

# Table of Contents

Uralstech.Utils.Singleton .....	2
DontCreateNewSingleton<T> .....	3
Singleton<T> .....	5

# Namespace Uralstech.Utils.Singleton

## Classes

### [DontCreateNewSingleton<T>](#)

Utility class to make inheriting types singletons.

### [Singleton<T>](#)

Utility class to make inheriting types singletons.

# Class DontCreateNewSingleton<T>

Namespace: [Uralstech.Utils.Singleton](#)

Utility class to make inheriting types singletons.

```
public class DontCreateNewSingleton<T> : MonoBehaviour where T : Component
```

## Type Parameters

**T**

The type to be made a singleton.

### Inheritance

System.Object ← DontCreateNewSingleton<T>

## Remarks

Unlike [Singleton<T>](#), this does not create a new instance of **T** if there are no existing ones.

## Properties

### HasInstance

Is there an existing instance of type **T**?

```
public static bool HasInstance { get; }
```

### Property Value

bool

## Instance

The active instance of type **T**.

```
public static T Instance { get; protected set; }
```

Property Value

T

## Methods

### Awake()

```
protected virtual void Awake()
```

Remarks

Checks if [Instance](#) is [null](#)[↗](#), and if yes, sets it to this instance. If no, destroys the current GameObject.

# Class Singleton<T>

Namespace: [Uralstech.Utils.Singleton](#)

Utility class to make inheriting types singletons.

```
public class Singleton<T> : MonoBehaviour where T : Component
```

## Type Parameters

**T**

The type to be made a singleton.

### Inheritance

System.Object ← Singleton<T>

## Properties

### HasInstance

Is there an existing instance of type **T**?

```
public static bool HasInstance { get; }
```

### Property Value

bool

### Remarks

If there is no set instance, this will try to find them. If more than one instance is found, the extra instances are deleted.

## Instance

The active instance of type **T**.

```
public static T Instance { get; }
```

## Property Value

T

## Remarks

If there is no set instance, this will try to find them. If more than one instance is found, the extra instances are deleted. If none are found, a new instance is created.