



Program Listing for File gxf_component.hpp

[Return to documentation for file \(include/holoscan/core/gxf/gxf_component.hpp\)](#)

```
/* * 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_CORE_GXF_GXF_COMPONENT_HPP #define
HOLOSCAN_CORE_GXF_GXF_COMPONENT_HPP #include <gxf/core/gxf.h> #include
<iostream> #include <memory> #include <string> #include
<gxf/app/graph_entity.hpp> #include "../parameter.hpp" #include "./gxf_utils.hpp"
namespace holoscan::gxf { class GXFComponent { public: GXFComponent() =
default; virtual ~GXFComponent() = default; virtual const char* gxf_typename()
const { return "nvidia::gxf::Component"; } void gxf_context(gxf_context_t
gxf_context) { gxf_context_ = gxf_context; } gxf_context_t gxf_context() const { return
gxf_context_; } void gxf_eid(gxf_uid_t gxf_eid) { gxf_eid_ = gxf_eid; } gxf_uid_t gxf_eid()
const { return gxf_eid_; } void gxf_tid(gxf_tid_t gxf_tid) { gxf_tid_ = gxf_tid; } gxf_tid_t
gxf_tid() const { return gxf_tid_; } void gxf_cid(gxf_uid_t gxf_cid) { gxf_cid_ = gxf_cid; }
gxf_uid_t gxf_cid() const { return gxf_cid_; } std::string& gxf_cname() { return
gxf_cname_; } void gxf_cname(const std::string& name) { gxf_cname_ = name; }
std::shared_ptr<nvidia::gxf::GraphEntity> gxf_graph_entity() { return
gxf_graph_entity_; } void gxf_graph_entity(std::shared_ptr<nvidia::gxf::GraphEntity>
graph_entity) { gxf_graph_entity_ = graph_entity; } void* gxf_cptr() { return gxf_cptr_;
} nvidia::gxf::Handle<nvidia::gxf::Component> gxf_component() { return
gxf_component_; } void gxf_initialize(); void set_gxf_parameter(const std::string&
component_name, const std::string& key, ParameterWrapper& param_wrap); void
reset_gxf_graph_entity() { gxf_graph_entity_.reset(); } protected: gxf_context_t
gxf_context_ = nullptr; gxf_uid_t gxf_eid_ = 0; gxf_tid_t gxf_tid_ = {}; gxf_uid_t gxf_cid_
= 0; std::shared_ptr<nvidia::gxf::GraphEntity> gxf_graph_entity_; std::string
gxf_cname_; // TODO: remove gxf_cptr_ and use the Component Handle everywhere
instead? nvidia::gxf::Handle<nvidia::gxf::Component> gxf_component_; void*
```

```
gxf_cptr_ = nullptr; }; } // namespace holoscan::gxf #endif/*  
HOLOSCAN_CORE_GXF_GXF_COMPONENT_HPP */
```

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