "Hardware Store", "General Store");
```

FindShopsByCategory(ItemCategory)

Finds all shops that sell items in the specified category.

```
public static Shop[] FindShopsByCategory(ItemCategory category)
```

Parameters

category [ItemCategory](#)

The item category to search for.

Returns

[Shop](#) []

Array of shops that sell items in this category.

Examples

```
// Find all shops that sell tools
var toolShops = ShopManager.FindShopsByCategory(ItemCategory.Tools);
```

FindShopsByItem(string)

Finds all shops that sell a specific item.

```
public static Shop[] FindShopsByItem(string itemId)
```

Parameters

`itemId` [string](#) ↗

The ID of the item to search for.

Returns

[Shop](#) []

Array of shops that sell this item.

GetAllShops()

Gets all shops currently loaded in the game. Results are cached until the scene changes.

```
public static Shop[] GetAllShops()
```

Returns

[Shop](#) []

Array of all available shops.

GetShopByName(string)

Gets a shop by its display name (case-insensitive).

```
public static Shop GetShopByName(string shopName)
```

Parameters

shopName [string](#) ↗

The name of the shop to find.

Returns

[Shop](#)

The shop if found, null otherwise.

Examples

```
var hardwareStore = ShopManager.GetShopByName("Hardware Store");
if (hardwareStore != null)
{
    hardwareStore.AddItem(myCustomItem);
}
```

RefreshItemIcon(ItemDefinition)

Refreshes the icon displayed in shop listings for the specified item. This is useful when an item's icon is generated or updated after it has been added to shops.

```
public static int RefreshItemIcon(ItemDefinition item)
```

Parameters

item [ItemDefinition](#)

The item whose icon should be refreshed in shop listings.

Returns

[int](#)

The number of shop listings that were updated.

Examples

```
// After generating a custom icon for an item
myItem.Icon = generatedSprite;
int updated = ShopManager.RefreshItemIcon(myItem);
Logger.Msg($"Updated icon in {updated} shop listing(s)");
```

RefreshItemIcon(string)

Refreshes the icon displayed in shop listings for the specified item by ID. This is useful when an item's icon is generated or updated after it has been added to shops.

```
public static int RefreshItemIcon(string itemId)
```

Parameters

itemId [string](#)

The ID of the item whose icon should be refreshed.

Returns

[int](#)

The number of shop listings that were updated.

Namespace S1API.Storage

Classes

[StorageEntity](#)

Represents a storage entity in the game world. Provides cross-runtime safe manipulation of storage slots and properties.

[StorageEventArgs](#)

Event arguments for storage-specific events.

[StorageEvents](#)

[StorageLoadingEventArgs](#)

Event arguments for storage loading operations. Raised when storage is being loaded from a save file.

Class StorageEntity

Namespace: [S1API.Storage](#)

Assembly: S1API.dll

Represents a storage entity in the game world. Provides cross-runtime safe manipulation of storage slots and properties.

```
public sealed class StorageEntity
```

Inheritance

[object](#) ← StorageEntity

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This is the structural layer: it owns slot topology changes (add/remove/set count), display row adjustments, and placeable storage IO-slot wiring while handling Il2Cpp/Mono differences. Use this when modifying capacity or layout. For runtime interaction with an existing storage instance (metadata, item add/remove, events), prefer [StorageInstance](#).

Properties

CustomName

Gets or sets the custom name set by the player via the management clipboard. Returns null if this is not placeable storage or no custom name is set. Setting this value updates the Configuration.Name field which persists to save files.

```
public string CustomName { get; set; }
```

Property Value

[string](#)

Remarks

This is different from [Name](#) which is the base StorageEntityName. CustomName reflects the user-editable name shown in the clipboard UI. When a player renames storage via the clipboard, this property is updated.

DisplayRowCount

Number of rows to display in the storage UI. Automatically set to 2 when SlotCount exceeds 6.

```
public int DisplayRowCount { get; set; }
```

Property Value

[int](#)

HasCustomName

Gets whether this storage has a custom name set by the player.

```
public bool HasCustomName { get; }
```

Property Value

[bool](#)

IsPlaceable

Whether this storage entity is placeable in the world.

```
public bool IsPlaceable { get; }
```

Property Value

[bool](#) ↗

ItemId

The item ID of the storage container. Convenience property for filtering by storage type. Returns null if this storage is not placeable.

```
public string ItemId { get; }
```

Property Value

[string](#) ↗

ItemInstance

The item instance this storage is part of (e.g., the storage rack item). Returns null if this storage is not placeable.

```
public ItemInstance ItemInstance { get; }
```

Property Value

[ItemInstance](#)

MaxSlots

Maximum number of slots this storage can have. Default is 20 for most storage types.

```
public int MaxSlots { get; set; }
```

Property Value

[int ↗](#)

Name

The display name of this storage container shown in the UI.

```
public string Name { get; set; }
```

Property Value

[string ↗](#)

SlotCount

Current number of slots in this storage container. Setting this value will add or remove slots as needed.

```
public int SlotCount { get; set; }
```

Property Value

[int ↗](#)

SlotsAreFilterable

Whether slots in this storage are filterable (can restrict item types).

```
public bool SlotsAreFilterable { get; }
```

Property Value

[bool](#)

Methods

AddSlots(int)

Adds the specified number of slots to this storage container. Automatically updates input/output slots for placeable storage.

```
public bool AddSlots(int count)
```

Parameters

[count](#) [int](#)

Number of slots to add (must be positive)

Returns

[bool](#)

True if slots were added successfully

Remarks

This method handles all runtime-specific logic internally. Display row count is automatically adjusted for > 6 slots.

GetEmptySlotCount()

Gets the number of empty slots in this storage.

```
public int GetEmptySlotCount()
```

Returns

[int](#)

GetOccupiedSlotCount()

Gets the number of occupied slots in this storage.

```
public int GetOccupiedSlotCount()
```

Returns

[int](#)

HasItems()

Checks if this storage has any items in it.

```
public bool HasItems()
```

Returns

[bool](#)

IsEmpty()

Checks if this storage is completely empty.

```
public bool IsEmpty()
```

Returns

[bool](#)

RemoveSlots(int)

Removes the specified number of slots from this storage container. Only removes empty slots from the end.

```
public bool RemoveSlots(int count)
```

Parameters

[count](#) [int](#)

Number of slots to remove (must be positive)

Returns

[bool](#)

True if slots were removed successfully

Remarks

This method will fail if any of the slots to be removed contain items. Always check slot contents before removing.

SetSlotCount(int)

Sets the total number of slots, expanding or contracting as needed.

```
public bool SetSlotCount(int targetSlotCount)
```

Parameters

[targetSlotCount](#) [int](#)

Target number of slots

Returns

[bool](#)

True if operation was successful

SyncCustomNameToDisplayName()

Synchronizes the StorageEntityName with the custom name from Configuration. Call this before opening the storage menu to ensure the custom name is displayed.

```
public void SyncCustomNameToDisplayName()
```

Remarks

The base game's StorageMenu uses StorageEntityName for display, but custom names set via the clipboard are stored in Configuration.Name. This method syncs them.

Class StorageEventArgs

Namespace: [S1API.Storage](#)

Assembly: S1API.dll

Event arguments for storage-specific events.

```
public class StorageEventArgs
```

Inheritance

[object](#) ← StorageEventArgs

Derived

[StorageLoadingEventArgs](#)

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.HasValueTuple\(object\)](#)

Properties

ItemId

The item ID of the storage container. Convenience property for filtering by storage type.

```
public string ItemId { get; }
```

Property Value

[string](#)

Storage

The storage entity involved in the event.

```
public StorageEntity Storage { get; }
```

Property Value

[StorageEntity](#)

Class StorageEvents

Namespace: [S1API.Storage](#)

Assembly: S1API.dll

```
public static class StorageEvents
```

Inheritance

[object](#) ← StorageEvents

Inherited Members

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

Events

OnStorageCreated

Event raised after a storage entity is created and initialized in the world. This event fires when storage items are placed by the player.

```
public static event Action<StorageEventArgs> OnStorageCreated
```

Event Type

[Action](#)<[StorageEventArgs](#)>

Remarks

Use this event to customize storage properties when items are placed. This is the primary event for expanding storage slots on placement.

OnStorageLoading

Event raised before items are loaded into storage from a save file. Use this event to expand storage slots to accommodate saved items.

```
public static event Action<StorageLoadingEventArgs> OnStorageLoading
```

Event Type

[Action<StorageLoadingEventArgs>](#)

Remarks

This event is critical for save compatibility when you've expanded storage slots. Check args.NeedsMoreSlots to determine if expansion is required.

OnStorageOpening

Event raised just before the storage menu opens for a storage entity. Use this event to update the display name or perform other pre-open actions.

```
public static event Action<StorageEventArgs> OnStorageOpening
```

Event Type

[Action<StorageEventArgs>](#)

Examples

```
// Ensure custom names are displayed when opening storage
StorageEvents.OnStorageOpening += (args) =>
{
    args.Storage.SyncCustomNameToDisplayName();
};
```

Remarks

This event is useful for syncing custom names (set via clipboard) to the display name. The base game's StorageMenu uses StorageEntityName for display, but custom names are stored separately in Configuration.Name. Subscribe to this event and call [SyncCustomNameToDisplayName\(\)](#) to ensure custom names are shown.

Class StorageLoadingEventArgs

Namespace: [S1API.Storage](#)

Assembly: S1API.dll

Event arguments for storage loading operations. Raised when storage is being loaded from a save file.

```
public class StorageLoadingEventArgs : StorageEventArgs
```

Inheritance

[object](#) ← [StorageEventArgs](#) ← StorageLoadingEventArgs

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

AdditionalSlotsNeeded

Number of additional slots needed to fit all items. Returns 0 if no additional slots are needed.

```
public int AdditionalSlotsNeeded { get; }
```

Property Value

[int](#)

CurrentSlotCount

Current slot count before loading.

```
public int CurrentSlotCount { get; }
```

Property Value

[int](#) ↗

ItemCountBeingLoaded

Number of items being loaded into this storage. Use this to determine if slot expansion is needed.

```
public int ItemCountBeingLoaded { get; }
```

Property Value

[int](#) ↗

NeedsMoreSlots

Whether additional slots are needed to fit all items.

```
public bool NeedsMoreSlots { get; }
```

Property Value

[bool](#) ↗

Namespace S1API.Storages

Classes

[StorageInstance](#)

Represents a storage container in-game.

[StorageManager](#)

Provides methods for managing and retrieving storage containers within the game.

Enums

[StorageAccessSettings](#)

Access settings for storage containers.

Enum StorageAccessSettings

Namespace: [S1API.Storages](#)

Assembly: S1API.dll

Access settings for storage containers.

```
public enum StorageAccessSettings
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

`Closed = 0`

Storage is closed and cannot be accessed.

`Full = 2`

Storage can be accessed by all players.

`SinglePlayerOnly = 1`

Storage can only be accessed by one player at a time.

Class StorageInstance

Namespace: [S1API.Storages](#)

Assembly: S1API.dll

Represents a storage container in-game.

```
public class StorageInstance
```

Inheritance

[object](#) ← StorageInstance

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Remarks

This is the user-facing layer for interacting with an existing storage instance: reading metadata (name, subtitle, slots/items), checking fit, adding/removing items, and subscribing to open/close/content-change events. It intentionally does not mutate slot topology or placeable wiring; use [StorageEntity](#) for capacity and structural changes.

Properties

AccessSettings

The access settings for this storage container.

```
public StorageAccessSettings AccessSettings { get; set; }
```

Property Value

[StorageAccessSettings](#)

IsOpened

Whether this storage container is currently opened by a player.

```
public bool IsOpened { get; }
```

Property Value

[bool](#) ↗

ItemCount

The current number of items stored in this storage container.

```
public int ItemCount { get; }
```

Property Value

[int](#) ↗

Name

The display name of this storage container.

```
public string Name { get; }
```

Property Value

[string](#) ↗

SlotCount

The total number of slots available in this storage container.

```
public int SlotCount { get; }
```

Property Value

[int](#)

Slots

An array of all slots available on the storage container.

```
public ItemSlotInstance[] Slots { get; }
```

Property Value

[ItemSlotInstance](#)[]

Subtitle

The subtitle of this storage container.

```
public string Subtitle { get; }
```

Property Value

[string](#)

Methods

AddItem(ItemInstance)

Adds an item instance to this storage container.

```
public void AddItem(ItemInstance itemInstance)
```

Parameters

itemInstance [ItemInstance](#)

The item instance you want to store.

CanItemFit(ItemInstance, int)

Whether an item can fit inside this storage container or not.

```
public bool CanItemFit(ItemInstance itemInstance, int quantity = 1)
```

Parameters

itemInstance [ItemInstance](#)

The item instance you want to store.

quantity [int](#)

The quantity of item you want to store.

Returns

[bool](#)

Whether the item will fit or not.

FromGameObject(GameObject)

Gets a storage instance from a GameObject that has a StorageEntity component.

```
public static StorageInstance? FromGameObject(GameObject gameobject)
```

Parameters

gameobject GameObject

The GameObject to check for a StorageEntity component.

Returns

[StorageInstance](#)

A StorageInstance wrapper if found, otherwise null.

Remarks

This method first checks the GameObject itself, then searches up the parent hierarchy. Useful for getting storage from spawned prefabs like Display Cabinets, Wall Shelves, etc.

FromGameObjectInChildren(GameObject)

Gets a storage instance from a GameObject that has a StorageEntity component in its children.

```
public static StorageInstance? FromGameObjectInChildren(GameObject gameobject)
```

Parameters

gameobject GameObject

The GameObject to search.

Returns

[StorageInstance](#)

A StorageInstance wrapper if found, otherwise null.

Remarks

This method searches the GameObject and all its children for a StorageEntity component. Use this when the storage component might be on a child object.

GetContentsDictionary()

Gets a dictionary mapping item instances to their quantities in this storage container.

```
public Dictionary<ItemInstance, int> GetContentsDictionary()
```

Returns

[Dictionary](#)<[ItemInstance](#), [int](#)>

A dictionary where keys are item instances and values are quantities.

GetItems()

Gets all item instances currently stored in this storage container.

```
public ItemInstance[] GetItems()
```

Returns

[ItemInstance](#)[]

An array of item instances.

RemoveAllOfDefinition(string)

Removes all items matching the given item definition ID from this storage container. This performs a soft removal (decrements quantity) and does not spawn world items.

```
public int RemoveAllOfDefinition(string itemDefinitionId)
```

Parameters

itemDefinitionId [string](#)

The ID of the item definition to remove.

Returns

[int](#)

The total quantity that was removed.

RemoveItem(ItemInstance)

Removes a specific item instance from this storage container. This performs a soft removal (decrements quantity) and does not spawn world items.

```
public int RemoveItem(ItemInstance itemInstance)
```

Parameters

itemInstance [ItemInstance](#)

The item instance to remove.

Returns

[int](#)

The quantity that was removed.

TryRemoveQuantity(string, int)

Attempts to remove a specific quantity of items matching the given item definition ID. This performs a soft removal (decrements quantity) and does not spawn world items.

```
public int TryRemoveQuantity(string itemDefinitionId, int quantity)
```

Parameters

itemDefinitionId [string](#)

The ID of the item definition to remove.

quantity [int](#)

The quantity to remove.

Returns

[int](#)

The actual quantity that was removed (may be less than requested if insufficient items exist).

Events

OnClosed

An action fired when the storage container is closed by the player.

```
public event Action OnClosed
```

Event Type

[Action](#)

OnContentsChanged

An action fired when the contents of the storage container change (items added or removed).

```
public event Action OnContentsChanged
```

Event Type

[Action ↗](#)

OnOpened

An action fired when the storage container is opened by the player.

```
public event Action OnOpened
```

Event Type

[Action ↗](#)

Class StorageManager

Namespace: [S1API.Storages](#)

Assembly: S1API.dll

Provides methods for managing and retrieving storage containers within the game.

```
public static class StorageManager
```

Inheritance

[object](#) ← StorageManager

Inherited Members

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

Methods

FindByName(string)

Finds a storage entity by its display name.

```
public static StorageInstance? FindByName(string name)
```

Parameters

name [string](#)

The display name of the storage entity to find.

Returns

[StorageInstance](#)

A storage instance if found; otherwise, null.

FindByPredicate(Func<StorageInstance, bool>)

Finds storage entities matching a given predicate.

```
public static StorageInstance[] FindByPredicate(Func<StorageInstance,  
bool> predicate)
```

Parameters

predicate [Func<StorageInstance, bool>](#)

A function to test each storage entity for a condition.

Returns

[StorageInstance\[\]](#)

An array of storage instances that match the predicate.

GetAll()

Gets all world storage entities currently in the game.

```
public static StorageInstance[] GetAll()
```

Returns

[StorageInstance\[\]](#)

An array of storage instances representing all world storage entities.

Namespace S1API.UI

Classes

[CharacterCreatorManager](#)

Provides programmatic control over the in-game character creator UI system. Allows mods to open the character creator, listen for completion events, and retrieve customized avatar settings.

[MainMenuRig](#)

Modder-facing wrapper for the MainMenuRig component in the main menu scene. Provides access to the menu avatar and related UI components without exposing game types.

[UIFactory](#)

Utility class for constructing and configuring various UI elements in Unity.

Class CharacterCreatorManager

Namespace: [S1API.UI](#)

Assembly: S1API.dll

Provides programmatic control over the in-game character creator UI system. Allows mods to open the character creator, listen for completion events, and retrieve customized avatar settings.

```
public static class CharacterCreatorManager
```

Inheritance

[object](#) ← CharacterCreatorManager

Inherited Members

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

Properties

ActiveSettings

The current avatar settings being edited in the character creator. Returns null if the creator is not open.

```
public static BasicAvatarSettings ActiveSettings { get; }
```

Property Value

[BasicAvatarSettings](#)

IsOpen

Whether the character creator is currently open and active.

```
public static bool IsOpen { get; }
```

Property Value

[bool](#) ↗

Methods

Close()

Closes the character creator without saving changes.

```
public static void Close()
```

Complete()

Completes the character customization and closes the creator. Fires the OnCompleted event with the final settings. Camera restoration is handled in the OnCreatorCompleted callback.

```
public static void Complete()
```

GetAvailablePresets()

Gets a list of available preset names.

```
public static string[] GetAvailablePresets()
```

Returns

[string](#) ↗ []

An array of preset names available in the character creator.

Open(BasicAvatarSettings, bool)

Opens the character creator with the specified initial settings.

```
public static void Open(BasicAvatarSettings initialSettings = null, bool showUI = true)
```

Parameters

initialSettings [BasicAvatarSettings](#)

Optional initial avatar settings. If null, player's current avatar settings are loaded, or default settings if player has none.

showUI [bool](#)

Whether to display the UI. Set to false to customize programmatically without showing UI.

PreRegisterAsActiveUI()

Pre-registers the character creator as an active UI element to prevent other systems (like dialogue) from restoring the camera. Call this before ending dialogue or other UI systems to ensure smooth camera transitions.

```
public static void PreRegisterAsActiveUI()
```

SelectPreset(string)

Selects a preset by name from the available presets.

```
public static void SelectPreset(string presetName)
```

Parameters

presetName [string](#)

The name of the preset to select.

SetRigRotation(float)

Rotates the character rig in the character creator.

```
public static void SetRigRotation(float normalizedValue)
```

Parameters

normalizedValue [float](#)

Rotation value (0.0 to 1.0), where 0.0 is 0 degrees and 1.0 is 359 degrees.

Events

OnClosed

Fired when the character creator is closed without completion.

```
public static event Action OnClosed
```

Event Type

[Action](#)

OnCompleted

Fired when character customization is completed successfully.

```
public static event Action<BasicAvatarSettings> OnCompleted
```

Event Type

[Action](#)<[BasicAvatarSettings](#)>

Remarks

The BasicAvatarSettings parameter contains the finalized character configuration.

OnOpened

Fired when the character creator is opened.

```
public static event Action OnOpened
```

Event Type

[Action](#)

Class MainMenuRig

Namespace: [S1API.UI](#)

Assembly: S1API.dll

Modder-facing wrapper for the MainMenuRig component in the main menu scene. Provides access to the menu avatar and related UI components without exposing game types.

```
public sealed class MainMenuRig
```

Inheritance

[object](#) ← MainMenuRig

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Properties

Avatar

The avatar displayed in the main menu.

```
public Avatar? Avatar { get; }
```

Property Value

[Avatar](#)

Methods

FindInScene(bool)

Finds all MainMenuRig instances in the current scene.

```
public static MainMenuRig[] FindInScene(bool includeInactive = false)
```

Parameters

`includeInactive` [bool](#)

Whether to include inactive GameObjects in the search.

Returns

[MainMenuRig](#)[]

An array of MainMenuRig wrappers found in the scene.

Class UIFactory

Namespace: [S1API.UI](#)

Assembly: S1API.dll

Utility class for constructing and configuring various UI elements in Unity.

```
public static class UIFactory
```

Inheritance

[object](#) ← UIFactory

Inherited Members

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

Remarks

Provides static helpers for building complex hierarchies—panels, text, layouts, scroll views, buttons, etc.—while handling RectTransform setup, layout components, and consistent styling in one place.

Methods

BindAcceptButton(Button, Text, string, UnityAction)

Updates a button's display text and binds the supplied click callback.

```
public static void BindAcceptButton(Button btn, Text label, string text,  
UnityAction callback)
```

Parameters

btn Button

Button to configure.

label Text

Text component associated with the button.

text [string](#)

Text to assign to the label.

callback UnityAction

Delegate invoked when the button is clicked.

ButtonRow(string, Transform, float, TextAnchor)

Creates a container configured with a UnityEngine.UI.HorizontalLayoutGroup for arranging buttons in a row.

```
public static GameObject ButtonRow(string name, Transform parent, float spacing = 12, TextAnchor alignment = 4)
```

Parameters

name [string](#)

Name of the row GameObject.

parent Transform

Transform that will hold the row.

spacing [float](#)

Spacing between children in the layout (defaults to 12).

alignment TextAnchor

Child alignment; defaults to UnityEngine.TextAnchor.MiddleCenter.

Returns

GameObject

The created row GameObject.

ButtonWithLabel(string, string, Transform, Color, float, float)

Creates a rectangular button with a centered text label and returns useful components.

```
public static (GameObject, Button, Text) ButtonWithLabel(string name, string label,  
Transform parent, Color bgColor, float Width, float Height)
```

Parameters

name [string](#)

Name assigned to the button GameObject.

label [string](#)

Text displayed in the button.

parent [Transform](#)

Parent transform for the button.

bgColor [Color](#)

Background color for the button image.

Width [float](#)

Desired width in pixels.

Height [float](#)

Desired height in pixels.

Returns

(GameObject, Button, Text)

A tuple of (GameObject, Button, Text) for further configuration.

ClearChildren(Transform)

Destroys all child GameObjects of the provided parent Transform.

```
public static void ClearChildren(Transform parent)
```

Parameters

parent Transform

Transform to clear.

CreateQuestRow(string, Transform, out GameObject, out GameObject)

Constructs a quest row entry with dedicated icon/text panels and common layout settings.

```
public static GameObject CreateQuestRow(string name, Transform parent, out  
GameObject iconPanel, out GameObject textPanel)
```

Parameters

name string

Name suffix applied to generated GameObjects.

parent Transform

Parent transform for the row.

iconPanel GameObject

Outputs the panel intended for quest icons.

textPanel GameObject

Outputs the panel intended for quest title/description.

Returns

GameObject

The fully constructed row GameObject.

CreateRowButton(GameObject, UnityAction, bool)

Adds a UnityEngine.UI.Button component to a row container and wires up click/interactivity.

```
public static void CreateRowButton(GameObject go, UnityAction clickHandler,  
bool enabled)
```

Parameters

go GameObject

Target GameObject that already has an UnityEngine.UI.Image for visuals.

clickHandler UnityAction

Callback to invoke on button click.

enabled bool

Whether the resulting button should start as interactable.

CreateTextBlock(Transform, string, string, bool)

Creates a stacked title/subtitle block and optionally appends a completed label.

```
public static void CreateTextBlock(Transform parent, string title, string subtitle,  
bool isCompleted)
```

Parameters

parent Transform

Transform that will receive the text elements.

title [string](#)

Primary heading displayed in bold.

subtitle [string](#)

Secondary descriptive text below the title.

isCompleted [bool](#)

Adds an "Already Delivered" status line when true.

FitContentHeight(RectTransform)

Ensures a content UnityEngine.RectTransform grows tall enough to fit its children.

```
public static void FitContentHeight(RectTransform content)
```

Parameters

content RectTransform

The RectTransform whose vertical size should adapt to its children.

Remarks

Adds a UnityEngine.UIContentSizeFitter if one is missing and configures it for preferred vertical sizing.

HorizontalLayoutOnGO(GameObject, int, int, int, int, int, TextAnchor)

Adds a UnityEngine.UI.HorizontalLayoutGroup to a GameObject and preconfigures its sizing behavior.

```
public static void HorizontalLayoutOnGO(GameObject go, int spacing = 10, int paddingLeft = 0, int paddingRight = 0, int paddingTop = 0, int paddingBottom = 0, TextAnchor alignment = 4)
```

Parameters

go GameObject

GameObject that should host the layout group.

spacing [int ↗](#)

Spacing between children; defaults to 10.

padLeft [int ↗](#)

Left padding value.

padRight [int ↗](#)

Right padding value.

padTop [int ↗](#)

Top padding value.

padBottom [int ↗](#)

Bottom padding value.

alignment TextAnchor

Child alignment; defaults to UnityEngine.TextAnchor.MiddleCenter.

Panel(string, Transform, Color, Vector2?, Vector2?, bool)

Creates a UI panel GameObject with an Image background and configurable anchoring.

```
public static GameObject Panel(string name, Transform parent, Color bgColor,
Vector2? anchorMin = null, Vector2? anchorMax = null, bool fullAnchor = false)
```

Parameters

name [string ↗](#)

Name of the panel GameObject.

parent Transform

Transform that becomes the parent of the panel.

bgColor Color

Color applied to the panel background.

anchorMin Vector2?

Optional minimum anchor; defaults to centered anchor.

anchorMax Vector2?

Optional maximum anchor; defaults to centered anchor.

fullAnchor bool ↗

When true, stretches the panel to fill its parent regardless of anchor arguments.

Returns

GameObject

The created panel GameObject (with RectTransform/Image components attached).

RoundedButtonWithLabel(string, string, Transform, Color, float, float, int, Color)

Creates a rounded button composed of a mask container, button, and centered text label.

```
public static (GameObject, Button, Text) RoundedButtonWithLabel(string name, string label, Transform parent, Color bgColor, float width, float height, int fontSize, Color textColor)
```

Parameters

name string ↗

Name of the underlying button GameObject.

label string ↗

Text shown inside the button.

parent Transform

Parent transform for the mask container.

bgColor Color

Background color applied to the inner button.

width float

Preferred width for the control (also applied to a LayoutElement).

height float

Preferred height for the control.

fontSize int

Font size for the label.

textColor Color

Color of the label text.

Returns

(GameObject, Button, Text)

The tuple (mask container GameObject, Button component, Text component).

ScrollableVerticalList(string, Transform, out ScrollRect)

Builds a ScrollRect hierarchy configured for vertical scrolling and returns the content RectTransform.

```
public static RectTransform ScrollableVerticalList(string name, Transform parent,  
out ScrollRect scrollRect)
```

Parameters

name string

Name of the root scroll view GameObject.

parent Transform

Transform that will own the scroll view.

scrollRect ScrollRect

Outputs the created UnityEngine.UI.ScrollRect component.

Returns

RectTransform

The RectTransform of the content container where list items should be added.

SetIcon(Sprite, Transform)

Sets an icon as a child of the specified parent transform with the given sprite.

```
public static void SetIcon(Sprite sprite, Transform parent)
```

Parameters

sprite Sprite

The sprite to be used as the icon.

parent Transform

The transform that will act as the parent of the icon.

SetLayoutGroupPadding(LayoutGroup, int, int, int, int)

Convenience helper to configure the padding on any UnityEngine.UI.LayoutGroup.

```
public static void SetLayoutGroupPadding(LayoutGroup layoutGroup, int left, int right, int top, int bottom)
```

Parameters

layoutGroup LayoutGroup

Target layout group.

left int ↗

Left padding value.

right int ↗

Right padding value.

top int ↗

Top padding value.

bottom int ↗

Bottom padding value.

Text(string, string, Transform, int, TextAnchor, FontStyle)

Creates a `UnityEngine.UI.Text` element configured with the supplied content and styling.

```
public static Text Text(string name, string content, Transform parent, int fontSize  
= 14, TextAnchor anchor = 0, FontStyle style = 0)
```

Parameters

name string ↗

Name of the GameObject to create.

content string ↗

Initial string displayed inside the text component.

parent Transform

Transform that will contain the new text element.

fontSize [int](#)

Font size to apply, defaults to 14.

anchor [TextAnchor](#)

Text alignment; defaults to UnityEngine.TextAnchor.UpperLeft.

style [FontStyle](#)

Font style flag; defaults to UnityEngine.FontStyle.Normal.

Returns

Text

The configured [Text\(string, string, Transform, int, TextAnchor, FontStyle\)](#) component.

TopBar(string, Transform, string, float, int, int, int, int)

Creates a top-bar container with padding, title text, and layout metadata.

```
public static GameObject TopBar(string name, Transform parent, string title, float topbarSize, int paddingLeft, int paddingRight, int paddingTop, int paddingBottom)
```

Parameters

name [string](#)

Name of the bar GameObject.

parent [Transform](#)

Transform that will contain the bar.

title [string](#)

Display text shown in the bar.

topbarSize [float](#)

Normalized height (Y anchor) reserved for the bar.

paddingLeft [int ↗](#)

Left padding applied by the layout group.

paddingRight [int ↗](#)

Right padding applied by the layout group.

paddingTop [int ↗](#)

Top padding applied by the layout group.

paddingBottom [int ↗](#)

Bottom padding applied by the layout group.

Returns

GameObject

The instantiated bar GameObject.

VerticalLayoutOnGO(GameObject, int, RectOffset?)

Adds and configures a UnityEngine.UI.VerticalLayoutGroup on the supplied GameObject.

```
public static void VerticalLayoutOnGO(GameObject go, int spacing = 10, RectOffset?  
padding = null)
```

Parameters

go **GameObject**

GameObject that should host the layout.

spacing [int ↗](#)

Spacing between children in pixels (default 10).

padding **RectOffset**

Optional padding override; defaults to 10px on every side.

Namespace S1API.Utils

Classes

[ArrayExtensions](#)

Extensions for Arrays. This class is intended for public use by mod developers.

[ButtonUtils](#)

Utility helpers for managing Unity UI UnityEngine.UI.Buttons. This class is intended for public use by mod developers.

[ColorUtils](#)

Utilities for the UnityEngine.Color class. This class is intended for public use by mod developers.

[ImageUtils](#)

A utility class to assist with loading images into the game. Useful for icons such as on phone apps, custom NPCs, quests, etc. This class is intended for public use by mod developers.

[RandomUtils](#)

A utility class providing random selection functionality for lists and numeric ranges. This class is intended for public use by mod developers.

[ReflectionUtils](#)

Provides reflection-based utility methods for mod developers.

[ToggleUtils](#)

Utilities for subscribing to and managing Toggle value change events in a cross-compatible way between Mono and IL2CPP. Handles Unity versions where Toggle.onValueChanged is exposed as either a field or a property. This class is intended for public use by mod developers.

Class ArrayExtensions

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

Extensions for Arrays. This class is intended for public use by mod developers.

```
public static class ArrayExtensions
```

Inheritance

[object](#) ← ArrayExtensions

Inherited Members

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

Methods

AddItemToArray<T>(T[], T)

Add's an item to an existing array

```
public static T[] AddItemToArray<T>(this T[]? array, T item)
```

Parameters

array T[]

item T

Returns

T[]

Type Parameters

Class ButtonUtils

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

Utility helpers for managing Unity UI UnityEngine.UI.Buttons. This class is intended for public use by mod developers.

```
public static class ButtonUtils
```

Inheritance

[object](#) ← ButtonUtils

Inherited Members

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

Methods

AddListener(Button, Action)

Adds a click listener to the specified button, ensuring compatibility with IL2CPP and Mono.

```
public static void AddListener(Button button, Action action)
```

Parameters

button Button

action [Action](#)

ClearListeners(Button)

Removes all listeners from the specified button safely.

```
public static void ClearListeners(Button button)
```

Parameters

button Button

Disable(Button, Text?, string?)

Disables the button and optionally updates the label.

```
public static void Disable(Button button, Text? label = null, string? text = null)
```

Parameters

button Button

label Text

text [string](#)

Enable(Button, Text?, string?)

Enables the button and optionally updates the label.

```
public static void Enable(Button button, Text? label = null, string? text = null)
```

Parameters

button Button

label Text

text [string](#)

RemoveListener(Button, Action)

Removes a previously added click listener from the specified button.

```
public static void RemoveListener(Button button, Action action)
```

Parameters

button Button

action [Action](#)

SetLabel(Text, string)

Sets the label text of a button with a known Text child.

```
public static void SetLabel(Text label, string text)
```

Parameters

label Text

text [string](#)

SetStyle(Button, Text, string, Color)

Sets the button label and background color.

```
public static void SetStyle(Button button, Text label, string text, Color bg)
```

Parameters

button Button

label Text

text [string ↗](#)

bg Color

Class ColorUtils

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

Utilities for the UnityEngine.Color class. This class is intended for public use by mod developers.

```
public static class ColorUtils
```

Inheritance

[object](#) ← ColorUtils

Inherited Members

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

Methods

ToColor(uint)

Convert's a [int](#) value to UnityEngine.Color

```
public static Color ToColor(this uint hexColor)
```

Parameters

hexColor [uint](#)

Returns

Color

Class ImageUtils

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

A utility class to assist with loading images into the game. Useful for icons such as on phone apps, custom NPCs, quests, etc. This class is intended for public use by mod developers.

```
public static class ImageUtils
```

Inheritance

[object](#) ← ImageUtils

Inherited Members

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

Methods

LoadImage(string)

Loads an image from the specified file path and converts it into a Sprite object.

```
public static Sprite? LoadImage(string fileName)
```

Parameters

fileName [string](#)

The name of the file (with path) containing the image to load.

Returns

Sprite

A Sprite object representing the loaded image, or null if the image could not be loaded or the file does not exist.

LoadImageFromResource(Assembly, string, float, FilterMode)

Loads an image from an embedded resource stream and converts it into a Sprite object.

```
public static Sprite? LoadImageFromResource(Assembly assembly, string resourceName,  
float pixelsPerUnit = 100, FilterMode filterMode = 1)
```

Parameters

assembly [Assembly](#)

The assembly containing the embedded resource.

resourceName [string](#)

The fully qualified name of the embedded resource (e.g., "Namespace.Assets.Icon.png").

pixelsPerUnit [float](#)

The pixels per unit for the sprite. Defaults to 100f.

filterMode [FilterMode](#)

The filter mode for the texture. Defaults to FilterMode.Bilinear.

Returns

Sprite

A Sprite object representing the loaded image, or null if the resource could not be found or loaded.

LoadImageRaw(byte[])

Loads an image from a byte array and converts it into a Sprite object.

```
public static Sprite? LoadImageRaw(byte[] data)
```

Parameters

data [byte](#)[]

The byte array containing the image data to load.

Returns

Sprite

A Sprite object representing the loaded image, or null if the image could not be loaded.

TextureToSprite(Texture2D?, float)

Converts a Texture2D to a Sprite.

```
public static Sprite? TextureToSprite(Texture2D? texture, float pixelsPerUnit = 100)
```

Parameters

texture Texture2D

The texture to convert.

pixelsPerUnit [float](#)

The pixels per unit for the sprite. Defaults to 100f.

Returns

Sprite

A Sprite object, or null if texture is null.

Class RandomUtils

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

A utility class providing random selection functionality for lists and numeric ranges. This class is intended for public use by mod developers.

```
public static class RandomUtils
```

Inheritance

[object](#) ← RandomUtils

Inherited Members

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

Methods

PickMany<T>(IList<T>, int)

Returns a specified number of unique random elements from a list. If the count exceeds the number of available elements, returns all elements in random order.

```
public static List<T> PickMany<T>(this IList<T> list, int count)
```

Parameters

list [IList](#)<T>

The list of items to pick from.

count [int](#)

The number of random items to pick.

Returns

[List](#)<T>

A list containing the selected random items, or an empty list if the input list is null or empty.

Type Parameters

T

PickOne<T>(IList<T>)

Returns a random element from the provided list, or the default value of type T if the list is null or empty.

```
public static T PickOne<T>(this IList<T> list)
```

Parameters

list [IList](#)<T>

The list from which to select a random element.

Returns

T

A randomly selected element from the list, or the default value of type T if the list is null or empty.

Type Parameters

T

PickUnique<T>(IList<T>, Func<T, bool>, int)

Returns a random element from a list that satisfies the given condition, with a maximum number of attempts. If no such element can be found within the allowed attempts, returns the default value of the

type.

```
public static T PickUnique<T>(this IList<T> list, Func<T, bool> isDuplicate, int maxTries = 10)
```

Parameters

list [IList](#)<T>

The list of items to pick from.

isDuplicate [Func](#)<T, bool>

A function to determine if the selected item satisfies the duplicate condition.

maxTries [int](#)

The maximum number of attempts to find a valid item.

Returns

T

A randomly selected item that satisfies the condition, or the default value of the type if no valid item is found.

Type Parameters

T

RangeInt(int, int)

Generates a random integer within the specified range.

```
public static int RangeInt(int minInclusive, int maxExclusive)
```

Parameters

`minInclusive` [int](#)

The inclusive lower bound of the random number.

`maxExclusive` [int](#)

The exclusive upper bound of the random number.

Returns

[int](#)

A random integer greater than or equal to `minInclusive` and less than `maxExclusive`.

Class ReflectionUtils

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

Provides reflection-based utility methods for mod developers.

```
public static class ReflectionUtils
```

Inheritance

[object](#) ← ReflectionUtils

Inherited Members

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

Methods

GetMethod(Type?, string, BindingFlags)

Recursively searches for a method by name from a class down to the object type.

```
public static MethodInfo? GetMethod(Type? type, string methodName,  
BindingFlags bindingFlags)
```

Parameters

type [Type](#)

The type you want to recursively search.

methodName [string](#)

The name of the method you're searching for.

bindingFlags [BindingFlags](#)

The binding flags to apply during the search.

Returns

[MethodInfo](#)

The method info if found, otherwise null.

GetValueTupleItems(object)

Retrieves the items from the ValueTuple instance.

```
public static object[]? GetValueTupleItems(this object obj)
```

Parameters

[obj](#) [object](#)

The ValueTuple instance.

Returns

[object](#)[]

The items in the ValueTuple instance.

IsValueTuple(object)

Checks whether the object is a ValueTuple.

```
public static bool IsValueTuple(this object obj)
```

Parameters

[obj](#) [object](#)

The object type to check.

Returns

[bool](#)

Whether the type is a ValueTuple or not.

TryGetFieldOrProperty(object, string)

Attempts to get a field or property value from an object using reflection. Tries field first, then property. Handles both public and non-public members.

```
public static object? TryGetFieldOrProperty(object target, string memberName)
```

Parameters

[target](#) [object](#)

The target object to get the member from.

[memberName](#) [string](#)

The name of the field or property.

Returns

[object](#)

The value of the member, or [null](#) if not found or inaccessible.

TrySetFieldOrProperty(object, string, object)

Attempts to set a field or property value on an object using reflection. Tries field first, then property. Handles both public and non-public members.

```
public static bool TrySetFieldOrProperty(object target, string memberName,  
object value)
```

Parameters

target [object](#)

The target object to set the member on.

memberName [string](#)

The name of the field or property.

value [object](#)

The value to set.

Returns

[bool](#)

true if the member was successfully set; otherwise, **false**.

Class ToggleUtils

Namespace: [S1API.Utils](#)

Assembly: S1API.dll

Utilities for subscribing to and managing Toggle value change events in a cross-compatible way between Mono and IL2CPP. Handles Unity versions where Toggle.onValueChanged is exposed as either a field or a property. This class is intended for public use by mod developers.

```
public static class ToggleUtils
```

Inheritance

[object](#) ← ToggleUtils

Inherited Members

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

Methods

AddListener(Toggle, Action<bool>)

Adds a listener to a Toggle's onValueChanged event in an IL2CPP-safe manner.

```
public static void AddListener(Toggle toggle, Action<bool> listener)
```

Parameters

`toggle` Toggle

`listener` Action<bool>

ClearListeners(Toggle)

Removes all listeners from a Toggle's onValueChanged event.

```
public static void ClearListeners(Toggle toggle)
```

Parameters

toggle Toggle

GetGraphic(Toggle)

Gets the Toggle's checkmark graphic in a version-agnostic manner.

```
public static Graphic? GetGraphic(Toggle toggle)
```

Parameters

toggle Toggle

Returns

Graphic

RemoveListener(Toggle, Action<bool>)

Removes a previously added listener from a Toggle's onValueChanged event.

```
public static void RemoveListener(Toggle toggle, Action<bool> listener)
```

Parameters

toggle Toggle

listener Action<bool>

SetGraphic(Toggle, Graphic)

Sets the Toggle's checkmark graphic in a version-agnostic manner (field or property).

```
public static void SetGraphic(Toggle toggle, Graphic graphic)
```

Parameters

`toggle` Toggle

`graphic` Graphic

Namespace S1API.Vehicles

Classes

[LandVehicle](#)

Represents a land vehicle in the game.

[VehicleRegistry](#)

Registry utilities for discovering and resolving vehicles without exposing game types.

Enums

[ParkingAlignment](#)

Parking alignment type used by S1API. Maps to game enum internally.

[VehicleColor](#)

Represents available colors for vehicles.

Class LandVehicle

Namespace: [S1API.Vehicles](#)

Assembly: S1API.dll

Represents a land vehicle in the game.

```
public class LandVehicle
```

Inheritance

[object](#) ← LandVehicle

Inherited Members

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

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

LandVehicle(string)

Creates a new LandVehicle instance.

```
public LandVehicle(string vehicleCode)
```

Parameters

vehicleCode [string](#)

Properties

Color

Vehicle's color.

```
public VehicleColor Color { get; set; }
```

Property Value

[VehicleColor](#)

GUID

Unique GUID string for this vehicle.

```
public string GUID { get; }
```

Property Value

[string](#)

IsOccupied

If this vehicle has any occupants

```
public bool IsOccupied { get; set; }
```

Property Value

[bool](#)

IsPlayerOwned

If the vehicle is owned by the player.

```
public bool IsPlayerOwned { get; set; }
```

Property Value

[bool](#) ↗

Storage

Trunk space

```
public StorageInstance Storage { get; }
```

Property Value

[StorageInstance](#)

TopSpeed

Vehicle's top speed.

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

Property Value

[float](#) ↗

VehiclePrice

Vehicle price.

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

Property Value

[float](#)

Methods

AlignTo(Transform, ParkingAlignment, bool)

```
public void AlignTo(Transform target, ParkingAlignment type, bool network = false)
```

Parameters

target Transform

type [ParkingAlignment](#)

network bool

ApplyColor(VehicleColor)

```
public void ApplyColor(VehicleColor col)
```

Parameters

col [VehicleColor](#)

DestroyVehichle()

Deletes the land vehicle

```
public void DestroyVehichle()
```

ExitPark(bool)

Exit parking spot, and optionally, the parking lot

```
public void ExitPark(bool moveToPoint = true)
```

Parameters

moveToPoint [bool](#)

Park(ParkingData, bool)

Parks the vehicle in the specified slot [ParkingData](#)

```
public void Park(ParkingData parkData, bool network)
```

Parameters

parkData [ParkingData](#)

network [bool](#)

SetVisible(bool)

Set this vehicle as visible or not

```
public void SetVisible(bool vis)
```

Parameters

vis [bool](#)

Spawn(Vector3, Quaternion)

Spawns the vehicle in the game world.

```
public void Spawn(Vector3 position, Quaternion rotation)
```

Parameters

position Vector3

Position in the world

rotation Quaternion

Rotation of the vehicle

Events

OnCollision

When this vehicle has collided with something

```
public event Action<Collision> OnCollision
```

Event Type

[Action](#)<Collision>

OnHandbrakeApplied

When the handbrake has been applied

```
public event Action OnHandbrakeApplied
```

Event Type

[Action](#)

OnVehicleStart

When this vehicle has started

```
public event Action OnVehicleStart
```

Event Type

[Action](#)

OnVehicleStop

When this vehicle has stopped

```
public event Action OnVehicleStop
```

Event Type

[Action](#)

Enum ParkingAlignment

Namespace: [S1API.Vehicles](#)

Assembly: S1API.dll

Parking alignment type used by S1API. Maps to game enum internally.

```
public enum ParkingAlignment
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

FrontToKerb = 0

RearToKerb = 1

Enum VehicleColor

Namespace: [S1API.Vehicles](#)

Assembly: S1API.dll

Represents available colors for vehicles.

```
public enum VehicleColor
```

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Fields

Black = 0

Custom = 16

Cyan = 13

DarkBlue = 11

DarkGreen = 15

DarkGrey = 1

DullRed = 7

LightBlue = 12

LightGreen = 14

LightGrey = 2

Navy = 10

Orange = 5

Pink = 8

Purple = 9

Red = 6

White = 3

Yellow = 4

Class VehicleRegistry

Namespace: [S1API.Vehicles](#)

Assembly: S1API.dll

Registry utilities for discovering and resolving vehicles without exposing game types.

```
public static class VehicleRegistry
```

Inheritance

[object](#) ← VehicleRegistry

Inherited Members

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

Methods

CreateVehicle(string)

Creates a new vehicle instance using a vehicle code and returns a wrapper. Useful when you need a vehicle that doesn't exist yet.

```
public static LandVehicle CreateVehicle(string vehicleCode)
```

Parameters

vehicleCode [string](#)

The vehicle code to spawn (e.g., "Sedan", "SUV", etc.).

Returns

[LandVehicle](#)

A new vehicle wrapper, or null if creation fails.

GetAll()

Returns all currently spawned land vehicles wrapped for modder use.

```
public static LandVehicle[] GetAll()
```

Returns

[LandVehicle](#) []

GetByGUID(string)

Finds a vehicle by GUID string. Returns null if not found.

```
public static LandVehicle GetByGUID(string guid)
```

Parameters

guid [string](#)

Returns

[LandVehicle](#)

GetByName(string)

Finds a vehicle by GameObject name. Useful when vehicles aren't spawned yet. Falls back to finding in all vehicles if GameObject.Find fails.

```
public static LandVehicle GetByName(string gameObjectName)
```

Parameters

gameObjectName [string](#)

The name of the GameObject containing the vehicle.

Returns

[LandVehicle](#)

A vehicle wrapper, or null if not found.

RemoveVehicle(string)

Removes a vehicle from the game's lists, for permanently destroying a vehicle. This will be called automatically in OnDestroy, so there's likely no need to call this manually.

```
public static void RemoveVehicle(string guidString)
```

Parameters

guidString [string](#)

The GUID string of the vehicle to remove.

Class MyPluginInfo

Namespace: [S1API](#)

Assembly: S1API.dll

```
public static class MyPluginInfo
```

Inheritance

[object](#) ← MyPluginInfo

Inherited Members

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

Fields

PLUGIN_GUID

```
public const string PLUGIN_GUID = "S1API"
```

Field Value

[string](#)

PLUGIN_NAME

```
public const string PLUGIN_NAME = "S1API"
```

Field Value

[string](#)

PLUGIN_VERSION

```
public const string PLUGIN_VERSION = "2.9.1"
```

Field Value

[string](#) ↗

Class S1API

Namespace: [S1API](#)

Assembly: S1API.dll

S1API root MelonMod. Provides lifecycle hooks for internal systems.

```
public class S1API : MelonMod
```

Inheritance

[Object](#) ← MelonBase ← MelonTypeBase<MelonMod> ← MelonMod ← S1API

Inherited Members

[MelonMod.OnLevelWasLoaded\(int\)](#) , [MelonMod.OnLevelWasInitialized\(int\)](#) , MelonMod.InfoAttribute ,
MelonMod.GameAttributes ,
[MelonTypeBase<MelonMod>.ExecuteAll\(LemonAction<MelonMod>, bool, string\)](#) ,
MelonTypeBase<MelonMod>.RegisteredMelons , MelonTypeBase<MelonMod>.TypeName ,
MelonTypeBase<MelonMod>.MelonTypeName , MelonBase.OnMelonRegistered ,
MelonBase.OnMelonUnregistered , MelonBase.OnMelonInitializing , MelonBase.OnRegister ,
MelonBase.OnUnregister ,
[MelonBase.CreateWrapper<T>\(string, string, string, MelonGameAttribute\[\], MelonProcessAttribute\[\], int, Color?, Color?, string\)](#) ,
[MelonBase.RegisterSorted<T>\(IEnumerable<T>\)](#) , MelonBase.OnUpdate() ,
MelonBase.OnFixedUpdate() , MelonBase.OnLateUpdate() , MelonBase.OnGUI() ,
MelonBase.OnApplicationQuit() , MelonBase.OnPreferencesSaved() ,
[MelonBase.OnPreferencesSaved\(string\)](#) , MelonBase.OnPreferencesLoaded() ,
[MelonBase.OnPreferencesLoaded\(string\)](#) , MelonBase.OnEarlyInitializeMelon() ,
MelonBase.OnInitializeMelon() , MelonBase.OnLateInitializeMelon() , MelonBase.OnDeinitializeMelon() ,
[MelonBase.FindIncompatibilities\(MelonGameAttribute, string, string, string, string, MelonPlatformAttribute.CompatiblePlatforms, MelonPlatformDomainAttribute.CompatibleDomains\)](#) ,
[MelonBase.FindIncompatibilities\(MelonGameAttribute, string, string, SemVersion, string, MelonPlatformAttribute.CompatiblePlatforms, MelonPlatformDomainAttribute.CompatibleDomains\)](#) ,
MelonBase.FindIncompatibilitiesFromContext() ,
MelonBase.PrintIncompatibilities(MelonBase.Incompatibility[], MelonBase) , MelonBase.Register() ,
[MelonBase.FindMelon\(string, string\)](#) , [MelonBase.Unregister\(string, bool\)](#) ,
[MelonBase.ExecuteAll\(LemonAction<MelonBase>, bool, string\)](#) ,
[MelonBase.ExecuteList<T>\(LemonAction<T>, List<T>, bool, string\)](#) ,
[MelonBase.SendMessageAll\(string, params object\[\]\)](#) ,
[MelonBase.SendMessage\(string, params object\[\]\)](#) , MelonBase.OnApplicationLateStart()

MelonBase.OnApplicationStart() , MelonBase.OnModSettingsApplied() , MelonBase.MelonAssembly ,
MelonBase.Priority , MelonBase.ConsoleColor , MelonBase.AuthorConsoleColor , MelonBase.Info ,
MelonBase.AdditionalCredits , MelonBase.SupportedProcesses , MelonBase.Games ,
MelonBase.SupportedGameVersions , MelonBase.OptionalDependencies ,
MelonBase.SupportedPlatforms , MelonBase.SupportedDomain , MelonBase.SupportedMLVersion ,
MelonBase.SupportedMLBuild , MelonBase.HarmonyInstance , MelonBase.LoggerInstance ,
MelonBase.ID , MelonBase.Registered , MelonBase.harmonyInstance , MelonBase.Harmony ,
MelonBase.Assembly , MelonBase.HarmonyDontPatchAll , MelonBase.Hash , MelonBase.Location ,
[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[ReflectionUtils.GetValueTupleItems\(object\)](#) , [ReflectionUtils.IsValueTuple\(object\)](#)

Constructors

S1API()

```
public S1API()
```

Methods

OnPreSupportModule()

```
public override void OnPreSupportModule()
```

OnSceneWasInitialized(int, string)

```
public override void OnSceneWasInitialized(int buildIndex, string sceneName)
```

Parameters

buildIndex [int](#)

sceneName [string](#)

OnSceneWasLoaded(int, string)

```
public override void OnSceneWasLoaded(int buildIndex, string sceneName)
```

Parameters

buildIndex [int](#)

sceneName [string](#)

OnSceneWasUnloaded(int, string)

```
public override void OnSceneWasUnloaded(int buildIndex, string sceneName)
```

Parameters

buildIndex [int](#)

sceneName [string](#)

Class VersionChecker

Namespace: [S1API](#)

Assembly: S1API.dll

Version checking functionality for MelonLoader compatibility. Credits: estonia__ and k073l (S1 Modding Discord)

```
public static class VersionChecker
```

Inheritance

[object](#) ← VersionChecker

Inherited Members

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

Methods

CheckMelonLoaderVersion()

Checks the current MelonLoader version and warns the user if it's a known problematic version.

```
public static void CheckMelonLoaderVersion()
```