



## **Class Clock**

# Table of contents

Inheritance Relationships

---

Class Documentation

---

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

## Inheritance Relationships

### Base Type

- `public holoscan::gxf::GXFResource` ([Class GXFResource](#))

### Derived Types

- `public holoscan::ManualClock` ([Class ManualClock](#))
- `public holoscan::RealtimeClock` ([Class RealtimeClock](#))

## Class Documentation

class Clock : public holoscan::gxf::GXFResource

Base clock class.

[Clock](#) classes are used by a [Scheduler](#) to control the flow of time in an application.

Subclassed by [holoscan::ManualClock](#), [holoscan::RealtimeClock](#)

Public Functions

```
template<typename ArgT, typename ...ArgsT, typename =
std::enable_if_t<!std::is_base_of_v<::holoscan::Resource, 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 Clock(ArgT &&arg, ArgsT&&... args)
```

Clock() = default

Clock(const std::string &name, nvidia::gxf::Clock \*component)

inline virtual const char \*gxf\_typename() const override

The underlying GXF component's name.

```
virtual double time() const = 0
```

The current time of the clock. Time is measured in seconds.

```
virtual int64_t timestamp() const = 0
```

The current timestamp of the clock. Timestamps are measured in nanoseconds.

```
virtual void sleep_for(int64_t duration_ns) = 0
```

Waits until the given duration has elapsed on the clock.

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

Set a duration to sleep.

Parameters

**duration** – The sleep duration of type `std::chrono::duration`.

```
virtual void sleep_until(int64_t target_time_ns) = 0
```

Waits until the given target time.

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

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