Built-in Operators and Extensions

Table of contents

The units of work of Holoscan applications are implemented within Operators, as described in the core concepts of the SDK. The operators included in the SDK provide domain-agnostic functionalities such as IO, machine learning inference, processing, and visualization, optimized for AI streaming pipelines, relying on a set of Core Technologies.

Operators

The operators below are defined under the `holoscan::ops` namespace for C++ and CMake, and under the `holoscan.operators` module in Python.

Class	CMake target/lib	Documentation
AJASourceOp	`aja`	`C++` / `Python`
BayerDemosaicOp	`bayer_demosaic`	`C++` / `Python`
FormatConverterOp	`format_converter`	`C++` / `Python`
HolovizOp	`holoviz`	`C++` / `Python`
InferenceOp	`inference`	`C++` / `Python`
InferenceProcessorOp	`inference_processor`	`C++` / `Python`
PingRxOp	`ping_rx`	`C++` / `Python`
PingTxOp	`ping_tx`	`C++` / `Python`
SegmentationPostprocessorOp	`segmentation_postprocessor`	`C++` / `Python`
VideoStreamRecorderOp	`video_stream_recorder`	`C++` / `Python`
VideoStreamReplayerOp	`video_stream_replayer`	`C++` / `Python`
V4L2VideoCaptureOp	`v4l2`	`C++` / `Python`

Given an instance of an operator class, you can print a human-readable description of its specification to inspect the inputs, outputs, and parameters that can be configured on that operator class:

Ingested Tab Module

Note

The Holoscan SDK uses meta-programming with templating and `std::any` to support arbitrary data types. Because of this, some type information (and therefore values) might not be retrievable by the `description` API. If more details are needed, we recommend inspecting the list of `Parameter` members in the operator header to identify their type.

Extensions

The Holoscan SDK also includes some GXF extensions with GXF codelets, which are typically wrapped as operators, or present for legacy reasons. In addition to the core GXF extensions (std, cuda, serialization, multimedia) listed here, the Holoscan SDK includes the following GXF extensions:

- gxf_holoscan_wrapper
- ucx_holoscan

GXF Holoscan Wrapper

The `gxf_holoscan_wrapper` extension includes the `holoscan::gxf::OperatorWrapper` codelet. It is used as a utility base class to wrap a holoscan operator to interface with the GXF framework.

Learn more about it in the Using Holoscan Operators in GXF Applications section.

UCX (Holoscan)

The `ucx_holoscan` extension includes `nvidia::holoscan::UcxHoloscanComponentSerializer` which is a `nvidia::gxf::ComponentSerializer` that handles serialization of `holoscan::Message` and `holoscan::Tensor` types for transmission using the Unified Communication X (UCX)

library. UCX is the library used by Holoscan SDK to enable communication of data between fragments in distributed applications.

> **Note**
>
> The `UcxHoloscanComponentSerializer` is intended for use in combination with other UCX components defined in the GXF UCX extension. Specifically, it can be used by the `UcxEntitySerializer` where it can operate alongside the `UcxComponentSerializer` that serializes GXF-specific types (`nvidia::gxf::Tensor`, `nvidia::gxf::VideoBuffer`, etc.). This way both GXF and Holoscan types can be serialized by distributed applications.

HoloHub

Visit the HoloHub repository to find a collection of additional Holoscan operators and extensions.