



## **Class NetworkContext**

# Table of contents

Inheritance Relationships

---

Class Documentation

---

- Defined in [File network\\_context.hpp](#)

## Inheritance Relationships

### Base Type

- `public holoscan::Component` ([Class Component](#))

### Derived Type

- `public holoscan::gxf::GXFNetworkContext` ([Class GXFNetworkContext](#))

## Class Documentation

class `NetworkContext` : public `holoscan::Component`

Base class for all network contexts.

This class is the base class for all network contexts (e.g. `holoscan::UcxContext`). It is used to define the common interface for all network contexts.

Subclassed by [holoscan::gxf::GXFNetworkContext](#)

Public Functions

`NetworkContext()` = default

`NetworkContext(NetworkContext&&)` = default

```
template<typename ArgT, typename ...ArgsT, typename =
std::enable_if_t<!std::is_base_of_v<::holoscan::NetworkContext, std::decay_t<ArgT>>
&& (std::is_same_v<::holoscan::Arg, std::decay_t<ArgT>> ||
std::is_same_v<::holoscan::ArgList, std::decay_t<ArgT>>)>>
inline explicit NetworkContext(ArgT &&arg, ArgsT&&... args)
```

Construct a new [NetworkContext](#) object.

`~NetworkContext()` override = default

inline [NetworkContext](#) &id(int64\_t id)

Set the NetworkContext ID.

Parameters

**id** – The ID of the network context.

Returns

The reference to this network context.

```
inline NetworkContext &name(const std::string &name) &
```

Set the name of the network context.

Parameters

**name** – The name of the network context.

Returns

The reference to the network context.

```
inline NetworkContext &&name(const std::string &name) &&
```

Set the name of the network context.

Parameters

**name** – The name of the network context.

Returns

The reference to the network context.

```
inline NetworkContext &fragment(Fragment *fragment)
```

Set the fragment of the network context.

Parameters

**fragment** – The pointer to the fragment of the network context.

Returns

The reference to the network context.

```
inline NetworkContext &spec(const std::shared_ptr<ComponentSpec> &spec)
```

Set the component specification to the network context.

Parameters

**spec** – The component specification.

Returns

The reference to the network context.

```
inline ComponentSpec *spec()
```

Get the component specification of the network context.

Returns

The pointer to the component specification.

```
inline std::shared_ptr<ComponentSpec> spec_shared()
```

Get the shared pointer to the component spec.

Returns

The shared pointer to the component spec.

```
inline void add_arg(const std::shared_ptr<Resource> &arg)
```

Add a resource to the network context.

Parameters

**arg** – The resource to add.

```
inline void add_arg(std::shared_ptr<Resource> &&arg)
```

Add a resource to the network context.

Parameters

**arg** – The resource to add.

```
inline std::unordered_map<std::string, std::shared_ptr<Resource>> &resources()
```

Get the resources of the network context.

Returns

The resources of the network context.

```
inline virtual void setup(ComponentSpec &spec)
```

Define the network context specification.

Parameters

**spec** – The reference to the component specification.

```
virtual void initialize() override
```

Initialize the network context.

This function is called after the network context is created by [holoscan::Fragment::make\\_network\\_context\(\)](#).

```
virtual YAML::Node to_yaml_node() const override
```

Get a YAML representation of the network context.

Returns

YAML node including the base component properties, component spec and resources.

Protected Functions

```
virtual void reset_graph_entities() override
```

Reset the GXF GraphEntity of any components associated with the scheduler.

## Protected Attributes

`std::unordered_map<std::string, std::shared_ptr<Resource>> resources_`

The resources used by the network context.

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024