



Program Listing for File `double_buffer_transmitter.hpp`

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

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

```
/* * SPDX-FileCopyrightText: Copyright (c) 2022-2024 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_DOUBLE_BUFFER_TRANSMITTER_HPP #define
HOLOSCAN_CORE_RESOURCES_GXF_DOUBLE_BUFFER_TRANSMITTER_HPP #include
<string> #include "gxf/std/double_buffer_transmitter.hpp" #include
"./transmitter.hpp" namespace holoscan { // Forward declarations class
AnnotatedDoubleBufferTransmitter; class DoubleBufferTransmitter : public
Transmitter { public:
HOLOSCAN_RESOURCE_FORWARD_ARGS_SUPER(DoubleBufferTransmitter,
Transmitter) DoubleBufferTransmitter() = default; DoubleBufferTransmitter(const
std::string& name, nvidia::gxf::DoubleBufferTransmitter* component);
DoubleBufferTransmitter(const std::string& name,
AnnotatedDoubleBufferTransmitter* component); const char* gxf_typename()
const override; void setup(ComponentSpec& spec) override; void track();
nvidia::gxf::DoubleBufferTransmitter* get() const; Parameter<uint64_t> capacity_;
Parameter<uint64_t> policy_; private: bool tracking_ = false; }; } // namespace
holoscan #endif/*
HOLOSCAN_CORE_RESOURCES_GXF_DOUBLE_BUFFER_TRANSMITTER_HPP */
```

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