holoscan.resources

This module provides a Python API to underlying C++ API Resources.

holoscan.resources.Allocator	Base allocator class.
holoscan.resources.BlockMemoryPool	Block memory pool resource.
holoscan.resources.Clock	Base clock class.
holoscan.resources.CudaStreamPool	CUDA stream pool.
holoscan.resources.DoubleBufferReceiver	Receiver using a double-buffered queue.

holoscan.resources.DoubleBufferTransmitter	Transmitter using a double-buffered queue.
holoscan.resources.GXFComponentResource(...)	Class that wraps a GXF Component as a Holoscan Resource.
holoscan.resources.ManualClock	Manual clock.
holoscan.resources.MemoryStorageType	Members:

holoscan.resources.RealtimeClock	Realtime clock.
holoscan.resources.Receiver	Base GXF receiver class.
holoscan.resources.SerializationBuffer	Serialization Buffer.
holoscan.resources.StdComponentSerializer	Serializer for GXF Timestamp and Tensor components.
holoscan.resources.StdEntitySerializer	Default serializer for GXF entities.

holoscan.resources.Transmitter	Base GXF transmitter class.
holoscan.resources.UnboundedAllocator	Unbounded allocator.
holoscan.resources.UcxComponentSerializer	UCX component serializer.
holoscan.resources.UcxEntitySerializer	UCX entity serializer.

holoscan.resources.UcxHoloscanComponentSerializer	UCX Holoscan component serializer.
holoscan.resources.UcxReceiver	UCX network receiver using a double-buffered queue.
holoscan.resources.UcxSerializationBuffer	UCX serialization buffer.
holoscan.resources.UcxTransmitter	UCX network transmitter using a double-buffered queue.

class holoscan.resources.Allocator

Bases: holoscan.gxf._gxf.GXFResource

Base allocator class.

Attributes

args	The list of arguments associated with the component.
block_size	Get the block size of the allocator.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg(*args, **kwargs)	Overloaded function.
allocate(self, size, type)	Allocate the requested amount of memory.
free(self, pointer)	Free the allocated memory
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
is_available(self, size)	Boolean representing whether the resource is available.
setup(self, arg0)	setup method for the resource.

__init__(*self: holoscan.resources._resources.Allocator*) None

Base allocator class.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

allocate(*self: holoscan.resources._resources.Allocator*, *size: int*, *type: holoscan.resources._resources.MemoryStorageType*) int

Allocate the requested amount of memory.

Parameters

size

The amount of memory to allocate

type

Enum representing the type of memory to allocate.

Returns

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property args

The list of arguments associated with the component.

Returns

arglist

property block_size

Get the block size of the allocator.

Returns

int

The block size of the allocator. Returns 1 for byte-based allocators.

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

free(*self: holoscan.resources._resources.Allocator*, *pointer: int*) None

Free the allocated memory

Parameters

pointer

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

is_available(*self: holoscan.resources._resources.Allocator*, *size: int*) bool

Boolean representing whether the resource is available.

Returns

bool

Availability of the resource.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.core._core.Resource*, *arg0: holoscan.core._core.ComponentSpec*) None

setup method for the resource.

property spec

class holoscan.resources.BlockMemoryPool

Bases: holoscan.resources._resources.Allocator

Block memory pool resource.

Provides a maximum number of equally sized blocks of memory.

Parameters

fragment

The fragment to assign the resource to.

storage_type

The storage type (0=Host, 1=Device, 2=System).

block_size

The size of each block in the memory pool (in bytes).

num_blocks

The number of blocks in the memory pool.

dev_id

CUDA device ID. Specifies the device on which to create the memory pool.

name

The name of the memory pool.

Attributes

args	The list of arguments associated with the component.
block_size	Get the block size of the allocator.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

`add_arg`(*args, **kwargs)	Overloaded function.
`allocate`(self, size, type)	Allocate the requested amount of memory.
`free`(self, pointer)	Free the allocated memory
`gxf_initialize`(self)	Initialize the component.
`initialize`(self)	Initialize the component.
`is_availabl e`(self, size)	Boolean representing whether the resource is available.
`setup`(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.BlockMemoryPool*, *fragment: holoscan.core._core.Fragment*, *storage_type: int*, *block_size: int*, *num_blocks: int*, *dev_id:*

int = 0, *name: str = 'block_memory_pool'*) None

Block memory pool resource.

Provides a maximum number of equally sized blocks of memory.

Parameters

fragment

The fragment to assign the resource to.

storage_type

The storage type (0=Host, 1=Device, 2=System).

block_size

The size of each block in the memory pool (in bytes).

num_blocks

The number of blocks in the memory pool.

dev_id

CUDA device ID. Specifies the device on which to create the memory pool.

name

The name of the memory pool.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

allocate(*self: holoscan.resources._resources.Allocator*, *size: int*, *type: holoscan.resources._resources.MemoryStorageType*) int

Allocate the requested amount of memory.

Parameters

size

The amount of memory to allocate

type

Enum representing the type of memory to allocate.

Returns

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property args

The list of arguments associated with the component.

Returns

arglist

property block_size

Get the block size of the allocator.

Returns

int

The block size of the allocator. Returns 1 for byte-based allocators.

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

free(*self: holoscan.resources._resources.Allocator*, *pointer: int*) None

Free the allocated memory

Parameters

pointer

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

is_available(*self: holoscan.resources._resources.Allocator*, *size: int*) bool

Boolean representing whether the resource is available.

Returns

bool

Availability of the resource.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.BlockMemoryPool*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.Clock

Bases: holoscan.gxf._gxf.GXFResource

Base clock class.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.

id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args, **kwargs)	Overloaded function.
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the component.
setup (self, arg0)	setup method for the resource.

__init__(**args*, ***kwargs*)

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.core._core.Resource*, *arg0: holoscan.core._core.ComponentSpec*) None

setup method for the resource.

property spec

class holoscan.resources.CudaStreamPool

Bases: `holoscan.resources._resources.Allocator`

CUDA stream pool.

Parameters

fragment

The fragment to assign the resource to.

dev_id

CUDA device ID. Specifies the device on which to create the stream pool.

stream_flags

Flags for CUDA streams in the pool. This will be passed to CUDA's cudaStreamCreateWithPriority [R47f67d935b70-1] when creating the streams. The default value of 0 corresponds to `cudaStreamDefault`. A value of 1 corresponds to `cudaStreamNonBlocking`, indicating that the stream can run concurrently with work in stream 0 (default stream) and should not perform any implicit synchronization with it.

stream_priority

Priority value for CUDA streams in the pool. This is an integer value passed to cudaSreamCreateWithPriority [R47f67d935b70-1]. Lower numbers represent higher priorities.

reserved_size

The number of CUDA streams to initially reserve in the pool (prior to first request).

max_size

The maximum number of streams that can be allocated, unlimited by default.

name

The name of the stream pool.

References

[1]

https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__STREAM.html

Attributes

`args`	The list of arguments associated with the component.
`block_size`	Get the block size of the allocator.
`description`	YAML formatted string describing the resource.
`fragment`	Fragment that the resource belongs to.
`gxf_cid`	The GXF component ID.
`gxf_cname`	The name of the component.
`gxf_context`	The GXF context of the component.
`gxf_eid`	The GXF entity ID.
`gxf_typename`	The GXF type name of the resource.
`id`	The identifier of the component.
`name`	The name of the resource.

spec	

Methods

`add_arg`(*args,	Overloaded function.

**kwargs)	
allocate(self, size, type)	Allocate the requested amount of memory.
free(self, pointer)	Free the allocated memory
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
is_availalble(self, size)	Boolean representing whether the resource is available.
setup(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.CudaStreamPool*, *fragment: holoscan.core._core.Fragment*, *dev_id: int = 0*, *stream_flags: int = 0*, *stream_priority: int = 0*, *reserved_size: int = 1*, *max_size: int = 0*, *name: str = 'cuda_stream_pool'*) None

CUDA stream pool.

Parameters

fragment

The fragment to assign the resource to.

dev_id

CUDA device ID. Specifies the device on which to create the stream pool.

stream_flags

Flags for CUDA streams in the pool. This will be passed to CUDA's cudaStreamCreateWithPriority [Rb9bddbe55e1a-1] when creating the streams. The default value of 0 corresponds to `cudaStreamDefault`. A value of 1 corresponds to `cudaStreamNonBlocking`, indicating that the stream can run concurrently with work in stream 0 (default stream) and should not perform any implicit synchronization with it.

stream_priority

Priority value for CUDA streams in the pool. This is an integer value passed to cudaSreamCreateWithPriority [Rb9bddbe55e1a-1]. Lower numbers represent higher priorities.

reserved_size

The number of CUDA streams to initially reserve in the pool (prior to first request).

max_size

The maximum number of streams that can be allocated, unlimited by default.

name

The name of the stream pool.

References

[1]

https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__STREAM.html

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

allocate(*self: holoscan.resources._resources.Allocator*, *size: int*, *type: holoscan.resources._resources.MemoryStorageType*) int

Allocate the requested amount of memory.

Parameters

size

The amount of memory to allocate

type

Enum representing the type of memory to allocate.

Returns

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property args

The list of arguments associated with the component.

Returns

arglist

property block_size

Get the block size of the allocator.

Returns

int

The block size of the allocator. Returns 1 for byte-based allocators.

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

free(*self: holoscan.resources._resources.Allocator*, *pointer: int*) None

Free the allocated memory

Parameters

pointer

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

is_available(*self: holoscan.resources._resources.Allocator*, *size: int*) bool

Boolean representing whether the resource is available.

Returns

bool

Availability of the resource.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.CudaStreamPool*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.DoubleBufferReceiver

Bases: holoscan.resources._resources.Receiver

Receiver using a double-buffered queue.

New messages are first pushed to a back stage.

Parameters

fragment

The fragment to assign the resource to.

capacity

The capacity of the receiver.

policy

The policy to use (0=pop, 1=reject, 2=fault).

name

The name of the receiver.

Attributes

`args`	The list of arguments associated with the component.
`description`	YAML formatted string describing the resource.
`fragment`	Fragment that the resource belongs to.
`gxf_cid`	The GXF component ID.
`gxf_cname`	The name of the component.
`gxf_context`	The GXF context of the component.
`gxf_eid`	The GXF entity ID.
`gxf_typename`	The GXF type name of the resource.
`id`	The identifier of the component.
`name`	The name of the resource.

spec	

Methods

add_arg(*args, **kwargs)	Overloaded function.
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
setup(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.DoubleBufferReceiver*, *fragment: holoscan.core._core.Fragment*, *capacity: int = 1*, *policy: int = 2*, *name: str = 'double_buffer_receiver'*) None

Receiver using a double-buffered queue.

New messages are first pushed to a back stage.

Parameters

fragment

The fragment to assign the resource to.

capacity

The capacity of the receiver.

policy

The policy to use (0=pop, 1=reject, 2=fault).

name

The name of the receiver.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.DoubleBufferReceiver*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.DoubleBufferTransmitter

Bases: holoscan.resources._resources.Transmitter

Transmitter using a double-buffered queue.

Messages are pushed to a back stage after they are published.

Parameters

fragment

The fragment to assign the resource to.

capacity

The capacity of the transmitter.

policy

The policy to use (0=pop, 1=reject, 2=fault).

name

The name of the transmitter.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

`add_arg`(*args, **kwargs)	Overloaded function.
`gxf_initialize`(self)	Initialize the component.
`initialize`(self)	Initialize the component.
`setup`(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.DoubleBufferTransmitter*, *fragment: holoscan.core._core.Fragment*, *capacity: int = 1*, *policy: int = 2*, *name: str = 'double_buffer_transmitter'*) None

Transmitter using a double-buffered queue.

Messages are pushed to a back stage after they are published.

Parameters

fragment

The fragment to assign the resource to.

capacity

The capacity of the transmitter.

policy

The policy to use (0=pop, 1=reject, 2=fault).

name

The name of the transmitter.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.DoubleBufferTransmitter*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.GXFComponentResource(*fragment*, **args*, ***kwargs*)

Bases: `holoscan.resources._resources.GXFComponentResource`

Class that wraps a GXF Component as a Holoscan Resource.

Parameters

fragment

The fragment that the resource belongs to.

gxf_typename

The GXF type name that identifies the specific GXF Component being wrapped.

name

The name of the resource. Default value is `"gxf_component"`.

****kwargs**

The additional keyword arguments that can be passed depend on the underlying GXF Component. These parameters can provide further

customization and functionality to the resource.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

`add_arg`(*args, **kwargs)	Overloaded function.
`gxf_initialize`(self)	Initialize the component.
`initialize`(self)	Initialize the resource.
`setup`(self, spec)	Define the resource specification.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.resources._resources.GXFComponentResource*) None

Initialize the resource.

This method is called only once when the resource is created for the first time, and uses a light-weight initialization.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.GXFComponentResource*, *spec: holoscan.core._core.ComponentSpec*) None

Define the resource specification.

Parameters

spec

The resource specification.

property spec

class holoscan.resources.ManualClock

Bases: `holoscan.resources._resources.Clock`

Manual clock.

Parameters

fragment

The fragment to assign the resource to.

initial_timestamp

The initial timestamp on the clock (in nanoseconds).

name

The name of the clock.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.

gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args, **kwargs)	Overloaded function.
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the component.
setup (self, spec)	Define the component specification.
sleep_for (self, arg0)	Set the GXF scheduler to sleep for a specified duration.
sleep_until (self, target_t	Set the GXF scheduler to sleep until a specified timestamp.

ime_ns)	
time (self)	The current time of the clock (in seconds).
timestamp (self)	The current timestamp of the clock (in nanoseconds).

__init__(*self: holoscan.resources._resources.ManualClock*, *fragment: holoscan.core._core.Fragment*, *initial_timestamp: int = 0*, *name: str = 'realtime_clock'*) None

Manual clock.

Parameters

fragment

The fragment to assign the resource to.

initial_timestamp

The initial timestamp on the clock (in nanoseconds).

name

The name of the clock.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.ManualClock*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

sleep_for(*self: holoscan.resources._resources.ManualClock*, *arg0: object*) None

Set the GXF scheduler to sleep for a specified duration.

Parameters

duration_ns

The duration to sleep (in nanoseconds).

sleep_until(*self: holoscan.resources._resources.ManualClock*, *target_time_ns: int*) None

Set the GXF scheduler to sleep until a specified timestamp.

Parameters

target_time_ns

The target timestamp (in nanoseconds).

property spec

time(*self: holoscan.resources._resources.ManualClock*) float

The current time of the clock (in seconds).

Parameters

time

The current time of the clock (in seconds).

timestamp(*self: holoscan.resources._resources.ManualClock*) int

The current timestamp of the clock (in nanoseconds).

Parameters

timestamp

The current timestamp of the clock (in nanoseconds).

class holoscan.resources.MemoryStorageType

Bases: pybind11_builtins.pybind11_object

Members:

HOST

DEVICE

SYSTEM

Attributes

name	

value	

DEVICE = *<MemoryStorageType.DEVICE: 1>*

HOST = *<MemoryStorageType.HOST: 0>*

SYSTEM = *<MemoryStorageType.SYSTEM: 2>*

__init__(*self: holoscan.resources._resources.MemoryStorageType*, *value: int*) None

property name

property value

class holoscan.resources.RealtimeClock

Bases: holoscan.resources._resources.Clock

Realtime clock.

Parameters

fragment

The fragment to assign the resource to.

initial_timestamp

The initial time offset used until time scale is changed manually.

initial_time_scale

The initial time scale used until time scale is changed manually.

use_time_since_epoch

If `True`, clock time is time since epoch + *initial_time_offset* at `initialize()`. Otherwise clock time is *initial_time_offset* at `initialize()`.

name

The name of the clock.

Attributes

`args`	The list of arguments associated with the component.
`description`	YAML formatted string describing the resource.
`fragment`	Fragment that the resource belongs to.
`gxf_cid`	The GXF component ID.
`gxf_cname`	The name of the component.
`gxf_context`	The GXF context of the component.
`gxf_eid`	The GXF entity ID.
`gxf_typename`	The GXF type name of the resource.
`id`	The identifier of the component.

name	The name of the resource.

spec	

Methods

add_arg (*args, **kwargs)	Overloaded function.
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the component.
set_time_scale (self, time_scale)	Adjust the time scaling used by the clock.
setup (self, spec)	Define the component specification.
sleep_for (self, arg0)	Set the GXF scheduler to sleep for a specified duration.

`sleep_until`(self, target_time_ns)	Set the GXF scheduler to sleep until a specified timestamp.
`time`(self)	The current time of the clock (in seconds).
`timestamp`(self)	The current timestamp of the clock (in nanoseconds).

__init__(*self: holoscan.resources._resources.RealtimeClock*, *fragment: holoscan.core._core.Fragment*, *initial_time_offset: float = 0.0*, *initial_time_scale: float = 1.0*, *use_time_since_epoch: bool = False*, *name: str = 'realtime_clock'*) None

Realtime clock.

Parameters

fragment

The fragment to assign the resource to.

initial_timestamp

The initial time offset used until time scale is changed manually.

initial_time_scale

The initial time scale used until time scale is changed manually.

use_time_since_epoch

If `True`, clock time is time since epoch + *initial_time_offset* at `initialize()`. Otherwise clock time is *initial_time_offset* at `initialize()`.

name

The name of the clock.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

set_time_scale(*self: holoscan.resources._resources.RealtimeClock*, *time_scale: float*) None

Adjust the time scaling used by the clock.

Parameters

time_scale

Durations (e.g. for periodic condition or sleep_for) are reduced by this scale value. A scale of 1.0 represents real-time while a scale of 2.0 would represent a clock where time elapses twice as fast.

setup(*self: holoscan.resources._resources.RealtimeClock*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

sleep_for(*self: holoscan.resources._resources.RealtimeClock*, *arg0: object*) None

Set the GXF scheduler to sleep for a specified duration.

Parameters

duration_ns

The duration to sleep (in nanoseconds).

sleep_until(*self: holoscan.resources._resources.RealtimeClock*, *target_time_ns: int*) None

Set the GXF scheduler to sleep until a specified timestamp.

Parameters

target_time_ns

The target timestamp (in nanoseconds).

property spec

time(*self: holoscan.resources._resources.RealtimeClock*) float

The current time of the clock (in seconds).

Parameters

time

The current time of the clock (in seconds).

timestamp(*self: holoscan.resources._resources.RealtimeClock*) int

The current timestamp of the clock (in nanoseconds).

Parameters

timestamp

The current timestamp of the clock (in nanoseconds).

class holoscan.resources.Receiver

Bases: holoscan.gxf._gxf.GXFResource

Base GXF receiver class.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.

gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args, **kwargs)	Overloaded function.
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the component.

setup(self, arg0)	setup method for the resource.

__init__(*self: holoscan.resources._resources.Receiver*) None

Base GXF receiver class.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.core._core.Resource*, *arg0: holoscan.core._core.ComponentSpec*) None

setup method for the resource.

property spec

class holoscan.resources.SerializationBuffer

Bases: holoscan.gxf._gxf.GXFResource

Serialization Buffer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

buffer_size

The size of the buffer in bytes.

name

The name of the serialization buffer

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args,	Overloaded function.

**kwargs)	
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the component.
setup (self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.SerializationBuffer*, *fragment: holoscan.core._core.Fragment*, *allocator: holoscan.resources._resources.Allocator = None*, *buffer_size: int = 4096*, *name: str = 'serialization_buffer'*) None

Serialization Buffer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

buffer_size

The size of the buffer in bytes.

name

The name of the serialization buffer

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.SerializationBuffer*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.StdComponentSerializer

Bases: holoscan.gxf._gxf.GXFResource

Serializer for GXF Timestamp and Tensor components.

Parameters

fragment

The fragment to assign the resource to.

name

The name of the serializer.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.

`gxf_cname`	The name of the component.
`gxf_context`	The GXF context of the component.
`gxf_eid`	The GXF entity ID.
`gxf_typename`	The GXF type name of the resource.
`id`	The identifier of the component.
`name`	The name of the resource.

spec	

Methods

`add_arg` (*args, **kwargs)	Overloaded function.
`gxf_initialize` (self)	Initialize the component.
`initialize` (self)	Initialize the resource
`setup` (self, s	Define the component specification.

pec)	

__init__(*self: holoscan.resources._resources.StdComponentSerializer*, *fragment: holoscan.core._core.Fragment*, *name: str = 'standard_component_serializer'*) None

Serializer for GXF Timestamp and Tensor components.

Parameters

fragment

The fragment to assign the resource to.

name

The name of the serializer.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.resources._resources.StdComponentSerializer*) None

Initialize the resource

This method is called only once when the resource is created for the first time, and uses a light-weight initialization.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.StdComponentSerializer*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.StdEntitySerializer

Bases: `holoscan.gxf._gxf.GXFResource`

Default serializer for GXF entities.

Parameters

fragment

The fragment to assign the resource to.

name

The name of the serializer.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

`add_arg`(*args, **kwargs)	Overloaded function.
`gxf_initialize`(self)	Initialize the component.
`initialize`(self)	Initialize the resource
`setup`(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.StdEntitySerializer*, *fragment: holoscan.core._core.Fragment*, *name: str = 'std_entity_serializer'*) None

Default serializer for GXF entities.

Parameters

fragment

The fragment to assign the resource to.

name

The name of the serializer.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.resources._resources.StdEntitySerializer*) None

Initialize the resource

This method is called only once when the resource is created for the first time, and uses a light-weight initialization.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.StdEntitySerializer*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.Transmitter

Bases: holoscan.gxf._gxf.GXFResource

Base GXF transmitter class.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.

gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg(*args, **kwargs)	Overloaded function.
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
setup(self, arg0)	setup method for the resource.

__init__(*self: holoscan.resources._resources.Transmitter*) None

Base GXF transmitter class.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.core._core.Resource*, *arg0: holoscan.core._core.ComponentSpec*)
None

setup method for the resource.

property spec

class holoscan.resources.UcxComponentSerializer

Bases: holoscan.gxf._gxf.GXFResource

UCX component serializer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

name

The name of the component serializer.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.

`gxf_cname`	The name of the component.
`gxf_context`	The GXF context of the component.
`gxf_eid`	The GXF entity ID.
`gxf_typename`	The GXF type name of the resource.
`id`	The identifier of the component.
`name`	The name of the resource.

spec	

Methods

`add_arg` (*args, **kwargs)	Overloaded function.
`gxf_initialize` (self)	Initialize the component.
`initialize` (self)	Initialize the component.
`setup` (self, s	Define the component specification.

pec)	

__init__(*self: holoscan.resources._resources.UcxComponentSerializer*, *fragment: holoscan.core._core.Fragment*, *allocator: holoscan.resources._resources.Allocator* = *None*, *name: str = 'ucx_component_serializer'*) None

UCX component serializer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

name

The name of the component serializer.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.UcxComponentSerializer*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.UcxEntitySerializer

Bases: `holoscan.gxf._gxf.GXFResource`

UCX entity serializer.

Parameters

fragment

The fragment to assign the resource to.

component_serializer

The component serializers used by the entity serializer.

verbose_warning

Whether to use verbose warnings during serialization.

name

The name of the entity serializer.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.

`gxf_typename`	The GXF type name of the resource.
`id`	The identifier of the component.
`name`	The name of the resource.

spec	

Methods

`add_arg` (*args, **kwargs)	Overloaded function.
`gxf_initialize` (self)	Initialize the component.
`initialize` (self)	Initialize the component.
`setup` (self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.UcxEntitySerializer*, *fragment: holoscan.core._core.Fragment*, *verbose_warning: bool = False*, *name: str = 'ucx_entity_serializer'*) None

UCX entity serializer.

Parameters

fragment

The fragment to assign the resource to.

component_serializer

The component serializers used by the entity serializer.

verbose_warning

Whether to use verbose warnings during serialization.

name

The name of the entity serializer.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.UcxEntitySerializer*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.UcxHoloscanComponentSerializer

Bases: holoscan.gxf._gxf.GXFResource

UCX Holoscan component serializer.

Attributes

args	The list of arguments associated with the component.

description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args, **kwargs)	Overloaded function.

`gxf_initialize`(self)	Initialize the component.
`initialize`(self)	Initialize the component.
`setup`(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.UcxHoloscanComponentSerializer*, *fragment: holoscan.core._core.Fragment*, *allocator: holoscan.resources._resources.Allocator* = *None*, *name: str = 'ucx_component_serializer'*) None

UCX Holoscan component serializer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

name

The name of the component serializer.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self:* *holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self:* *holoscan.resources._resources.UcxHoloscanComponentSerializer*, *spec:* *holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.UcxReceiver

Bases: holoscan.resources._resources.Receiver

UCX network receiver using a double-buffered queue.

New messages are first pushed to a back stage.

Parameters

fragment

The fragment to assign the resource to.

buffer

The serialization buffer used by the transmitter.

capacity

The capacity of the receiver.

policy

The policy to use (0=pop, 1=reject, 2=fault).

address

The IP address used by the transmitter.

port

The network port used by the transmitter.

name

The name of the receiver.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args,	Overloaded function.

**kwargs)	
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
setup(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.UcxReceiver*, *fragment: holoscan.core._core.Fragment*, *buffer: holoscan::UcxSerializationBuffer = None*, *capacity: int = 1*, *policy: int = 2*, *address: str = '0.0.0.0'*, *port: int = 13337*, *name: str = 'ucx_receiver'*)

None

UCX network receiver using a double-buffered queue.

New messages are first pushed to a back stage.

Parameters

fragment

The fragment to assign the resource to.

buffer

The serialization buffer used by the transmitter.

capacity

The capacity of the receiver.

policy

The policy to use (0=pop, 1=reject, 2=fault).

address

The IP address used by the transmitter.

port

The network port used by the transmitter.

name

The name of the receiver.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.UcxReceiver*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.UcxSerializationBuffer

Bases: holoscan.gxf._gxf.GXFResource

UCX serialization buffer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

buffer_size

The size of the buffer in bytes.

name

The name of the serialization buffer

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg(*args, **kwargs)	Overloaded function.
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
setup(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.UcxSerializationBuffer*, *fragment: holoscan.core._core.Fragment*, *allocator: holoscan.resources._resources.Allocator* = *None*, *buffer_size: int = 4096*, *name: str = 'serialization_buffer'*) None

UCX serialization buffer.

Parameters

fragment

The fragment to assign the resource to.

allocator

The memory allocator for tensor components.

buffer_size

The size of the buffer in bytes.

name

The name of the serialization buffer

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.UcxSerializationBuffer*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.UcxTransmitter

Bases: holoscan.resources._resources.Transmitter

UCX network transmitter using a double-buffered queue.

Messages are pushed to a back stage after they are published.

Parameters

fragment

The fragment to assign the resource to.

buffer

The serialization buffer used by the transmitter.

capacity

The capacity of the transmitter.

policy

The policy to use (0=pop, 1=reject, 2=fault).

receiver_address

The IP address used by the transmitter.

local_address

The local IP address to use for connection.

port

The network port used by the transmitter.

local_port

The local network port to use for connection.

maximum_connection_retries

The maximum number of times the transmitter will retry making a connection.

name

The name of the transmitter.

Attributes

args	The list of arguments associated with the component.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.

gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg (*args, **kwargs)	Overloaded function.
gxf_initialize (self)	Initialize the component.
initialize (self)	Initialize the component.
setup (self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.UcxTransmitter*, *fragment: holoscan.core._core.Fragment*, *buffer: holoscan::UcxSerializationBuffer = None*, *capacity: int = 1*, *policy: int = 2*, *receiver_address: str = '0.0.0.0'*, *local_address: str = '0.0.0.0'*, *port: int = 13337*, *local_port: int = 0*, *maximum_connection_retries: int = 10*, *name: str = 'ucx_transmitter'*) None

UCX network transmitter using a double-buffered queue.

Messages are pushed to a back stage after they are published.

Parameters

fragment

The fragment to assign the resource to.

buffer

The serialization buffer used by the transmitter.

capacity

The capacity of the transmitter.

policy

The policy to use (0=pop, 1=reject, 2=fault).

receiver_address

The IP address used by the transmitter.

local_address

The local IP address to use for connection.

port

The network port used by the transmitter.

local_port

The local network port to use for connection.

maximum_connection_retries

The maximum number of times the transmitter will retry making a connection.

name

The name of the transmitter.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

property args

The list of arguments associated with the component.

Returns

arglist

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.UcxTransmitter*, *spec: holoscan.core._core.ComponentSpec*) None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec

class holoscan.resources.UnboundedAllocator

Bases: holoscan.resources._resources.Allocator

Unbounded allocator.

This allocator uses dynamic memory allocation without an upper bound.

Parameters

fragment

The fragment to assign the resource to.

name

The name of the serializer.

Attributes

args	The list of arguments associated with the component.
block_size	Get the block size of the allocator.
description	YAML formatted string describing the resource.
fragment	Fragment that the resource belongs to.
gxf_cid	The GXF component ID.
gxf_cname	The name of the component.
gxf_context	The GXF context of the component.
gxf_eid	The GXF entity ID.
gxf_typename	The GXF type name of the resource.
id	The identifier of the component.
name	The name of the resource.

spec	

Methods

add_arg(*args, **kwargs)	Overloaded function.
allocate(self, size, type)	Allocate the requested amount of memory.
free(self, pointer)	Free the allocated memory
gxf_initialize(self)	Initialize the component.
initialize(self)	Initialize the component.
is_available(self, size)	Boolean representing whether the resource is available.
setup(self, spec)	Define the component specification.

__init__(*self: holoscan.resources._resources.UnboundedAllocator*, *fragment: holoscan.core._core.Fragment*, *name: str = 'unbounded_allocator'*) None

Unbounded allocator.

This allocator uses dynamic memory allocation without an upper bound.

Parameters

fragment

The fragment to assign the resource to.

name

The name of the serializer.

add_arg(**args*, ***kwargs*)

Overloaded function.

1. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.Arg) -> None

Add an argument to the component.

2. add_arg(self: holoscan.core._core.ComponentBase, arg: holoscan.core._core.ArgList) -> None

Add a list of arguments to the component.

allocate(*self: holoscan.resources._resources.Allocator*, *size: int*, *type: holoscan.resources._resources.MemoryStorageType*) int

Allocate the requested amount of memory.

Parameters

size

The amount of memory to allocate

type

Enum representing the type of memory to allocate.

Returns

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property args

The list of arguments associated with the component.

Returns

arglist

property block_size

Get the block size of the allocator.

Returns

int

The block size of the allocator. Returns 1 for byte-based allocators.

property description

YAML formatted string describing the resource.

property fragment

Fragment that the resource belongs to.

Returns

name

free(*self: holoscan.resources._resources.Allocator*, *pointer: int*) None

Free the allocated memory

Parameters

pointer

Opaque PyCapsule object representing a std::byte* pointer to the allocated memory.

property gxf_cid

The GXF component ID.

property gxf_cname

The name of the component.

property gxf_context

The GXF context of the component.

property gxf_eid

The GXF entity ID.

gxf_initialize(*self: holoscan.gxf._gxf.GXFComponent*) None

Initialize the component.

property gxf_typename

The GXF type name of the resource.

Returns

str

The GXF type name of the resource

property id

The identifier of the component.

The identifier is initially set to `-1`, and will become a valid value when the component is initialized.

With the default executor (*holoscan.gxf.GXFExecutor*), the identifier is set to the GXF component ID.

Returns

id

initialize(*self: holoscan.gxf._gxf.GXFResource*) None

Initialize the component.

is_available(*self: holoscan.resources._resources.Allocator*, *size: int*) → bool

Boolean representing whether the resource is available.

Returns

bool

Availability of the resource.

property name

The name of the resource.

Returns

name

setup(*self: holoscan.resources._resources.UnboundedAllocator*, *spec: holoscan.core._core.ComponentSpec*) → None

Define the component specification.

Parameters

spec

Component specification associated with the resource.

property spec