TinyEcs, simple pure .NET ECS
Welcome to TinyEcs

[This is preliminary documentation and is subject to change.]

TinyEcs, a small pure ecs written entirely in C#.

- Getting Started
  - First World

- See Also

  Other Resources
  Version History
TinyEcs, simple pure .NET ECS
Getting started by creating your first world.

**Creating a world.**

1. Create a message.

```csharp
// A message is a struct implementing the IMessage
// Systems respond to specific messages
struct UpdateMessage : IMessage
{
    private float deltaTime;
    public UpdateMessage(float deltaTime)
    {
        this.deltaTime = deltaTime;
    }
    public float DeltaTime => deltaTime;
}
```

2. Create a component.

```csharp
// A component is a mutable struct implementing IComponent
// Components are data containers for your entity
struct Position : IComponent
{
    public float X, Y;
}
```

3. Create a system.
4. Create and simulate the world.

```csharp
var world = World.Create();
var archetype = World.CreateArchetype(typeof(Position));
var entity = World.CreateEntity(archetype);
Console.Out.WriteLine($"x = {world.Ref<Position>(entity).X}");
world.Post(new UpdateMessage(0.16f));
Console.Out.WriteLine($"x = {world.Ref<Position>(entity).X}");
world.Post(new UpdateMessage(0.16f));
```
The positions of all entities will be updated by moving to the right.

See Also

Reference

- World
- IMessage
- IComponent
- GroupAttribute
- Entity
- ComponentSystemT
TinyEcs, simple pure .NET ECS
Working With Systems

[This is preliminary documentation and is subject to change.]

A brief introduction to working with `ComponentSystemT`

Creating a system that operates on multiple components

```csharp
public class MovementSystem : ComponentSystem<UpdateMessage>
{
    public class Data
    {
        public int Length;
        public RwDataStream<Position> Positions;
        public RoDataStream<Heading> Headings;
    }

    [Group] public Data data; // Will select all entities with Position and Heading component

    public void Execute(World world, UpdateMessage message)
    {
        for (var i = 0; i < data.Length; i++)
        {
            // The component streams are always stored in parallel
        }
    }
}
```

Getting a stream of entities

```csharp
public class HealthSystem : ComponentSystem<UpdateMessage>
{
}
```
public class Data
{
    public int Length;
    public RoDataStream<Entity> Entities; // Must
    public RoDataStream<Health> Healths;
}

public HealthSystem(SomeDependency dependency)
{
    this.dependency = dependency;
}

public void Execute(World world, UpdateMessage message)
{
    for (var i = 0; i < data.Length; i++)
    {
        if (data.Healths[i].Value <= 0)
        {
            dependency.DoSomethingWithEntity(data.Entities[i]);
        }
    }
}

public class RegenSystem : ComponentSystem<UpdateMessage>
{
    public class Data
    {
        public int Length;
        public RoDataStream<Health> Healths;
    }

    Including entities with a tag

    public class Data
    {
        public int Length;
        public RoDataStream<Health> Healths;
    }

    public HealthSystem(SomeDependency dependency)
    {
        this.dependency = dependency;
    }

    public void Execute(World world, UpdateMessage message)
    {
        for (var i = 0; i < data.Length; i++)
        {
            if (data.Healths[i].Value <= 0)
            {
                dependency.DoSomethingWithEntity(data.Entities[i]);
            }
        }
    }
}
Excluding entities marked by a tag

```csharp
public class MovementSystem : ComponentSystem<UpdateMessage>
{
  public class Data
  {
    public int Length;
    public RwDataStream<Position> Positions;
    public RoDataStream<Heading> Headings;
    [Exclude] public FrozenTag ExcludeFrozenTag;
    // As many tags as desired can be added
  }
  [Group] public Data data; // Will select all entities marked with a Position and Heading component and without a FrozenTag tag.

  public void Execute(World world, UpdateMessage message)
  {
    for (var i = 0; i < data.Length; i++)
    {
    }
  }
}
```
TinyEcs, simple pure .NET ECS
Version History

[This is preliminary documentation and is subject to change.]

The topics in this section describe the various changes made to the [TODO: Project Title] over the life of the project.

▶ Version History

Select a version below to see a description of its changes.

- Version 1.0.0.0
- [TODO: Add links to each specific version page]

▶ See Also

Other Resources
Welcome to TinyEcs
TinyEcs, simple pure .NET ECS
Version 1.0.0.0

[This is preliminary documentation and is subject to change.]

Version [TODO: Version] was released on [TODO: Date].

- Changes in This Release
  - [TODO: Add change items here]

- See Also
  - Other Resources
  - Version History
TinyEcs, simple pure .NET ECS
## TinyEcs Namespace

[This is preliminary documentation and is subject to change.]

A small pure entity-component-system written entirely in C#.

### Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ComponentSystemT</strong></td>
<td>Base class for systems.</td>
</tr>
<tr>
<td><strong>DoubleWriteLockException</strong></td>
<td>This exception is thrown when two different systems are attempting to write to the same entity.</td>
</tr>
<tr>
<td><strong>ExcludeAttribute</strong></td>
<td>Exclude entities that are marked by a tag.</td>
</tr>
<tr>
<td><strong>GroupAttribute</strong></td>
<td>Mark a field as an injected component group.</td>
</tr>
<tr>
<td><strong>World</strong></td>
<td>The World class glues entities, components and systems together. Use this class to create and destroy entities, get references to components and schedule systems. It</td>
</tr>
</tbody>
</table>
also allows very basic Dependency Injection for ComponentSystems and a simple resource locator.

## Structures

<table>
<thead>
<tr>
<th>Structure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archetype</td>
<td>An archetype is used to quickly instantiate entities with the same type of components.</td>
</tr>
<tr>
<td>Entity</td>
<td>An entity is a key that defines a relationship between different components.</td>
</tr>
<tr>
<td>RoDataStreamT</td>
<td>A read-only array of components or entities.</td>
</tr>
<tr>
<td>RwDataStreamT</td>
<td>A readable and writable array of components.</td>
</tr>
</tbody>
</table>

## Interfaces

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IComponent</td>
<td>Implement this interface to mark a struct as a component type. Allows the struct to be wrapped by a Ro/Rw-DataStream.</td>
</tr>
<tr>
<td>IData</td>
<td>Interface to mark a struct as usable for a RoDataStream.</td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IMessage</td>
<td>Implement this for messages.</td>
</tr>
<tr>
<td>IOnLoad</td>
<td>Allows a dependency to have a <code>OnLoad</code> method that gets called after a <code>World</code> has been constructed for further initialization dependant on a fully constructed <code>World</code>.</td>
</tr>
<tr>
<td>ITag</td>
<td>Implement this interface to mark a component as a tag. (Marker component)</td>
</tr>
</tbody>
</table>
TinyEcs, simple pure .NET ECS
Archetype Structure

[This is preliminary documentation and is subject to change.]

An archetype is used to quickly instantiate entities with the same type of components.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0  
The Archetype type exposes the following members.

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle</td>
<td>A handle representing this archetype.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance. (Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns the fully qualified type</td>
</tr>
</tbody>
</table>
name of this instance.
(Inherited from ValueType.)

See Also

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
Archetype Properties

[This is preliminary documentation and is subject to change.]

The Archetype type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle</td>
<td>A handle representing this archetype.</td>
</tr>
</tbody>
</table>

## See Also

Reference

Archetype Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ArchetypeHandle Property

[This is preliminary documentation and is subject to change.]

A handle representing this archetype.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
- Archetype Structure
- TinyEcs Namespace
Archetype Methods

[This is preliminary documentation and is subject to change.]

The Archetype type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns the fully qualified type name of this instance.</td>
</tr>
</tbody>
</table>

Top

### See Also

Reference
- Archetype Structure
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ComponentSystem<T> Class

[This is preliminary documentation and is subject to change.]

Base class for systems.

Inheritance Hierarchy

System
  Object
  TinyEcs
  ComponentSystemT

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

The ComponentSystemT type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComponentSystemT</td>
<td>Initializes a new instance of the ComponentSystemT class</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Execute</td>
<td>Executed when a message of type T is posted.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

### Remarks
When implementing a non-default constructor, the arguments will be injected. The dependencies need to have a default constructor in order for the dependency injection to work. Only one constructor is allowed for each ComponentSystemT. Arguments of the same type will share the same instance across ComponentSystems.

### See Also
Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ComponentSystem<T> Constructor

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the ComponentSystem<T> class

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
ComponentSystem<T> Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ComponentSystem$T Methods

[This is preliminary documentation and is subject to change.]

The ComponentSystem$T generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Execute</td>
<td>Executed when a message of type $T$ is posted.</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>
ToString

Returns a string that represents the current object. (Inherited from Object.)

See Also

Reference
ComponentSystemT Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ComponentSystemTExecute Method

[This is preliminary documentation and is subject to change.]

Executed when a message of type \( T \) is posted.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll)  
Version: 1.0.0

### Remarks


### See Also

Reference  
*ComponentSystemT Class*  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException Class

[This is preliminary documentation and is subject to change.]

This exception is thrown when two different systems are attempting to write to the same entity.

**Inheritance Hierarchy**

```plaintext
System
    Object
    System
        Exception
            TinyEcs
                DoubleWriteLockException
```

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

The DoubleWriteLockException type exposes the following members.

**Constructors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon" alt="DoubleWriteLockException" /></td>
<td>Initializes a new instance of the Exception class.</td>
</tr>
<tr>
<td><img src="icon" alt="DoubleWriteLockException(String)" /></td>
<td>Initializes a new instance of the Exception class with a specified</td>
</tr>
</tbody>
</table>
DoubleWriteLockException(SerializationInfo, StreamingContext)

Initializes a new instance of the Exception class with serialized data.

DoubleWriteLockException(String, Exception)

Initializes a new instance of the Exception class with a specified error message and a reference to the inner exception that is the cause of this exception.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value</td>
</tr>
</tbody>
</table>
pairs that provide additional user-defined information about the exception.  
(Inherited from Exception.)

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception.</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception.</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the Exception instance that caused the current exception.</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception.</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error.</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack.</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception.</td>
</tr>
</tbody>
</table>

(Inherited from Exception.)
## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the <code>Exception</code> that is the root cause of one or more subsequent exceptions. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetObjectData</td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>
**MemberwiseClone**  Creates a shallow copy of the current `Object`.
(Inherited from `Object`.)

**ToString**  Creates and returns a string representation of the current exception.
(Inherited from `Exception`.)

### Events

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SerializeObjectState</strong></td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception.</td>
</tr>
<tr>
<td>(Inherited from <code>Exception</code>)</td>
<td></td>
</tr>
</tbody>
</table>

### See Also

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException

Constructor

[This is preliminary documentation and is subject to change.]

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DoubleWriteLockException</td>
<td>Initializes a new instance of the Exception class.</td>
</tr>
<tr>
<td>DoubleWriteLockException(String)</td>
<td>Initializes a new instance of the Exception class with a specified error message.</td>
</tr>
<tr>
<td>DoubleWriteLockException(SerializationInfo,</td>
<td>Initializes a new instance of the Exception class with serialized data.</td>
</tr>
<tr>
<td>StreamingContext)</td>
<td></td>
</tr>
<tr>
<td><strong>DoubleWriteLockException(String, Exception)</strong></td>
<td>Initializes a new instance of the <code>Exception</code> class with a specified error message and a reference to the inner exception that is the cause of this exception.</td>
</tr>
</tbody>
</table>

**See Also**

Reference

`DoubleWriteLockException Class`

`TinyEcs Namespace`
TinyEcs, simple pure .NET ECS
DoubleWriteLockException
Constructor

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the Exception class.

Namespace:  TinyEcs
Assembly:  TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
DoubleWriteLockException Class
DoubleWriteLockException Overload
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException
Constructor (String)

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the Exception class with a specified error message.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
DoubleWriteLockException Class
DoubleWriteLockException Overload
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException
Constructor (SerializationInfo, StreamingContext)

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the Exception class with serialized data.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArgumentNullException</td>
<td>The info parameter is null.</td>
</tr>
<tr>
<td>SerializationException</td>
<td>The class name is null or HResult is zero (0).</td>
</tr>
</tbody>
</table>

See Also

Reference
DoubleWriteLockException Class
DoubleWriteLockException Overload
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException Constructor (String, Exception)

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the Exception class with a specified error message and a reference to the inner exception that is the cause of this exception.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
DoubleWriteLockException Class
DoubleWriteLockException Overload
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException

Properties

[This is preliminary documentation and is subject to change.]

The `DoubleWriteLockException` type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Gets a collection of key/value pairs that provide additional user-defined information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HelpLink</td>
<td>Gets or sets a link to the help file associated with this exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>HResult</td>
<td>Gets or sets HRESULT, a coded numerical value that is assigned to a specific exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>InnerException</td>
<td>Gets the <code>Exception</code> instance that caused the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Message</td>
<td>Gets a message that describes the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td>Source</td>
<td>Gets or sets the name of the application or the object that causes the error. (Inherited from Exception.)</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>StackTrace</td>
<td>Gets a string representation of the immediate frames on the call stack. (Inherited from Exception.)</td>
</tr>
<tr>
<td>TargetSite</td>
<td>Gets the method that throws the current exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

#### See Also

- Reference
  - DoubleWriteLockException Class
  - TinyEcs Namespace
TinyEcs, simple pure .NET ECS
DoubleWriteLockException

Methods

[This is preliminary documentation and is subject to change.]

The DoubleWriteLockException type exposes the following members.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>GetBaseException</td>
<td>When overridden in a derived class, returns the Exception that is the root cause of one or more subsequent exceptions.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Exception.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from Object.)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>GetObjectData</strong></td>
<td>When overridden in a derived class, sets the <code>SerializationInfo</code> with information about the exception. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the runtime type of the current instance. (Inherited from <code>Exception</code>.)</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Creates and returns a string representation of the current exception. (Inherited from <code>Exception</code>.)</td>
</tr>
</tbody>
</table>

**See Also**

- Reference
- `DoubleWriteLockException Class`
- `TinyEcs Namespace`
TinyEcs, simple pure .NET ECS
DoubleWriteLockException

Events

[This is preliminary documentation and is subject to change.]

The DoubleWriteLockException type exposes the following members.

- **Events**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="serialize-object-state" alt="Serializer" /></td>
<td>Occurs when an exception is serialized to create an exception state object that contains serialized data about the exception. (Inherited from Exception.)</td>
</tr>
</tbody>
</table>

See Also

Reference

- DoubleWriteLockException Class
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
Entity Structure

[This is preliminary documentation and is subject to change.]

An entity is a key that defines a relationship between different components.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0
The Entity type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Handle A handle that represents this entity.</td>
</tr>
</tbody>
</table>

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ ]</td>
<td>Equals Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ValueType.)</td>
</tr>
<tr>
<td>![ ]</td>
<td>GetHashCode Returns the hash code for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ValueType.)</td>
</tr>
<tr>
<td>![ ]</td>
<td>GetType Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>![ ]</td>
<td>ToString Returns the fully qualified type</td>
</tr>
</tbody>
</table>
See Also

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
Entity Properties

[This is preliminary documentation and is subject to change.]

The Entity type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Handle</td>
<td>A handle that represents this entity.</td>
</tr>
</tbody>
</table>

See Also

Reference
Entity Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
EntityHandle Property

[This is preliminary documentation and is subject to change.]

A handle that represents this entity.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
Entity Structure
TinyEcs Namespace
| TinyEcs, simple pure .NET ECS |  |
Entity Methods

[This is preliminary documentation and is subject to change.]

The **Entity** type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from <strong>Object</strong>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns the fully qualified type name of this instance. (Inherited from <strong>ValueType</strong>.)</td>
</tr>
</tbody>
</table>

Top

### See Also

- **Reference**
  - **Entity Structure**
  - **TinyEcs Namespace**
TinyEcs, simple pure .NET ECS
ExcludeAttribute Class

[This is preliminary documentation and is subject to change.]

Exclude entities that are marked by a tag.

## Inheritance Hierarchy

```
SystemObject  SystemAttribute
  TinyEcsExcludeAttribute
```

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0  
The *ExcludeAttribute* type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![ExcludeAttribute] ExcludeAttribute</td>
<td>Initializes a new instance of the <a href="#">ExcludeAttribute</a> class</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Typeld] Typeld</td>
<td>When implemented in a derived class, gets a unique identifier for this <a href="#">Attribute</a>.  (Inherited from <a href="#">Attribute</a>.)</td>
</tr>
</tbody>
</table>

### Methods
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsDefaultAttribute</td>
<td>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>Match</td>
<td>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**See Also**

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ExcludeAttribute Constructor

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the ExcludeAttribute class

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
ExcludeAttribute Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ExcludeAttribute Properties

[This is preliminary documentation and is subject to change.]
The ExcludeAttribute type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TyepId</td>
<td>When implemented in a derived class, gets a unique identifier for this Attribute. (Inherited from Attribute.)</td>
</tr>
</tbody>
</table>

See Also

Reference
ExcludeAttribute Class
TinyEcs Namespace
# ExcludeAttribute Methods

>This is preliminary documentation and is subject to change.

The `ExcludeAttribute` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>IsDefaultAttribute</td>
<td>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td>Match</td>
<td>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from <code>Attribute</code>.)</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from <strong>Object</strong>.)</td>
</tr>
</tbody>
</table>

**See Also**

Reference
- **ExcludeAttribute Class**
- **TinyEcs Namespace**
TinyEcs, simple pure .NET ECS
GroupAttribute Class

[This is preliminary documentation and is subject to change.]

Mark a field as an injected component group.

Inheritance Hierarchy

- System
  - Object
  - System
    - Attribute
  - TinyEcs
    - GroupAttribute

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

The `GroupAttribute` type exposes the following members.

Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GroupAttribute</td>
<td>Initializes a new instance of the <code>GroupAttribute</code> class</td>
</tr>
</tbody>
</table>

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typeld</td>
<td>When implemented in a derived class, gets a unique identifier for this <code>Attribute</code>. (Inherited from <code>Attribute</code>.)</td>
</tr>
</tbody>
</table>

Methods
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Returns a value that indicates whether this instance is equal to a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <strong>Type</strong> of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>IsDefaultAttribute</td>
<td>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>Match</td>
<td>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object. (Inherited from Attribute.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

See Also

Reference

TinyEcs Namespace
TinyEcs, simple pure .NET ECS
GroupAttribute Constructor

[This is preliminary documentation and is subject to change.]

Initializes a new instance of the GroupAttribute class

Namespace:  TinyEcs
Assembly:  TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
GroupAttribute Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
GroupAttribute Properties

[This is preliminary documentation and is subject to change.]

The GroupAttribute type exposes the following members.

## Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typeld</td>
<td>When implemented in a derived class, gets a unique identifier for this Attribute. (Inherited from Attribute.)</td>
</tr>
</tbody>
</table>

See Also

Reference
- GroupAttribute Class
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
The `GroupAttribute` type exposes the following members.

## Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Inherited From</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equals</strong></td>
<td>Returns a value that indicates whether this instance is equal to a specified object.</td>
<td><code>Attribute</code></td>
</tr>
<tr>
<td><strong>GetHashCode</strong></td>
<td>Returns the hash code for this instance.</td>
<td><code>Attribute</code></td>
</tr>
<tr>
<td><strong>GetType</strong></td>
<td>Gets the <code>Type</code> of the current instance.</td>
<td><code>Object</code></td>
</tr>
<tr>
<td><strong>IsDefaultAttribute</strong></td>
<td>When overridden in a derived class, indicates whether the value of this instance is the default value for the derived class.</td>
<td><code>Attribute</code></td>
</tr>
<tr>
<td><strong>Match</strong></td>
<td>When overridden in a derived class, returns a value that indicates whether this instance equals a specified object.</td>
<td><code>Attribute</code></td>
</tr>
</tbody>
</table>
ToString

Returns a string that represents the current object. (Inherited from Object.)

See Also

Reference
GroupAttribute Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
IComponent Interface

[This is preliminary documentation and is subject to change.]

Implement this interface to mark a struct as a component type. Allows the struct to be wrapped by a Ro/Rw-DataStream.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
IData Interface

[This is preliminary documentation and is subject to change.]

Interface to mark a struct as usable for a RoDataStream.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

▶ See Also

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
**IMessage Interface**

[This is preliminary documentation and is subject to change.]

Implement this for messages.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

▶ **See Also**

Reference  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
IOnLoad Interface

[This is preliminary documentation and is subject to change.]

Allows a dependency to have a `OnLoad` method that gets called after a `World` has been constructed for further initialization dependant on a fully constructed `World`.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0  
The IOnLoad type exposes the following members.

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>OnLoad</code></td>
<td>Gets called after a <code>World</code> has been constructed</td>
</tr>
</tbody>
</table>

### See Also

Reference TinyEcs Namespace
TinyEcs, simple pure .NET ECS
IOnLoad Methods

[This is preliminary documentation and is subject to change.]

The IOnLoad type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>⚜️ OnLoad</td>
<td>Gets called after a World has been constructed</td>
</tr>
</tbody>
</table>

Top

See Also

Reference
IOnLoad Interface
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
IOnLoad OnLoad Method

[This is preliminary documentation and is subject to change.]

Gets called after a World has been constructed

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
IOnLoad Interface
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
ITag Interface

[This is preliminary documentation and is subject to change.]

Implement this interface to mark a component as a tag. (Marker component)

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RoDataStream\(T\) Structure

[This is preliminary documentation and is subject to change.]

A read-only array of components or entities.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0  
The RoDataStream\(T\) type exposes the following members.

### Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoDataStream(T)</td>
<td>Wrap an array as a RoDataStream.</td>
</tr>
</tbody>
</table>

### Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Access element by index.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this</td>
</tr>
</tbody>
</table>
instance.  
(Inherited from ValueType.)

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns the fully qualified type name of this instance. (Inherited from ValueType.)</td>
</tr>
</tbody>
</table>

### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RoDataStreamT to T)</td>
<td>Cast to an array. O(1) operation. Length of the array may not match the length of the stream.</td>
</tr>
</tbody>
</table>

### See Also

- Reference
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RoDataStream\textit{T} Constructor

[This is preliminary documentation and is subject to change.]

Wrap an array as a RoDataStream.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference  
RoDataStream\textit{T} Structure  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RoDataStream\(T\) Properties

[This is preliminary documentation and is subject to change.]

The \texttt{RoDataStream} generic type exposes the following members.

\subsection*{Properties}

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Access element by index.</td>
</tr>
</tbody>
</table>

\section*{See Also}

Reference

\texttt{RoDataStream} Structure

TinyEcs Namespace
RoDataStreamTItem Property

[This is preliminary documentation and is subject to change.]

Access element by index.

Namespace:  TinyEcs
Assembly:  TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
RoDataStreamT Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RoDataStream$T$ Methods

[This is preliminary documentation and is subject to change.]

The RoDataStream$T$ generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance. (Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance. (Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns the fully qualified type name of this instance. (Inherited from ValueType.)</td>
</tr>
</tbody>
</table>

See Also

Reference
RoDataStream$T$ Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RoDataStream\(T\) Type
Conversions

[This is preliminary documentation and is subject to change.]

The RoDataStream\(T\) generic type exposes the following members.

Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RoDataStream(T) to (T))</td>
<td>Cast to an array. (O(1)) operation. Length of the array may not match the length of the stream.</td>
</tr>
</tbody>
</table>

See Also

Reference
RoDataStream\(T\) Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RoDataStream\textit{T} Conversion (RoDataStream\textit{T} to \textit{T})

[This is preliminary documentation and is subject to change.]

Cast to an array. \textit{O}(1) operation. Length of the array may not match the length of the stream.

\textbf{Namespace:} \textit{TinyEcs}
\textbf{Assembly:} \textit{TinyEcs} (in \textit{TinyEcs.dll}) Version: 1.0.0

\section*{See Also}
Reference
RoDataStream\textit{T} Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RwDataStreamT Structure

[This is preliminary documentation and is subject to change.]

A readable and writable array of components.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

The RwDataStreamT type exposes the following members.

▲ Constructors

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RwDataStreamT</td>
<td>Wrap an array as a RwDataStream.</td>
</tr>
</tbody>
</table>

▲ Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Access element by index.</td>
</tr>
</tbody>
</table>

▲ Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Indicates whether this instance and a specified object are equal. (Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this</td>
</tr>
</tbody>
</table>

Top
instance.
(Inherited from ValueType.)

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RwDataStreamT to T)</td>
<td>Cast to an array. O(1) operation. Length of the array may not match the length of the stream.</td>
</tr>
</tbody>
</table>

See Also

Reference
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RwDataStream\(T\) Constructor

[This is preliminary documentation and is subject to change.]

Wrap an array as a RwDataStream.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference  
RwDataStream\(T\) Structure  
TinyEcs Namespace
RwDataStreamT Properties

[This is preliminary documentation and is subject to change.]

The RwDataStreamT generic type exposes the following members.

Properties

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td>Access element by index.</td>
</tr>
</tbody>
</table>

See Also

Reference
RwDataStreamT Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RwDataStreamTItem Property

[This is preliminary documentation and is subject to change.]

Access element by index.

Namespace:  TinyEcs
Assembly:   TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
RwDataStreamT Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RwDataStreamT Methods

[This is preliminary documentation and is subject to change.]

The RwDataStreamT generic type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equals</td>
<td>Indicates whether this instance and a specified object are equal.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Returns the hash code for this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ValueType.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the Type of the current instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from Object.)</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns the fully qualified type name of this instance.</td>
</tr>
<tr>
<td></td>
<td>(Inherited from ValueType.)</td>
</tr>
</tbody>
</table>

Top

See Also

Reference
RwDataStreamT Structure
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RwDataStream\(T\) Type Conversions

[This is preliminary documentation and is subject to change.]

The RwDataStream\(T\) generic type exposes the following members.

### Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(RwDataStream(T) to (T))</td>
<td>Cast to an array. (O(1)) operation. Length of the array may not match the length of the stream.</td>
</tr>
</tbody>
</table>

See Also

Reference

RwDataStream\(T\) Structure

TinyEcs Namespace
TinyEcs, simple pure .NET ECS
RwDataStream\(T\) Conversion (RwDataStream\(T\) to \(T\))

[This is preliminary documentation and is subject to change.]

Cast to an array. \(O(1)\) operation. Length of the array may not match the length of the stream.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

- Reference
  - RwDataStream\(T\) Structure
  - TinyEcs Namespace
TinyEcs, simple pure .NET ECS
World Class

[This is preliminary documentation and is subject to change.]

The World class glues entities, components and systems together. Use this class to create and destroy entities, get references to components and schedule systems. It also allows very basic Dependency Injection for ComponentSystems and a simple resource locator.

Inheritance Hierarchy

```
SystemObject  TinyEcsWorld
```

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

The World type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="create_icon" alt="Create" /> Create</td>
<td>Create a new instance of a world. Will automatically create all ComponentSystems it can find and inject their dependencies.</td>
</tr>
<tr>
<td><img src="archetype_icon" alt="CreateArchetype" /> CreateArchetype(Type)</td>
<td>Create a new archetype.</td>
</tr>
<tr>
<td><img src="archetype_icon" alt="CreateArchetype" /> CreateArchetype(Archetype)</td>
<td>Create a new archetype by combining archetypes.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>CreateArchetype(Archetype, Type)</td>
<td>Derive a new archetype from an existing archetype.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype, Type)</td>
<td>Derive a new archetype from multiple archetypes.</td>
</tr>
<tr>
<td>CreateEntity</td>
<td>Create a new entity.</td>
</tr>
<tr>
<td>DestroyEntity</td>
<td>Destroy an existing entity.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object. (Inherited from Object.)</td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup operations before it is reclaimed by garbage collection. (Inherited from Object.)</td>
</tr>
<tr>
<td>GetArchetype</td>
<td>Get the archetype of an entity.</td>
</tr>
<tr>
<td>GetDependencyT</td>
<td>Gets a resource that has been injected as a dependency in a</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <code>Type</code> of the current instance. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>MemberwiseClone</td>
<td>Creates a shallow copy of the current <code>Object</code>. (Inherited from <code>Object</code>.)</td>
</tr>
<tr>
<td>Post</td>
<td>Post a message and let all eligible systems handle it.</td>
</tr>
<tr>
<td>RefT</td>
<td>Get a reference to a component belonging to an entity.</td>
</tr>
<tr>
<td>ToString</td>
<td>Returns a string that represents the current object. (Inherited from <code>Object</code>.)</td>
</tr>
</tbody>
</table>
Reference

TinyEcs Namespace
TinyEcs, simple pure .NET ECS
World Methods

[This is preliminary documentation and is subject to change.]

The World type exposes the following members.

Methods

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>Create a new instance of a world. Will automatically create all ComponentSystems it can find and inject their dependencies.</td>
</tr>
<tr>
<td>CreateArchetype(Type)</td>
<td>Create a new archetype.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype)</td>
<td>Create a new archetype by combining archetypes.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype, Type)</td>
<td>Derive a new archetype from an existing archetype.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype, Type)</td>
<td>Derive a new archetype from multiple archetypes.</td>
</tr>
<tr>
<td>CreateEntity</td>
<td>Create a new entity.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DestroyEntity</td>
<td>Destroy an existing entity.</td>
</tr>
<tr>
<td>Equals</td>
<td>Determines whether the specified object is equal to the current object.</td>
</tr>
<tr>
<td></td>
<td><em>(Inherited from Object.)</em></td>
</tr>
<tr>
<td>Finalize</td>
<td>Allows an object to try to free resources and perform other cleanup</td>
</tr>
<tr>
<td></td>
<td>operations before it is reclaimed by garbage collection.</td>
</tr>
<tr>
<td></td>
<td><em>(Inherited from Object.)</em></td>
</tr>
<tr>
<td>GetArchetype</td>
<td>Get the archetype of an entity.</td>
</tr>
<tr>
<td>GetDependencyT</td>
<td>Gets a resource that has been injected as a dependency in a ComponentSystem.</td>
</tr>
<tr>
<td>GetHashCode</td>
<td>Serves as the default hash function. <em>(Inherited from Object.)</em></td>
</tr>
<tr>
<td>GetType</td>
<td>Gets the <em>Type</em> of the current instance. <em>(Inherited from</em> Object.)*</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>MemberwiseClone</strong></td>
<td>Creates a shallow copy of the current Object. (Inherited from Object.)</td>
</tr>
<tr>
<td><strong>Post</strong></td>
<td>Post a message and let all eligible systems handle it.</td>
</tr>
<tr>
<td><strong>RefT</strong></td>
<td>Get a reference to a component belonging to an entity.</td>
</tr>
<tr>
<td><strong>ToString</strong></td>
<td>Returns a string that represents the current object. (Inherited from Object.)</td>
</tr>
</tbody>
</table>

**See Also**

- Reference
- World Class
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldCreate Method

[This is preliminary documentation and is subject to change.]

Create a new instance of a world. Will automatically create all ComponentSystems it can find and inject their dependencies.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
World Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldCreateArchetype Method

[This is preliminary documentation and is subject to change.]

Overload List

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateArchetype(Type)</td>
<td>Create a new archetype.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype)</td>
<td>Create a new archetype by combining archetypes.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype, Type)</td>
<td>Derive a new archetype from an existing archetype.</td>
</tr>
<tr>
<td>CreateArchetype(Archetype, Type)</td>
<td>Derive a new archetype from multiple archetypes.</td>
</tr>
</tbody>
</table>

See Also

Reference
World Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldCreateArchetype Method (Type)

[This is preliminary documentation and is subject to change.]

Create a new archetype.

**Namespace:**  TinyEcs
**Assembly:**  TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
  World Class
  CreateArchetype Overload
  TinyEcs Namespace
TinyEcs, simple pure .NET ECS
World

CreateArchetype Method
(Archetype)

Create a new archetype by combining archetypes.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
World Class
CreateArchetype Overload
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldCreateArchetype Method (Archetype, Type)

[This is preliminary documentation and is subject to change.]

Derive a new archetype from an existing archetype.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

- Reference
- World Class
- CreateArchetype Overload
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldCreateArchetype Method (Archetype, Type)

[This is preliminary documentation and is subject to change.]

Derive a new archetype from multiple archetypes.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

**See Also**

Reference  
World Class  
CreateArchetype Overload  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldCreateEntity Method

[This is preliminary documentation and is subject to change.]

Create a new entity.

**Namespace:** TinyEcs
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference
World Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldDestroyEntity Method

[This is preliminary documentation and is subject to change.]

Destroy an existing entity.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference  
World Class  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldGetArchetype Method

[This is preliminary documentation and is subject to change.]

Get the archetype of an entity.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference  
World Class  
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldGetDependencyT Method

(This is preliminary documentation and is subject to change.)

Gets a resource that has been injected as a dependency in a ComponentSystem.

Namespace: TinyEcs
Assembly: TinyEcs (in TinyEcs.dll) Version: 1.0.0

Exceptions

<table>
<thead>
<tr>
<th>Exception</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyNotFoundException</td>
<td>Thrown when the requested resource was not injected.</td>
</tr>
</tbody>
</table>

See Also

Reference
World Class
TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldPost Method

[This is preliminary documentation and is subject to change.]

Post a message and let all eligible systems handle it.

**Namespace:** TinyEcs  
**Assembly:** TinyEcs (in TinyEcs.dll) Version: 1.0.0

⚠️ Remarks

Systems will run in parallel, directly using this class in the Execute method of a system is unsafe.

⚠️ See Also

Reference
- World Class
- TinyEcs Namespace
TinyEcs, simple pure .NET ECS
WorldRefT Method

[This is preliminary documentation and is subject to change.]

Get a reference to a component belonging to an entity.

**Namespace**: TinyEcs  
**Assembly**: TinyEcs (in TinyEcs.dll) Version: 1.0.0

See Also

Reference  
World Class  
TinyEcs Namespace