



## **Program Listing for File timer.hpp**

[Return to documentation for file \(include/holoscan/utils/timer.hpp\)](#)

```
/* * SPDX-FileCopyrightText: Copyright (c) 2022 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_UTILS_TIMER_HPP #define HOLOSCAN_UTILS_TIMER_HPP #include
<fmt/format.h> #include <chrono> namespace holoscan { class Timer { public:
explicit Timer(const char* message, bool auto_start = true, bool auto_output = true)
{ message_ = message; is_auto_output_ = auto_output; if (auto_start) {
elapsed_seconds_ = 0.0; start_ = std::chrono::high_resolution_clock::now(); } }
~Timer() { if (elapsed_seconds_ <= 0.0) { end_ =
std::chrono::high_resolution_clock::now(); elapsed_seconds_ =
std::chrono::duration_cast<std::chrono::duration<double>>(end_ - start_).count(); } if
(is_auto_output_) { print(); } } void start() { elapsed_seconds_ = 0.0; start_ =
std::chrono::high_resolution_clock::now(); } double stop() { end_ =
std::chrono::high_resolution_clock::now(); elapsed_seconds_ =
std::chrono::duration_cast<std::chrono::duration<double>>(end_ - start_).count();
return elapsed_seconds_; } double elapsed_time() { return elapsed_seconds_; } void
print(const char* message = nullptr) { if (message) { fmt::print(stderr, message,
elapsed_seconds_); } else { fmt::print(stderr, message_, elapsed_seconds_); } }
private: const char* message_ = nullptr; bool is_auto_output_ = false; double
elapsed_seconds_ = -1; std::chrono::time_point<std::chrono::system_clock> start_{};
std::chrono::time_point<std::chrono::system_clock> end_{}; }; } // namespace
holoscan #endif/* HOLOSCAN_UTILS_TIMER_HPP */
```

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