



Program Listing for File periodic.hpp

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

```
/* * SPDX-FileCopyrightText: Copyright (c) 2023-2024 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_CONDITIONS_GXF_PERIODIC_HPP #define
HOLOSCAN_CORE_CONDITIONS_GXF_PERIODIC_HPP #include <string> #include
<chrono> #include "../gxf/gxf_condition.hpp" namespace holoscan { class
PeriodicCondition : public gxf::GXFCondition { public:
HOLOSCAN_CONDITION_FORWARD_ARGS_SUPER(PeriodicCondition, GXFCondition)
PeriodicCondition() = default; explicit PeriodicCondition(int64_t recess_period_ns);
template <typename Rep, typename Period> explicit
PeriodicCondition(std::chrono::duration<Rep, Period> recess_period_duration) {
recess_period_ns_ = std::chrono::duration_cast<std::chrono::nanoseconds>
(recess_period_duration).count(); recess_period_ = std::to_string(recess_period_ns_);
} PeriodicCondition(const std::string& name, nvidia::gxf::PeriodicSchedulingTerm*
term); const char* gxf_typename() const override { return
"nvidia::gxf::PeriodicSchedulingTerm"; } void setup(ComponentSpec& spec)
override; void recess_period(int64_t recess_period_ns); template <typename Rep,
typename Period> void recess_period(std::chrono::duration<Rep, Period>
recess_period_duration) { int64_t recess_period_ns =
std::chrono::duration_cast<std::chrono::nanoseconds>
(recess_period_duration).count(); recess_period(recess_period_ns); } int64_t
recess_period_ns(); int64_t last_run_timestamp();
nvidia::gxf::PeriodicSchedulingTerm* get() const; private: Parameter<std::string>
recess_period_; int64_t recess_period_ns_ = 0; }; } // namespace holoscan #endif/*
HOLOSCAN_CORE_CONDITIONS_GXF_PERIODIC_HPP */
```

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