



## **Program Listing for File `async_ping_rx.hpp`**

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

`include/holoscan/operators/async_ping_rx/async_ping_rx.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_OPERATORS_ASYNC_PING_RX_HPP #define
HOLOSCAN_OPERATORS_ASYNC_PING_RX_HPP #include <atomic> #include
<cstdint> #include <memory> #include <thread> #include
<holoscan/holoscan.hpp> namespace holoscan::ops { class AsyncPingRxOp : public
Operator { public: HOLOSCAN_OPERATOR_FORWARD_ARGS(AsyncPingRxOp)
AsyncPingRxOp() = default; void setup(OperatorSpec& spec) override; void initialize()
override; void start() override; void compute(InputContext&, OutputContext&
op_output, ExecutionContext&) override; void stop() override; void async_ping();
private: Parameter<int64_t> delay_;
Parameter<std::shared_ptr<AsynchronousCondition>> async_condition_; // internal
state std::atomic<bool> should_stop_{false}; std::thread async_thread_; }; } //
namespace holoscan::ops #endif/* HOLOSCAN_OPERATORS_ASYNC_PING_RX_HPP */
```

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