



## **Program Listing for File forward\_def.hpp**

[Return to documentation for file \(include/holoscan/core/forward\\_def.hpp\)](#)

```
/* * 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_FORWARD_DEF_HPP #define
HOLOSCAN_CORE_FORWARD_DEF_HPP #include <cinttypes> namespace holoscan {
class Application; class Arg; class ArgumentSetter; enum class ArgContainerType :
uint8_t; enum class ArgElementType; class ArgList; class ArgType; class CLIOptions;
class CLIParser; template <typename typeT> struct codec; class CodecRegistry; class
Condition; enum class ConditionType; class Config; class ComponentBase; class
Component; class ComponentSpec; class Endpoint; class ExecutionContext; class
ExtensionManager; class Executor; class Fragment; template <typename NodeT,
typename EdgeDataElementT> class Graph; class GXFParameterAdaptor; class
InputContext; class IOSpec; class Logger; class Message; class MessageLabel; class
Operator; class OperatorSpec; class OperatorTimestampLabel; class OutputContext;
template <typename ValueT> class MetaParameter; template <typename ValueT>
using Parameter = MetaParameter<ValueT>; class ParameterWrapper; enum class
ParameterFlag; class NetworkContext; class Resource; class Scheduler; //
holoscan::gxf namespace gxf { class Entity; class GXFComponent; class GXFCondition;
class GXFInputContext; class GXFOutputContext; class GXFResource; class
GXFExtensionManager; class GXFNetworkContext; class GXFScheduler; } //
namespace gxf // Distributed Application class AppDriver; class AppWorker; //
holoscan::service namespace service { class AppDriverServer; class AppDriverClient;
class AppWorkerServer; class AppWorkerClient; } // namespace service //
NetworkContexts class UcxContext; // Schedulers enum class SchedulerType; class
EventBasedScheduler; class GreedyScheduler; class MultiThreadScheduler; //
holoscan::ops namespace ops { class GXFOperator; } // Conditions class
AsynchronousCondition; class BooleanCondition; class CountCondition; class
```

```
DownstreamMessageAffordableCondition; class MessageAvailableCondition; class
PeriodicCondition; // Resources class Allocator; class
AnnotatedDoubleBufferReceiver; class AnnotatedDoubleBufferTransmitter; class
Clock; class BlockMemoryPool; class CudaStreamPool; class DoubleBufferReceiver;
class DoubleBufferTransmitter; class ManualClock; class Receiver; class
RealtimeClock; class SerializationBuffer; class StdComponentSerializer; class
StdEntitySerializer; class Transmitter; class UcxComponentSerializer; class
UcxEntitySerializer; class UcxHoloscanComponentSerializer; class UcxReceiver; class
UcxSerializationBuffer; class UcxTransmitter; class UnboundedAllocator; // Domain
objects class Tensor; class TensorMap; } // namespace holoscan namespace YAML {
class Node; } // namespace YAML #endif/* HOLOSCAN_CORE_FORWARD_DEF_HPP */
```

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