



Program Listing for File boolean.hpp

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

```
/* * 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_CONDITIONS_GXF_BOOLEAN_HPP #define
HOLOSCAN_CORE_CONDITIONS_GXF_BOOLEAN_HPP #include <string> #include
"../gxf/gxf_condition.hpp" namespace holoscan { class BooleanCondition : public
gxf::GXFCondition { public:
HOLOSCAN_CONDITION_FORWARD_ARGS_SUPER(BooleanCondition, GXFCondition)
explicit BooleanCondition(bool enable_tick = true) : enable_tick_(enable_tick) {}
BooleanCondition(const std::string& name, nvidia::gxf::BooleanSchedulingTerm*
term); const char* gxf_typename() const override { return
"nvidia::gxf::BooleanSchedulingTerm"; } void enable_tick(); void disable_tick(); bool
check_tick_enabled(); void setup(ComponentSpec& spec) override;
nvidia::gxf::BooleanSchedulingTerm* get() const; private: Parameter<bool>
enable_tick_; }; } // namespace holoscan #endif/*
HOLOSCAN_CORE_CONDITIONS_GXF_BOOLEAN_HPP */
```

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