



Program Listing for File process_manager.hpp

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

`modules/holoinfer/src/manager/process_manager.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_PROCESS_MANAGER_H #define _HOLOSCAN_PROCESS_MANAGER_H
#include <future> #include <iostream> #include <map> #include <memory>
#include <string> #include <vector> #include <holoinfer.hpp> #include
<holoinfer_utils.hpp> #include <process/data_processor.hpp> namespace holoscan
{ namespace inference { /* * @brief Manager class for processing */ class
ManagerProcessor { public: /* * @brief Default Constructor */ ManagerProcessor()
{} /* * @brief Destructor */ ~ManagerProcessor() {} /* * @brief Initializes the
underlying contexts and checks the validity of operations * * @param
process_operations Map where tensor name is the key, and operations to perform
on * the tensor as vector of strings. Each value in the vector of strings is the
supported * operation. * * @return InferStatus with appropriate code and message
*/ InferStatus initialize(const MultiMappings& process_operations, const std::string
config_path); /* * @brief Executes post processing operations and generates the
result * * @param tensor_oper_map Map with tensor name as the key, and
operations to perform on * the tensor as vector of strings. * @param
in_out_tensor_map Map with input tensor name as the key, and generated output
tensor * names is the value (as vector of strings) * @param inferred_result_map
Map with output tensor name as key, and related DataBuffer as * value * @param
dimension_map Map with tensor name as key and related output dimension as
value. * @return InferStatus with appropriate code and message */ InferStatus
process(const MultiMappings& tensor_oper_map, const MultiMappings&
in_out_tensor_map, DataMap& inferred_result_map, const std::map<std::string,
std::vector<int>>& dimension_map); /* * @brief Executes post processing
```

```

operations for multi tensor I/O * * @param tensor_name String containing input
tensor names separated by : * @param tensor_oper_map Map with tensor names
as the key, and operations to perform on * the tensor as vector of strings. *
@param inferred_result_map Map Contains output tensor name as key, and related
DataBuffer as * value * @param dimension_map Map with tensor name as key and
related output dimension as value. * @return InferStatus with appropriate code and
message */ InferStatus process_multi_tensor_operation( const std::string
tensor_name, const std::vector<std::string>& tensor_oper_map, DataMap&
inferred_result_map, const std::map<std::string, std::vector<int>>&
dimension_map); /* * @brief Get processed data * * @return DataMap with tensor
name as key and related DataBuffer as value */ DataMap get_processed_data()
const; /* * @brief Get processed data dimensions * * @return DataMap with tensor
name as key and related dimension as value */ DimType get_processed_data_dims()
const; private: std::unique_ptr<DataProcessor> infer_data_; DataMap
processed_data_map_; DimType processed_dims_map_; };
std::unique_ptr<ManagerProcessor> process_manager; } // namespace inference } //
namespace holoscan #endif

```

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