

# Pooler Developer manual

## 0.0.1

Generated on Sun Aug 10 2025 22:31:16 for Pooler Developer manual by Doxygen 1.14.0

Sun Aug 10 2025 22:31:16



# Chapter 1

## Namespace Index

### 1.1 Package List

Here are the packages with brief descriptions (if available):

<a href="#">Soor</a> . . . . .	??
<a href="#">Soor.Pooler</a> . . . . .	??



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

MonoBehaviour	
Soor.Pooler.Poolable	??
Soor.Pooler.PoolsManager	??
Soor.Pooler.Pooler	??



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Soor.Pooler.Poolable</a>	Manages a GameObject's lifecycle within an object pool . . . . .	??
<a href="#">Soor.Pooler.Pooler</a>	Manages the lifecycle of a <a href="#">Poolable</a> object pool, including creation, reuse, and cleanup . . . .	??
<a href="#">Soor.Pooler.PoolsManager</a>	. . . . .	??





## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/Poolable.cs	??
D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/Pooler.cs	??
D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/PoolManager.cs	??



## Chapter 5

# Namespace Documentation

### 5.1 Soor Namespace Reference

#### Namespaces

- namespace [Pooler](#)

### 5.2 Soor.Pooler Namespace Reference

#### Classes

- class [Poolable](#)  
*Manages a GameObject's lifecycle within an object pool.*
- class [Pooler](#)  
*Manages the lifecycle of a [Poolable](#) object pool, including creation, reuse, and cleanup.*
- class [PoolsManager](#)



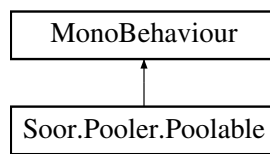
## Chapter 6

# Class Documentation

### 6.1 Soor.Pooler.Poolable Class Reference

Manages a GameObject's lifecycle within an object pool.

Inheritance diagram for Soor.Pooler.Poolable:



#### Public Member Functions

- void `OnCreate` ()  
*Called by the pool after instantiation.*
- void `OnGet` ()  
*Called by the pool when retrieved.*
- void `OnRelease` ()  
*Called by the pool when returned.*

#### Public Attributes

- UnityEvent `onCreateEvent`  
*Called when the object is first created by the pool.*
- UnityEvent `onEnableEvent`  
*Called when the object is enabled (OnEnable).*
- UnityEvent `onGetEvent`  
*Called when the object is taken from the pool.*
- UnityEvent `onReleaseEvent`  
*Called when the object is returned to the pool.*
- UnityEvent `onDisableEvent`  
*Called when the object is disabled (OnDisable).*
- UnityEvent `onDestroyEvent`  
*Called when the object is destroyed.*

## Properties

- bool `IsReleased` [get]

*A read-only property to access the `_isReleased` state from outside the class.*

### 6.1.1 Detailed Description

Manages a `GameObject`'s lifecycle within an object pool.

`UnityEvents` allow hooking custom logic to key lifecycle stages.

### 6.1.2 Member Function Documentation

#### 6.1.2.1 `OnCreate()`

```
void Soor.Pooler.Poolable.OnCreate ()
```

Called by the pool after instantiation.

#### 6.1.2.2 `OnGet()`

```
void Soor.Pooler.Poolable.OnGet ()
```

Called by the pool when retrieved.

#### 6.1.2.3 `OnRelease()`

```
void Soor.Pooler.Poolable.OnRelease ()
```

Called by the pool when returned.

### 6.1.3 Member Data Documentation

#### 6.1.3.1 `onCreateEvent`

```
UnityEvent Soor.Pooler.Poolable.onCreateEvent
```

Called when the object is first created by the pool.

#### 6.1.3.2 `onDestroyEvent`

```
UnityEvent Soor.Pooler.Poolable.onDestroyEvent
```

Called when the object is destroyed.

### 6.1.3.3 onDisableEvent

UnityEvent Soor.Pooler.Poolable.onDisableEvent

Called when the object is disabled (OnDisable).

### 6.1.3.4 onEnableEvent

UnityEvent Soor.Pooler.Poolable.onEnableEvent

Called when the object is enabled (OnEnable).

### 6.1.3.5 onGetEvent

UnityEvent Soor.Pooler.Poolable.onGetEvent

Called when the object is taken from the pool.

### 6.1.3.6 onReleaseEvent

UnityEvent Soor.Pooler.Poolable.onReleaseEvent

Called when the object is returned to the pool.

## 6.1.4 Property Documentation

### 6.1.4.1 IsReleased

bool Soor.Pooler.Poolable.IsReleased [get]

A read-only property to access the `_isReleased` state from outside the class.

True if the object is currently released (returned to the pool).

The documentation for this class was generated from the following file:

- D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/[Poolable.cs](#)

## 6.2 Soor.Pooler.Pooler Class Reference

Manages the lifecycle of a [Poolable](#) object pool, including creation, reuse, and cleanup.

## Public Member Functions

- **Pooler** (string poolName, List< **Poolable** > objectsToPool, int poolDefaultCapacity=10, int poolMaxCapacity=1000)  
*Creates a new **Pooler** instance with custom settings.*
- void **GenerateObjectPool** ()  
*Initializes the **ObjectPool** with core lifecycle callbacks and capacity settings.*
- void **DestroyObjectPool** ()  
*Disposes the **ObjectPool**, destroys its objects, and clears internal references.*

## Properties

- string **PoolName** [get]  
*Read-only property to access the pool's name.*
- **ObjectPool**< **Poolable** > **ObjectPool** [get]  
*Read-only property to access the **ObjectPool** itself.*

## 6.2.1 Detailed Description

Manages the lifecycle of a **Poolable** object pool, including creation, reuse, and cleanup.

## 6.2.2 Constructor & Destructor Documentation

### 6.2.2.1 Pooler()

```
Soor.Pooler.Pooler.Pooler (
    string poolName,
    List< Poolable > objectsToPool,
    int poolDefaultCapacity = 10,
    int poolMaxCapacity = 1000)
```

Creates a new **Pooler** instance with custom settings.

#### Parameters

<b>poolName</b>	Unique identifier for the pool.
<b>objectsToPool</b>	List of <b>Poolable</b> prefabs used for instantiation.
<b>poolDefaultCapacity</b>	Initial number of objects the pool can hold.
<b>poolMaxCapacity</b>	Maximum number of objects the pool can manage.

## 6.2.3 Member Function Documentation

### 6.2.3.1 DestroyObjectPool()

```
void Soor.Pooler.Pooler.DestroyObjectPool ()
```

Disposes the **ObjectPool**, destroys its objects, and clears internal references.



### 6.2.3.2 GenerateObjectPool()

```
void Soor.Pooler.Pooler.GenerateObjectPool ()
```

Initializes the ObjectPool with core lifecycle callbacks and capacity settings.

## 6.2.4 Property Documentation

### 6.2.4.1 ObjectPool

```
ObjectPool<Poolable> Soor.Pooler.Pooler.ObjectPool [get]
```

Read-only property to access the ObjectPool itself.

### 6.2.4.2 PoolName

```
string Soor.Pooler.Pooler.PoolName [get]
```

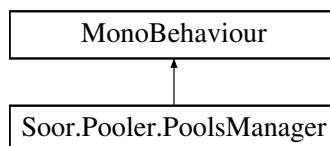
Read-only property to access the pool's name.

The documentation for this class was generated from the following file:

- D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/Pooler.cs

## 6.3 Soor.Pooler.PoolsManager Class Reference

Inheritance diagram for Soor.Pooler.PoolsManager:



### Public Member Functions

- void **AddPooler** (string poolName, List< **Poolable** > objectsToPool, int poolDefaultCapacity=10, int pool↔MaxCapacity=1000, bool generatePoolImmediately=true)  
*Adds a new **Pooler** instance to the list of managed pools.*
- void **GenerateObjectPool** (**Pooler** pooler)  
*Generates the ObjectPool<T> for the specified **Pooler** instance, if it has not been generated already.*
- void **GenerateObjectPool** (string name)  
*Finds and generates the ObjectPool<T> for the **Pooler** with the specified name.*
- **Pooler** **GetPooler** (string poolerName)  
*Retrieves a **Pooler** instance by its unique name.*
- void **DestroyObjectPool** (string name)  
*Finds and destroys the pool with the specified name, including all its GameObjects.*
- void **DestroyObjectPool** (**Pooler** pooler)  
*Destroys a pooler and its GameObjects.*

### 6.3.1 Member Function Documentation

#### 6.3.1.1 AddPooler()

```
void Soor.Pooler.PoolsManager.AddPooler (
    string poolName,
    List< Poolable > objectsToPool,
    int poolDefaultCapacity = 10,
    int poolMaxCapacity = 1000,
    bool generatePoolImmediately = true)
```

Adds a new [Pooler](#) instance to the list of managed pools.

##### Parameters

<a href="#">poolName</a>	The unique name for the new pool.
<a href="#">objectsToPool</a>	The list of poolable prefabs to be pooled.
<a href="#">poolDefaultCapacity</a>	The default initial capacity of the pool.
<a href="#">poolMaxCapacity</a>	The maximum number of objects the pool can hold.
<a href="#">generatePoolImmediately</a>	If true, the ObjectPool will be generated immediately after the <a href="#">Pooler</a> is added.

##### Exceptions

<a href="#">ArgumentException</a>	Thrown if a pool with the specified name already exists.
-----------------------------------	--

#### 6.3.1.2 DestroyObjectPool() [1/2]

```
void Soor.Pooler.PoolsManager.DestroyObjectPool (
    Pooler pooler)
```

Destroys a pooler and its GameObjects.

##### Parameters

<a href="#">pooler</a>	The pooler instance to destroy.
------------------------	---------------------------------

#### 6.3.1.3 DestroyObjectPool() [2/2]

```
void Soor.Pooler.PoolsManager.DestroyObjectPool (
    string name)
```

Finds and destroys the pool with the specified name, including all its GameObjects.

##### Parameters

<a href="#">name</a>	The unique name of the pool to destroy.
----------------------	---

#### 6.3.1.4 GenerateObjectPool() [1/2]

```
void Soor.Pooler.PoolsManager.GenerateObjectPool (
    Pooler pooler)
```

Generates the ObjectPool<T> for the specified [Pooler](#) instance, if it has not been generated already.

## Parameters

<code>pooler</code>	The <a href="#">Pooler</a> instance for which to generate the pool.
---------------------	---

**6.3.1.5 GenerateObjectPool()** [2/2]

```
void Soor.Pooler.PoolsManager.GenerateObjectPool (  
    string name)
```

Finds and generates the `ObjectPool<T>` for the [Pooler](#) with the specified name.

## Parameters

<code>name</code>	The unique name of the pool to generate.
-------------------	--

**6.3.1.6 GetPooler()**

```
Pooler Soor.Pooler.PoolsManager.GetPooler (  
    string poolerName)
```

Retrieves a [Pooler](#) instance by its unique name.

## Parameters

<code>poolerName</code>	The unique name of the pool to retrieve.
-------------------------	--

## Returns

The [Pooler](#) instance with the specified name.

## Exceptions

<code>KeyNotFoundException</code>	Thrown if a pool with the specified name does not exist.
-----------------------------------	--

The documentation for this class was generated from the following file:

- `D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/PoolsManager.cs`



## Chapter 7

# File Documentation

### 7.1 D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/Poolable.cs File Reference

#### Classes

- class [Soor.Pooler.Poolable](#)  
*Manages a GameObject's lifecycle within an object pool.*

#### Namespaces

- namespace [Soor](#)
- namespace [Soor.Pooler](#)

### 7.2 D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/Runtime/Pooler.cs File Reference

#### Classes

- class [Soor.Pooler.Pooler](#)  
*Manages the lifecycle of a [Poolable](#) object pool, including creation, reuse, and cleanup.*

#### Namespaces

- namespace [Soor](#)
- namespace [Soor.Pooler](#)

#### Typedefs

- using [Object](#) = UnityEngine.Object
- using [Random](#) = UnityEngine.Random

## 7.2.1 Typedef Documentation

### 7.2.1.1 Object

```
using Object = UnityEngine.Object
```

### 7.2.1.2 Random

```
using Random = UnityEngine.Random
```

## 7.3 D:/Projects/2021.3.16 projects/Pooler/Assets/ir.soor.pooler/Scripts/↵ Runtime/PoolManager.cs File Reference

### Classes

- class [Soor.Pooler.PoolManager](#)

### Namespaces

- namespace [Soor](#)
- namespace [Soor.Pooler](#)