# Class VirtualOperator

# Table of contents

- Defined in File virtual_operator.hpp

# Inheritance Relationships

## Base Type

- `public holoscan::Operator` (Class Operator)

## Derived Types

- `public holoscan::ops::VirtualReceiverOp` (Class VirtualReceiverOp)
- `public holoscan::ops::VirtualTransmitterOp` (Class VirtualTransmitterOp)

# Class Documentation

class VirtualOperator : public holoscan::Operator

Virtual operator.

This class is used when connecting fragments with transmitters and receivers (such as UCXTransmitter and UCXReceiver) that communicate with other fragments.

The input/output port of an operator in the fragment can be connected to this virtual operator so that the internal graph initialization mechanism (such as implicit broadcasting or multi-receivers) can be applied to the port of the operator connected to this operator in the same fragment.

Subclassed by holoscan::ops::VirtualReceiverOp, holoscan::ops::VirtualTransmitterOp

Public Functions

template<typename StringT, typename = std::enable_if_t<std::is_constructible_v<std::string, StringT>>>
inline VirtualOperator(StringT port_name, IOSpec::ConnectorType connector_type, ArgList arg_list)

inline VirtualOperator()

virtual void initialize() override

Initialize the Virtual operator.

This function does not call the Operator::initialize() method that is called when the fragment is initialized by Executor::initialize_fragment().

Instead, it just sets the operator type to `holoscan::Operator::OperatorType::kVirtual`.

inline const std::string &port_name() const

> Get the name of the port of the operator connected to this operator in the same fragment.
>
> Returns
>
> The name of the port of the operator connected to this operator in the same fragment.

inline void port_name(const std::string &port_name)

> Set the name of the port of the operator connected to this operator in the same fragment.
>
> Parameters
>
> **port_name** – The name of the port of the operator connected to this operator in the same fragment.

inline IOSpec::ConnectorType connector_type() const

> Get the connector type of this operator.
>
> Returns
>
> The connector type of this operator.

inline const ArgList &arg_list() const

> Get the argument list of this operator.

Returns

The argument list of this operator.

IOSpec *input_spec()

Get the input specification for this operator.

Returns

The pointer to the input specification for this operator.

IOSpec *output_spec()

Get the output specification for this operator.

Returns

The pointer to the output specification for this operator.

inline IOSpec::IOType io_type() const

Get the IO type of this operator.

Returns

The IO type of this operator.

Protected Attributes

std::string port_name_

The name of the port of the operator connected to this operator in the same fragment.

IOSpec::ConnectorType connector_type_

The connector type of this operator.

ArgList arg_list_

The argument list of this operator.

IOSpec *input_spec_ = nullptr

The pointer to the input specification for this operator.

IOSpec *output_spec_ = nullptr

The pointer to the output specification for this operator.

IOSpec::IOType io_type_ = IOSpec::IOType::kInput