



Class VideoStreamRecorderOp

Table of contents

Inheritance Relationships

Class Documentation

- Defined in [File video_stream_recorder.hpp](#)

Inheritance Relationships

Base Type

- `public holoscan::Operator` ([Class Operator](#))

Class Documentation

class VideoStreamRecorderOp : public holoscan::Operator

[Operator](#) class to record a video stream to a file.

==Named Inputs==

- **input**: `nvidia::gxf::Tensor`
 - A message containing a video frame to serialize to disk. The input tensor can be on either the CPU or GPU. This data location will be recorded as part of the metadata serialized to disk and if the data is later read back in via `VideoStreamReplayerOp`, the tensor output of that operator will be on the same device (CPU or GPU).

==Parameters==

- **directory**: Directory path for storing files.
- **basename**: User specified file name without extension.
- **flush_on_tick**: Flushes output buffer on every tick when `true`. Optional (default: `false`).

Public Functions

HOLOSCAN_OPERATOR_FORWARD_ARGS (VideoStreamRecorderOp)

VideoStreamRecorderOp()=default

~VideoStreamRecorderOp() override

virtual void setup(OperatorSpec &spec) override

Define the operator specification.

Parameters

spec – The reference to the operator specification.

virtual void initialize() override

Initialize the operator.

This function is called when the fragment is initialized by Executor::initialize_fragment().

virtual void compute(InputContext &op_input, OutputContext &op_output, ExecutionContext &context) override

Implement the compute method.

This method is called by the runtime multiple times. The runtime calls this method until the operator is stopped.

Parameters

- **op_input** – The input context of the operator.
- **op_output** – The output context of the operator.
- **context** – The execution context of the operator.

virtual void stop() override

Implement the shutdown logic of the operator.

This method is called multiple times over the lifecycle of the operator according to the order defined in the lifecycle, and used for heavy deinitialization tasks such as deallocation of all resources previously assigned in start.