



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

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

```
/* * 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_SIGNAL_HANDLER_HPP #define
HOLOSCAN_CORE_SIGNAL_HANDLER_HPP #include <signal> #include <functional>
#include <memory> #include <mutex> #include <unordered_map> #include
<utility> namespace holoscan { extern void static_handle_signal(int signal); class
SignalHandler { public: using GlobalSignalHandlerMap = std::unordered_map<int,
std::function<void(int)>>; using SignalHandlerMap = std::unordered_map<int,
std::function<void(void*, int)>>; using ContextSignalHandlerMap =
std::unordered_map<void*, SignalHandlerMap>; static SignalHandler&
get_instance(); static void static_handle_signal(int signal); static void
install_signal_handler(int signal = 0); static void register_global_signal_handler(int
signal, std::function<void(int)> handler, bool overwrite = false); static void
register_signal_handler(void* context, int signal, std::function<void(void*, int)>
handler, bool overwrite = false); static void unregister_global_signal_handler(int
signal); static void unregister_signal_handler(void* context, int signal); static void
clear_all_signal_handlers(); static void clear_global_signal_handlers(); static void
clear_signal_handlers(); private: SignalHandler(); ~SignalHandler(); void
install_signal_handler_impl(int signal = 0); void
register_global_signal_handler_impl(int signal, std::function<void(int)> handler, bool
overwrite = false); void register_signal_handler_impl(void* context, int signal,
std::function<void(void*, int)> handler, bool overwrite = false); void
unregister_global_signal_handler_impl(int signal); void
unregister_signal_handler_impl(void* context, int signal); void handle_signal(int
signal); GlobalSignalHandlerMap global_signal_handlers_; ContextSignalHandlerMap
signal_handlers_; std::recursive_mutex signal_handlers_mutex_; static struct
```

```
sigaction signal_handler_; std::unordered_map<int, struct sigaction>  
old_signal_handlers_; }; } // namespace holoscan #endif//  
HOLOSCAN_CORE_SIGNAL_HANDLER_HPP
```

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