



## **Program Listing for File ucx\_transmitter.hpp**

[Return to documentation for file \(](#)

`include/holoscan/core/resources/gxf/ucx_transmitter.hpp` )

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023-2024 NVIDIA CORPORATION &
AFFILIATES. All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed
under the Apache License, Version 2.0 (the "License"); * you may not use this file
except in compliance with the License. * You may obtain a copy of the License at * *
http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law
or agreed to in writing, software * distributed under the License is distributed on an
"AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. * See the License for the specific language governing
permissions and * limitations under the License. */ #ifndef
HOLOSCAN_CORE_RESOURCES_GXF_UCX_TRANSMITTER_HPP #define
HOLOSCAN_CORE_RESOURCES_GXF_UCX_TRANSMITTER_HPP #include <memory>
#include <string> #include <gxf/ucx/ucx_transmitter.hpp> #include
"./transmitter.hpp" #include
"holoscan/core/resources/gxf/ucx_serialization_buffer.hpp" #include
<gxf/std/transmitter.hpp> namespace nvidia::gxf { // Forward declarations class
UcxSerializationBuffer; class UcxTransmitter; } // namespace nvidia::gxf namespace
holoscan { class UcxTransmitter : public Transmitter { public:
HOLOSCAN_RESOURCE_FORWARD_ARGS_SUPER(UcxTransmitter, Transmitter)
UcxTransmitter() = default; UcxTransmitter(const std::string& name,
nvidia::gxf::Transmitter* component); const char* gxf_typename() const override {
return "nvidia::gxf::UcxTransmitter"; } void setup(ComponentSpec& spec) override;
void initialize() override; std::string receiver_address(); uint32_t port(); std::string
local_address(); uint32_t local_port(); Parameter<uint64_t> capacity_;
Parameter<uint64_t> policy_; nvidia::gxf::UcxTransmitter* get() const; private:
Parameter<std::string> receiver_address_; Parameter<std::string> local_address_;
Parameter<uint32_t> port_; Parameter<uint32_t> local_port_; Parameter<uint32_t>
maximum_connection_retries_;
Parameter<std::shared_ptr<holoscan::UcxSerializationBuffer>> buffer_; // TODO:
support GPUDevice nvidia::gxf::Resource //
nvidia::gxf::Resource<nvidia::gxf::Handle<nvidia::gxf::GPUDevice>> gpu_device_; }; } //
```

```
namespace holoscan #endif/*
```

```
HOLOSCAN_CORE_RESOURCES_GXF_UCX_TRANSMITTER_HPP */
```

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