



Class PeriodicCondition

Table of contents

Inheritance Relationships

Class Documentation

- Defined in [File periodic.hpp](#)

Inheritance Relationships

Base Type

- `public holoscan::gfx::GXFCondition` ([Class GXFCondition](#))

Class Documentation

class `PeriodicCondition` : public `holoscan::gfx::GXFCondition`

[Condition](#) class to support periodic execution of operators.

The recess (pause) period indicates the minimum amount of time that must elapse before the `compute()` method can be executed again. The period is specified as a string containing a number and an (optional) unit. If no unit is given the value is assumed to be in nanoseconds. Supported units are: ms, s, hz (case insensitive)

For example: "10000000", "10ms", "1s", "50Hz".

Using `std::string` as the first parameter of `make_condition<T>` is only available through `Arg`.

For example: `Arg` or `Arg`.

The recess (pause) period can also be specified as an integer value (type `int64_t`) in nanoseconds or as a value of type `std::chrono::duration<Rep, Period>` (see <https://en.cppreference.com/w/cpp/chrono/duration>).

Example:

- `1000` (1000 nanoseconds == 1 microsecond)
- `5ns`, `10us`, `1ms`, `0.5s`, `1min`, `0.5h`, etc.
 - requires `#include <chrono>` and `using namespace std::chrono_literals;`

- `std::chrono::milliseconds(10)`
- `std::chrono::duration<double, std::milli>(10)`
- `std::chrono::duration<double, std::ratio<1, 1000>>(10)`

This class wraps GXF SchedulingTerm(`nvidia::gxf::PeriodicSchedulingTerm`).

Public Functions

```
template<typename ArgT, typename ...ArgsT, typename =
std::enable_if_t<!std::is_base_of_v<::holoscan::Condition, std::decay_t<ArgT>>> &&
(std::is_same_v<::holoscan::Arg, std::decay_t<ArgT>> ||
std::is_same_v<::holoscan::ArgList, std::decay_t<ArgT>>>>
inline PeriodicCondition(ArgT &&arg, ArgsT&&... args)
```

PeriodicCondition() = default

explicit PeriodicCondition(int64_t recess_period_ns)

```
template<typename Rep, typename Period>
inline explicit PeriodicCondition(std::chrono::duration<Rep, Period>
recess_period_duration)
```

PeriodicCondition(const std::string &name, nvidia::gxf::PeriodicSchedulingTerm *term)

inline virtual const char *gxf_typename() const override

virtual void setup(ComponentSpec &spec) override

Define the condition specification.

Parameters

spec – The reference to the component specification.

void recess_period(int64_t recess_period_ns)

Set recess period.

Note that calling this method doesn't affect the behavior of the condition once the condition is initialized.

Parameters

recess_period_ns – The integer representing recess period in nanoseconds.

```
template<typename Rep, typename Period>  
inline void recess_period(std::chrono::duration<Rep, Period>  
recess_period_duration)
```

Set recess period.

Note that calling this method doesn't affect the behavior of the condition once the condition is initialized.

Parameters

recess_period_duration – The recess period of type `std::chrono::duration`.

```
int64_t recess_period_ns()
```

Get recess period in nano seconds.

Returns

The minimum time which needs to elapse between two executions (in nano seconds)

```
int64_t last_run_timestamp()
```

Get the last run time stamp.

Returns

The last run time stamp.

```
nvidia::gfx::PeriodicSchedulingTerm *get() const
```