



## **Program Listing for File `gxf_extension_manager.hpp`**

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

`include/holoscan/core/gxf/gxf_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
INCLUDE_HOLOSCAN_CORE_GXF_GXF_EXTENSION_MANAGER_HPP #define
INCLUDE_HOLOSCAN_CORE_GXF_GXF_EXTENSION_MANAGER_HPP #include <yaml-
cpp/yaml.h> #include <set> #include <string> #include <vector> #include
"gxf/core/gxf.h" #include "gxf/std/extension.hpp" #include
"holoscan/core/common.hpp" #include "holoscan/core/extension_manager.hpp"
namespace holoscan::gxf { namespace { // Method name to get the GXF extension
factory constexpr const char* kGxfExtensionFactoryName = "GxfExtensionFactory";
// Max size of extensions constexpr int kGXFExtensionsMaxSize = 1024; // Method
signature for the GXF extension factory using GxfExtensionFactory =
gxf_result_t(void**); } // namespace class GXFExtensionManager : public
ExtensionManager { public: explicit GXFExtensionManager(gxf_context_t context);
~GXFExtensionManager() override; void refresh() override; bool
load_extension(const std::string& file_name, bool no_error_message = false, const
std::string& search_path_envs = "HOLOSCAN_LIB_PATH") override; 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") override; bool load_extension(nvidia::gxf::Extension*
extension, void* handle = nullptr); bool is_extension_loaded(gxf_tid_t tid); static
std::vector<std::string> tokenize(const std::string& str, const std::string& delimiters);
protected: gxf_tid_t extension_tid_list_[kGXFExtensionsMaxSize] = {};
gxf_runtime_info runtime_info_{nullptr, kGXFExtensionsMaxSize,
extension_tid_list_}; std::set<gxf_tid_t> extension_tids_; std::set<void*>
```

```
extension_handles_; }; } // namespace holoscan::gxf #endif/*  
INCLUDE_HOLOSCAN_CORE_GXF_GXF_EXTENSION_MANAGER_HPP */
```

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