



## **Program Listing for File extension\_manager.hpp**

[Return to documentation for file](#) ( `include/holoscan/core/extension_manager.hpp` )

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023 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_EXTENSION_MANAGER_HPP #define
HOLOSCAN_CORE_EXTENSION_MANAGER_HPP #include <yaml-cpp/yaml.h>
#include <set> #include <string> #include "holoscan/core/common.hpp"
namespace holoscan { class ExtensionManager { public: explicit
ExtensionManager(void* context) : context_(context) {} virtual ~ExtensionManager()
= default; virtual void refresh() {} virtual bool load_extension(const std::string&
file_name, bool no_error_message = false, const std::string& search_path_envs =
"HOLOSCAN_LIB_PATH") { (void)file_name; (void)no_error_message;
(void)search_path_envs; return false; } virtual bool load_extensions_from_yaml(const
YAML::Node& node, bool no_error_message = false, const std::string&
search_path_envs = "HOLOSCAN_LIB_PATH", const std::string& key = "extensions") {
(void)node; (void)no_error_message; (void)search_path_envs; (void)key; return false;
} protected: void* context_ = nullptr; }; } // namespace holoscan #endif/*
HOLOSCAN_CORE_EXTENSION_MANAGER_HPP */
```

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