Symbulation

Release 0.0.1

Anya Vostinar

Mar 23, 2022

USING SYMBULATION

Authors Anya Vostinar and contributors.

GitHub https://github.com/anyaevostinar/SymbulationEmp

This is a software model of symbiosis which allows for evolution of parasitism and mutualism.

Click here to try it out in your web browser!

Checkout our quick start guides for more information on getting started.

Search for any function, class or variable using the search feature in the top left!

CHAPTER

ONE

TABLE OF CONTENTS

1.1 Projects using Symbulation

This is a running list of the projects that have used Symbulation! If you would like to have your project added to this list, please include a short description of you work, and a link to any git repository that you may have. You can complete a pull request to add your information to this list. See Getting Started with Symbulation Development for information on how to complete a pull request.

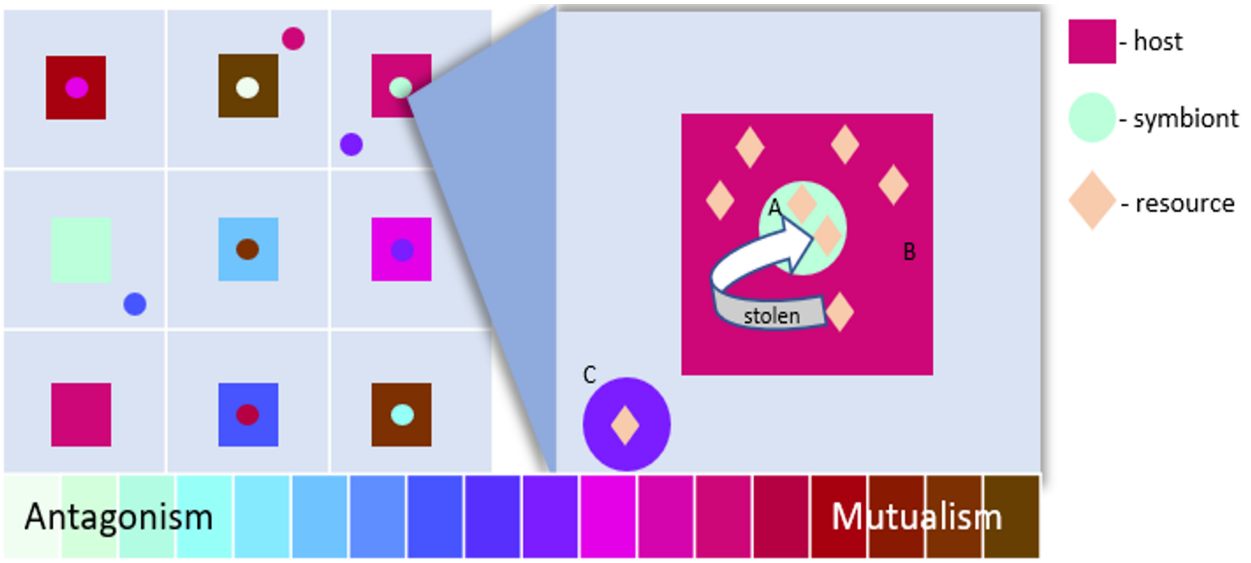
- An investigation into the conditions for the evolution of endosymbiosis
  - Symbulation: v0.2
  - Git: https://github.com/anyaevostinar/Evolution-of-Endosymbiosis-Paper
- An investigation into the evolution of lysis in bacteriophage
  - Symbulation: v0.2
  - Git: https://github.com/anyaevostinar/Evolution-of-Lysogeny-Paper
- An investigation into the dirty transmission hypothesis
  - Symbulation: v0.2
  - Git: https://github.com/anyaevostinar/Dirty-Transmission-Hypothesis-Paper
- An investigation into the impacts of within-host ecology on host-symbiont mutualism vs. parasitism
  - Symbulation: v0.1
  - Git: https://github.com/DolsonLab/2021-Symbiont-Ecology

1.2 Quick Start Guides

Contents:

1.2.1 Overview

The goal of Symbulation is to provide an evolutionary agent-based framework for studying the evolution of symbiosis. It supports a population of hosts and a population of symbionts. The symbionts can live inside of hosts (endosymbionts) or outside of hosts (free-living and/or ectosymbionts). The hosts and symbionts can engage in a relationship anywhere between full antagonism/parasitism to mutualism.



If you haven't already, we recommend first taking a look at the browser-based GUI.

Then you can start with the *Default Mode* guide, which includes running Symbulation both from the command line and a local browser-based GUI.

1.2.2 Default Mode

Symbulation allows you to compile your C++ to either:

- running at the command line (e.g., native machine code) and
- running in the web browser (e.g., javascript assembly code).

This how-to aims to walk you through all the nitty gritty required to successfully compile a simple "default" executable for running at the command line.

Install: Native C++ Compiler

In the Unix-verse (e.g., Linux / MacOS) commonly used compilers include `gcc` and `clang`. From this point onwards, we assume that you're working with `gcc`.Unless you really know what you're doing, you'll want to have `gcc` installed. The good news is: you might already!

Bring up a terminal and try entering:

```
which gcc
```

If `which` spits out a path, then you have `gcc` installed! If `which` says "not found," you'll need to go ahead and install `gcc`. For Linux users, your package manager (e.g., `yum`, `apt`, etc.) is probably the way to go about this. For MacOS users, you'll need to get Apple's "Command Line Tools for Xcode." Either way, give it a quick web search

(e.g., "install gcc on [my operating system]") and there should be plenty of how-to guides that walk you through step-by-step.

For Windows users, things are a bit more complicated. We recommend using Windows Subsystem for Linux and then following the steps above for `gcc`.

Compile & Run: Command Line

Assuming you haven't already pulled down a clone of Symbulation, let's get your working environment all set.

Using Cookiecutter

The easiest thing to do is use our Cookiecutter template.

- Install the latest Cookiecutter:

```
pip install -U cookiecutter
```

- Generate a Symbulation project:

```
cookiecutter https://github.com/anyaevostinar/SymbulationProjectTemplate.git
```

- Change directory into your newly created project:

```
cd SymbulationProject
```

Without Cookiecutter

If you have something against delicious cookies and want to do without Cookiecutter, here are the steps. (Note that these steps don't include making Data and Analysis files, which is assumed for the rest of the guide. If you are doing it this way, we assume you have your own preference for organizing things.)

1. Clone Symbulation:

```
git clone https://github.com/anyaevostinar/SymbulationEmp
```

2. In the same level as `SymbulationEmp`, you will need a recursive copy of Empirical:

```
git clone --recursive https://github.com/devosoft/Empirical.git
```

Compiling

Regardless of how you got everything downloaded, it's now time to compile!

```
cd SymbulationEmp
make
```

Which command you use to run the executable depends on your version. If you are using version 0.1:

```
./symbulation
```

If you are using a later version (or the `refactor` branch, which has the development version):

```
./symbulation_default
```

You can make changes to the file `SymSettings.cfg` to change the configuration options without needing to re-compile. These can also be altered at the command line:

```
./<executable file> -<Option to change> <new value>
```

For example:

```
./symbulation_default -VERTICAL_TRANSMISSION 0.5
```

To see how to use our workflow and scripts to collect and analyze data, please proceed to the Collecting Data quickstart guide!

Install: Web GUI

These instructions are for if you want to make *changes* to the web GUI and run it locally. You don't need to do this to be able to use Symbulation's web browser; you can instead access it here: https://anyaevostinar.github.io/SymbulationEmp/web/symbulation.html.

First, you'll need Emscripten. We recommend putting it in the same directory that you have SymbulationEmp and Empirical:

```
git clone https://github.com/emscripten-core/emsdk.git
cd emsdk
./emsdk install 3.1.2
./emsdk activate 3.1.2
```

Then every time that you want to build and run Symbulation's web GUI, you'll need to load Emscripten into your `PATH` for a given terminal session. If you've closed and opened a new terminal, you'll need to do this again (this is what people usually forget to do after they've been away for a while):

```
cd emsdk
source ./emsdk_env.sh
```

Then navigate back to SymbulationEmp and build the web version:

```
cd SymbulationEmp
make web
```

```
./symbulation_default
```

Web

You don't need to do this to be able to use Symbulation's web browser since it is hosted here: https://anyaevostinar.github.io/SymbulationEmp/web/symbulation.html.

However, if you want to make changes to the web GUI and run it locally, this is how you do that.

First, you'll need Emscripten, we recommend putting it in the same directory that you have SymbulationEmp and Empirical:

```
git clone https://github.com/emscripten-core/emsdk.git
cd emsdk
./emsdk install 2.0.5
./emsdk activate 2.0.5
```

Then every time that you want to build and run Symbulation's web GUI, you'll need to load Emscripten into your `PATH` for a given terminal session. If you've closed and opened a new terminal, you'll need to do this again (this is what people usually forget to do after they've been away for a while):

```
cd emsdk
source ./emsdk_env.sh
```

Then navigate back to SymbulationEmp and build the web version:

```
cd SymbulationEmp
make web
```

Now you have the website all built and in the `web` folder, so just navigate into that and run a website from it using Python's handy local webserver (you need Python 3.5 or above for this):

```
cd web
python3 -m http.server
```

You should then be able to go to http://localhost:8000/ to see you custom run Symbulation GUI!

1.2.3 Collecting Data with Symbulation

You are able to run Symbulation with whatever workflow you prefer to run many replicates. The following assumes that you used the Symbulation Cookiecutter to get setup.

We've provided a short script that can be used with `screen` to run several replicates and treatments.

We recommend (and have provided) a workflow where you have a `Data` folder that contains subfolders for each experiment and within each of those subfolders are:

- `simple_repeat.py`
- `SymSettings.cfg`
- Your executable file
- A `README.md` containing the date and the purpose of the experiment

Assuming that you are in the `SymbulationEmp` directory and have already compiled your `symbulation` or `symbulation_default` executable, copy your executable to your `Data` folder and change to that directory:

```
cp symbulation ../Data/sample_treatment
cd ../Data/sample_treatment
```

The `simple_repeat.py` script assumes that you already have a copy of the executable and `SymSettings.cfg` in the same directory. Within that directory, you can run `simple_repeat.py`:

```
python3 simple_repeat.py
```

By default, this will run 5 replicates of each treatment specified in `simple_repeat.py` and use the random seeds 21-25 (inclusive). You can specify the random seeds (and therefore also the number of replicates) using command line arguments, which are optional. The first command line argument is the start of the range of seeds (inclusive), and the second command line argument is the end of the range of seeds (exclusive).

For example, the input

```
python3 simple_repeat.py 10 15
```

will use seeds 10, 11, 12, 13, and 14.

1.2.4 Analyzing Data

We've also provided a basic analysis pipeline for visualizing your data. Once you have let `simple_repeat.py` run, you can change directory to the `Analysis` folder:

```
cd ../../Analysis/sample_treatment
```

and run our provided Python script:

```
python3 munge_data.py
```

These commands will output a file `munged_basic.dat` that contains the average *interaction value* of hosts and symbionts over time in each of your replicates and treatments.

You can then open the R script `SampleAnalysis.R`, set your working directory to the `Analysis` folder and run all of the lines to see a plot of the effect of vertical transmission on the evolved interaction value for hosts and symbionts.

1.2.5 Creating A New Mode

Note: this guide is only for versions greater than 0.1. If you'd like to check out this functionality sooner, it is on the `refactor` branch.

Once you have familiarized yourself with Symbulation and its default mode, you might be interested in designing your own experiment, adding functionality to the codebase, and collecting data. The existing modes include default (Host and Symbiont), efficient (Efficient Host and Efficient Symbiont), lysis (Bacterium and Phage), and public goods game or PGG (PGG Host and PGG Symbiont). There are several steps to creating your own world, including following conventions for file structure, adding your own organisms, designing tests, adding a WorldSetup file, adding targets to the makefile, and more.

- genindex
- search

1.3 ALIFE 2022 Symbulation Tutorial

We will be hosting a introduction to Symbulation tutorial at the ALIFE 2022 conferece!

1.3.1 Overview

Symbulation is an agent-based modeling platform that enables the study of symbiosis along the parasitism to mutualism spectrum and available at https://github.com/anyaevostinar/SymbulationEmp. It uses the Empirical platform and allows for a large amount of customization based on existing functionality, including allowing symbionts to live outside or inside of hosts, lysogenic and lytic life cycles, several different spatial structures, and enabling symbionts occupying the same host to engage in a public goods game. It also supports the creation of new modes and functionality for those with some C++ experience. There are 56 current configuration options, controlled by an easy-to-use configuration file and customizable data-tracking. It also includes a web-based graphical user interface that can be customized by those with C++ knowledge. Symbulation is already being used to study how spatial structure and multi-infection interact with the evolution of mutualistic symbiosis, the evolution of lysogeny, and the de novo evolution of endosymbiosis. It has also been used in classroom settings with advanced undergraduate computer science students.

This tutorial will guide new users in understanding the overall architecture of Symbulation, setting up a project (using our Cookiecutter template) to conduct an experiment using existing configuration options, customizing the browser-based graphical user interface, and creating new functionality for those comfortable with C++. The tutorial will build on the existing guides and documentation (available at https://symbulation.readthedocs.io/en/latest/) and a page for the tutorial will be created there. Attendees will also be able to shape the future directions of Symbulation by discussing new desirable features and contributing their new functionality back to the codebase if they so wish. Attendees will leave the tutorial with the start of a new research project and, ideally, lots of new ideas and potential collaborations.

All material will be hosted here, check back later for a schedule!

========== Library

1.4 Library API

1.4.1 Class Hierarchy

1.4.2 File Hierarchy

1.4.3 Full API

Namespaces

Namespace Catch

Contents

Namespaces

Classes

- *Struct MessageStream*
- *Struct NameAndTags*
- *Struct not_this_one*
- *Struct pluralise*
- *Struct RegistrarForTagAliases*
- *Struct ResultDisposition*
- *Struct ResultWas*
- *Struct RunTests*
- *Struct SectionEndInfo*
- *Struct SectionInfo*
- *Struct ShowDurations*
- *Struct SourceLineInfo*
- *Struct StreamEndStop*
- *Template Struct StringMaker*
- *Template Struct StringMaker< bool >*
- *Template Struct StringMaker< Catch::Detail::Approx >*
- *Template Struct StringMaker< char * >*
- *Template Struct StringMaker< char >*
- *Template Struct StringMaker< char const * >*
- *Template Struct StringMaker< char[SZ]>*
- *Template Struct StringMaker< double >*
- *Template Struct StringMaker< float >*
- *Template Struct StringMaker< int >*
- *Template Struct StringMaker< long >*
- *Template Struct StringMaker< long long >*
- *Template Struct StringMaker< R C::* >*
- *Template Struct StringMaker< R, typename std::enable_if< is_range< R >::value &&!::Catch::Detail::IsStreamInsertable< R >::value >::type >*
- *Template Struct StringMaker< signed char >*
- *Template Struct StringMaker< signed char[SZ]>*
- *Template Struct StringMaker< std::nullptr_t >*
- *Template Struct StringMaker< std::string >*
- *Template Struct StringMaker< std::wstring >*
- *Template Struct StringMaker< T * >*
- *Template Struct StringMaker< T[SZ]>*
- *Template Struct StringMaker< unsigned char >*

Enums

- *Enum Verbosity*

Functions

- *Function Catch::cerr*
- *Function Catch::cleanUp*
- *Function Catch::cleanUpContext*
- *Function Catch::clog*
- *Template Function Catch::compareEqual(LhsT const&, RhsT const&)*
- *Template Function Catch::compareEqual(T *const&, int)*
- *Template Function Catch::compareEqual(T *const&, long)*
- *Template Function Catch::compareEqual(int, T *const&)*
- *Template Function Catch::compareEqual(long, T *const&)*
- *Template Function Catch::compareNotEqual(LhsT const&, RhsT&&)*
- *Template Function Catch::compareNotEqual(T *const&, int)*
- *Template Function Catch::compareNotEqual(T *const&, long)*
- *Template Function Catch::compareNotEqual(int, T *const&)*
- *Template Function Catch::compareNotEqual(long, T *const&)*
- *Function Catch::contains*
- *Function Catch::cout*
- *Function Catch::endsWith(std::string const&, std::string const&)*
- *Function Catch::endsWith(std::string const&, char)*
- *Function Catch::filterTests*
- *Function Catch::formatReconstructedExpression*
- *Function Catch::getAllTestCasesSorted*
- *Function Catch::getCurrentContext*
- *Function Catch::getCurrentMutableContext*
- *Function Catch::getCurrentNanosecondsSinceEpoch*
- *Function Catch::getEstimatedClockResolution*
- *Function Catch::getMutableRegistryHub*
- *Function Catch::getRegistryHub*
- *Function Catch::getResultCapture*
- *Function Catch::handleExceptionMatchExpr(AssertionHandler&, std::string const&, StringRef const&)*
- *Function Catch::handleExceptionMatchExpr(AssertionHandler&, StringMatcher const&, StringRef const&)*
- *Function Catch::handleExpression(ITransientExpression const&)*

Typedefs

- *Typedef Catch::exceptionTranslateFunction*
- *Typedef Catch::ExceptionTranslators*
- *Typedef Catch::FunctionReturnType*
- *Typedef Catch::IConfigPtr*
- *Typedef Catch::IReporterFactoryPtr*
- *Typedef Catch::StringMatcher*

Variables

- *Variable Catch::begin*
- *Variable Catch::end*

Namespace Catch::Detail

Contents

Classes

- *Struct EnumInfo*
- *Class Approx*
- *Template Class IsStreamInsertable*

Functions

- *Template Function Catch::Detail::convertUnknownEnumToString*
- *Template Function Catch::Detail::convertUnstreamable(T const&)*
- *Template Function Catch::Detail::convertUnstreamable(T const&)*
- *Template Function Catch::Detail::convertUnstreamable(T const&)*
- *Template Function Catch::Detail::rangeToString*
- *Template Function Catch::Detail::rawMemoryToString(const T&)*
- *Function Catch::Detail::rawMemoryToString(const void *, std::size_t)*
- *Template Function Catch::Detail::stringify*

Variables

- *Variable Catch::Detail::unprintableString*

Namespace Catch::Generators

Contents

Namespaces

- *Namespace Catch::Generators::pf*

Classes

- *Template Struct as*
- *Template Struct IGenerator*
- *Template Class ChunkGenerator*
- *Template Class FilterGenerator*
- *Template Class FixedValuesGenerator*
- *Template Class Generators*
- *Class GeneratorUntypedBase*
- *Template Class GeneratorWrapper*
- *Template Class IteratorGenerator*
- *Template Class MapGenerator*
- *Template Class RandomFloatingGenerator*
- *Template Class RandomIntegerGenerator*
- *Template Class RangeGenerator*
- *Template Class RepeatGenerator*
- *Template Class SingleValueGenerator*
- *Template Class TakeGenerator*

Functions

- *Function Catch::Generators::acquireGeneratorTracker*
- *Template Function Catch::Generators::chunk*
- *Template Function Catch::Generators::filter*
- *Template Function Catch::Generators::from_range(InputIterator, InputSentinel)*
- *Template Function Catch::Generators::from_range(Container const&)*
- *Template Function Catch::Generators::generate*
- *Template Function Catch::Generators::makeGenerators(GeneratorWrapper<T>&&, Gs&&...)*
- *Template Function Catch::Generators::makeGenerators(GeneratorWrapper<T>&&)*
- *Template Function Catch::Generators::makeGenerators(T&&, Gs&&...)*
- *Template Function Catch::Generators::makeGenerators(as<T>, U&&, Gs&&...)*
- *Template Function Catch::Generators::map*
- *Template Function Catch::Generators::random(T, T)*
- *Template Function Catch::Generators::random(T, T)*
- *Template Function Catch::Generators::range(T const&, T const&, T const&)*
- *Template Function Catch::Generators::range(T const&, T const&)*
- *Template Function Catch::Generators::repeat*
- *Template Function Catch::Generators::table*
- *Template Function Catch::Generators::take*
- *Template Function Catch::Generators::value*
- *Template Function Catch::Generators::values*

Typedefs

- *Typedef Catch::Generators::GeneratorBasePtr*

Namespace Catch::Generators::pf

Contents

Functions

- *Template Function Catch::Generators::pf::make_unique*

Namespace Catch::literals

Contents

Functions

- *Function Catch::literals::operator"" _a(long double)*
- *Function Catch::literals::operator"" _a(unsigned long long)*

Namespace Catch::Matchers

Contents

Namespaces

- *Namespace Catch::Matchers::Exception*
- *Namespace Catch::Matchers::Floating*
- *Namespace Catch::Matchers::Generic*
- *Namespace Catch::Matchers::Impl*
- *Namespace Catch::Matchers::StdString*
- *Namespace Catch::Matchers::Vector*

Functions

- *Template Function Catch::Matchers::Approx*
- *Function Catch::Matchers::Contains(std::string const&, CaseSensitive::Choice)*
- *Template Function Catch::Matchers::Contains(std::vector<T> const&)*
- *Function Catch::Matchers::EndsWith*
- *Template Function Catch::Matchers::Equals(std::vector<T> const&)*
- *Function Catch::Matchers::Equals(std::string const&, CaseSensitive::Choice)*

Namespace Catch::Matchers::Exception

Contents

Classes

Namespace Catch::Matchers::Floating

Contents

Classes

Namespace Catch::Matchers::Generic

Contents

Namespaces

Classes

Namespace Catch::Matchers::Generic::Detail

Contents

Functions

Namespace Catch::Matchers::Impl

Contents

Classes

Namespace Catch::Matchers::StdString

Contents

Classes

Namespace Catch::Matchers::Vector

Contents

Classes

Namespace mpl_

Namespace std

Classes and Structs

Template Struct always_false

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public false_type`

Struct Documentation

template<typename **T**>
`struct always_false` : **`public`** false_type

Struct AssertionInfo

- Defined in file_source_catch_catch.hpp

Struct Documentation

`struct` `Catch::`**`AssertionInfo`**

Public Members

StringRef **`macroName`**

SourceLineInfo **`lineInfo`**

StringRef **`capturedExpression`**

ResultDisposition::*Flags* **`resultDisposition`**

Struct AssertionReaction

- Defined in file_source_catch_catch.hpp

Struct Documentation

`struct` `Catch::`**`AssertionReaction`**

Public Members

bool **`shouldDebugBreak`** = false

bool **`shouldThrow`** = false

Struct AutoReg

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::NonCopyable` (*Class NonCopyable*)

Struct Documentation

struct `Catch::`**AutoReg** : **public** Catch::*NonCopyable*

Public Functions

AutoReg(*ITestInvoker* *invoker, *SourceLineInfo* **const** &lineInfo, *StringRef* **const** &classOrMethod, *NameAndTags* **const** &nameAndTags) **noexcept**

~AutoReg()

Struct CaseSensitive

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct `Catch::`**CaseSensitive**

Public Types

enum Choice

Values:

enumerator Yes

enumerator No

Struct Counts

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**Counts**

Public Functions

Counts **operator-**(*Counts* **const** &*other*) **const**

Counts &**operator+=**(*Counts* **const** &*other*)

std::size_t **total**() **const**

bool **allPassed**() **const**

bool **allOk**() **const**

Public Members

std::size_t **passed** = 0

std::size_t **failed** = 0

std::size_t **failedButOk** = 0

Struct Decomposer

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**Decomposer**

Public Functions

template<typename **T**>
auto **operator<=**(*T* **const** &*lhs*) -> *ExprLhs*<*T* **const**&>

auto **operator<=**(bool *value*) -> *ExprLhs*<bool>

Struct EnumInfo

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::Detail::**EnumInfo**

Public Functions

~EnumInfo()

StringRef **lookup**(int *value*) **const**

Public Members

StringRef **m_name**

std::vector<std::pair<int, *StringRef*>> **m_values**

Template Struct as

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **T**>
struct as

Template Struct IGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Generators::GeneratorUntypedBase (*Class GeneratorUntypedBase*)

Derived Types

- public Catch::Generators::FilterGenerator< T, Predicate > (*Template Class FilterGenerator*)
- public Catch::Generators::FixedValuesGenerator< T > (*Template Class FixedValuesGenerator*)
- public Catch::Generators::Generators< T > (*Template Class Generators*)
- public Catch::Generators::IteratorGenerator< T > (*Template Class IteratorGenerator*)
- public Catch::Generators::MapGenerator< T, U, Func > (*Template Class MapGenerator*)
- public Catch::Generators::RangeGenerator< T > (*Template Class RangeGenerator*)

- `public Catch::Generators::RepeatGenerator< T >` (*Template Class RepeatGenerator*)
- `public Catch::Generators::SingleValueGenerator< T >` (*Template Class SingleValueGenerator*)
- `public Catch::Generators::TakeGenerator< T >` (*Template Class TakeGenerator*)

Struct Documentation

template<typename **T**>
struct `Catch::Generators::`**IGenerator** : **public** Catch::Generators::*GeneratorUntypedBase*
Subclassed by *Catch::Generators::FilterGenerator< T, Predicate >*, *Catch::Generators::FixedValuesGenerator< T >*, *Catch::Generators::Generators< T >*, *Catch::Generators::IteratorGenerator< T >*, *Catch::Generators::MapGenerator< T, U, Func >*, *Catch::Generators::RangeGenerator< T >*, *Catch::Generators::RepeatGenerator< T >*, *Catch::Generators::SingleValueGenerator< T >*, *Catch::Generators::TakeGenerator< T >*

Public Types

using type = *T*

Public Functions

~IGenerator() = default

T **const** &**get**() **const** = 0

Struct IConfig

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::NonCopyable` (*Class NonCopyable*)

Struct Documentation

struct `Catch::`**IConfig** : **public** Catch::*NonCopyable*

Public Functions

~IConfig()

bool **allowThrows**() **const** = 0

std::ostream &**stream**() **const** = 0

std::string **name**() **const** = 0

bool **includeSuccessfulResults**() **const** = 0

bool **shouldDebugBreak**() **const** = 0

bool **warnAboutMissingAssertions**() **const** = 0

bool **warnAboutNoTests**() **const** = 0

int **abortAfter**() **const** = 0

bool **showInvisibles**() **const** = 0

ShowDurations::*OrNot* **showDurations**() **const** = 0

TestSpec **const** &**testSpec**() **const** = 0

bool **hasTestFilters**() **const** = 0

std::vector<std::string> **const** &**getTestsOrTags**() **const** = 0

RunTests::*InWhatOrder* **runOrder**() **const** = 0

unsigned int **rngSeed**() **const** = 0

UseColour::*YesOrNo* **useColour**() **const** = 0

std::vector<std::string> **const** &**getSectionsToRun**() **const** = 0

Verbosity **verbosity**() **const** = 0

bool **benchmarkNoAnalysis**() **const** = 0

int **benchmarkSamples**() **const** = 0

double **benchmarkConfidenceInterval**() **const** = 0

unsigned int **benchmarkResamples**() **const** = 0

std::chrono::milliseconds **benchmarkWarmupTime**() **const** = 0

Struct IContext

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Type

- `public Catch::IMutableContext` (*Struct IMutableContext*)

Struct Documentation

struct `Catch::IContext`
Subclassed by *Catch::IMutableContext*

Public Functions

~IContext()

IResultCapture ***getResultCapture**() = 0

IRunner ***getRunner**() = 0

IConfigPtr **const** &**getConfig**() **const** = 0

Struct IExceptionTranslator

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct `Catch::IExceptionTranslator`

Public Functions

~IExceptionTranslator()

std::string **translate**(*ExceptionTranslators*::const_iterator *it*, *ExceptionTranslators*::const_iterator *itEnd*) **const** = 0

Struct IExceptionTranslatorRegistry

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IExceptionTranslatorRegistry**

Public Functions

~IExceptionTranslatorRegistry()

std::string **translateActiveException**() **const** = 0

Struct IGeneratorTracker

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IGeneratorTracker**

Public Functions

~IGeneratorTracker()

auto **hasGenerator**() **const** -> bool = 0

auto **getGenerator**() **const** -> Generators::*GeneratorBasePtr* **const**& = 0

void **setGenerator**(Generators::*GeneratorBasePtr* &&*generator*) = 0

Struct IMutableContext

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::IContext (*Struct IContext*)

Struct Documentation

struct Catch::**IMutableContext** : **public** Catch::*IContext*

Public Functions

~IMutableContext()

void **setResultCapture**(*IResultCapture* *resultCapture) = 0

void **setRunner**(*IRunner* *runner) = 0

void **setConfig**(*IConfigPtr* **const** &config) = 0

Struct IMutableEnumValuesRegistry

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IMutableEnumValuesRegistry**

Public Functions

~IMutableEnumValuesRegistry()

Detail::*EnumInfo* **const** &**registerEnum**(*StringRef* *enumName*, *StringRef* *allEnums*, std::vector<int> **const** &*values*) = 0

template<typename **E**>
Detail::*EnumInfo* **const** &**registerEnum**(*StringRef* *enumName*, *StringRef* *allEnums*, std::initializer_list<*E*> *values*)

Struct IMutableRegistryHub

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IMutableRegistryHub**

Public Functions

~IMutableRegistryHub()

void **registerReporter**(std::string **const** &*name*, *IReporterFactoryPtr* **const** &*factory*) = 0

void **registerListener**(*IReporterFactoryPtr* **const** &*factory*) = 0

void **registerTest**(*TestCase* **const** &*testInfo*) = 0

void **registerTranslator**(**const** *IExceptionTranslator* *translator) = 0

void **registerTagAlias**(std::string **const** &*alias*, std::string **const** &*tag*, *SourceLineInfo* **const** &*lineInfo*) = 0

void **registerStartupException**() **noexcept** = 0

IMutableEnumValuesRegistry &**getMutableEnumValuesRegistry**() = 0

Struct IRegistryHub

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IRegistryHub**

Public Functions

~IRegistryHub()

IReporterRegistry **const** &**getReporterRegistry**() **const** = 0

ITestCaseRegistry **const** &**getTestCaseRegistry**() **const** = 0

ITagAliasRegistry **const** &**getTagAliasRegistry**() **const** = 0

IExceptionTranslatorRegistry **const** &**getExceptionTranslatorRegistry**() **const** = 0

StartupExceptionRegistry **const** &**getStartupExceptionRegistry**() **const** = 0

Struct IResultCapture

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IResultCapture**

Public Functions

~IResultCapture()

bool **sectionStarted**(*SectionInfo* **const** &*sectionInfo*, *Counts* &*assertions*) = 0

void **sectionEnded**(*SectionEndInfo* **const** &*endInfo*) = 0

void **sectionEndedEarly**(*SectionEndInfo* **const** &*endInfo*) = 0

auto **acquireGeneratorTracker**(*SourceLineInfo* **const** &*lineInfo*) -> *IGeneratorTracker*& = 0

void **pushScopedMessage**(*MessageInfo* **const** &*message*) = 0

void **popScopedMessage**(*MessageInfo* **const** &*message*) = 0

void **emplaceUnscopedMessage**(*MessageBuilder* **const** &*builder*) = 0

void **handleFatalErrorCondition**(*StringRef* *message*) = 0

void **handleExpr**(*AssertionInfo* **const** &*info*, *ITransientExpression* **const** &*expr*, *AssertionReaction* &*reaction*) = 0

void **handleMessage**(*AssertionInfo* **const** &*info*, *ResultWas*::*OfType* *resultType*, *StringRef* **const** &*message*, *AssertionReaction* &*reaction*) = 0

void **handleUnexpectedExceptionNotThrown**(*AssertionInfo* **const** &*info*, *AssertionReaction* &*reaction*) = 0

void **handleUnexpectedInflightException**(*AssertionInfo* **const** &*info*, std::string **const** &*message*, *AssertionReaction* &*reaction*) = 0

void **handleIncomplete**(*AssertionInfo* **const** &*info*) = 0

void **handleNonExpr**(*AssertionInfo* **const** &*info*, *ResultWas*::*OfType* *resultType*, *AssertionReaction* &*reaction*) = 0

bool **lastAssertionPassed**() = 0

void **assertionPassed**() = 0

std::string **getCurrentTestName**() **const** = 0

const AssertionResult ***getLastResult**() **const** = 0

void **exceptionEarlyReported**() = 0

Struct IRunner

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IRunner**

Public Functions

~IRunner()

bool **aborting**() **const** = 0

Template Struct is_callable

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **T**>
struct is_callable

Template Struct is_callable< Fun(Args...)>

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public decltypeis_callable_tester::test< Fun, Args... >`

Struct Documentation

template<typename **Fun**, typename ...**Args**>
struct is_callable<*Fun* (*Args*...) > : **public** decltypeis_callable_tester::test<*Fun*, *Args*...>

Struct is_callable_tester

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**is_callable_tester**

Public Static Functions

template<typename Fun, typename... Args> true_given< decltype(std::declval< Fun >)(std

template<typename...>
std::false_type **test**(...)

Template Struct is_range

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **T**>
struct Catch::**is_range**

Public Static Attributes

const bool **value** = !std::is_same<decltype(*begin*(std::declval<*T*>())), *not_this_one*>::value && !std::is_same<decltype(*end*(s

Struct IStream

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**IStream**

Public Functions

~IStream()

std::ostream &**stream**() **const** = 0

Struct ITestCaseRegistry

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**ITestCaseRegistry**

Public Functions

~ITestCaseRegistry()

std::vector<*TestCase*> **const** &**getAllTests**() **const** = 0

std::vector<*TestCase*> **const** &**getAllTestsSorted**(*IConfig* **const** &*config*) **const** = 0

Struct ITestInvoker

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Type

- public Catch::TestInvokerAsMethod< C > (*Template Class TestInvokerAsMethod*)

Struct Documentation

struct Catch::**ITestInvoker**

Subclassed by *Catch::TestInvokerAsMethod< C >*

Public Functions

void **invoke**() **const** = 0

~ITestInvoker()

Struct ITransientExpression

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Types

- public Catch::BinaryExpr< LhsT, RhsT > (*Template Class BinaryExpr*)
- public Catch::MatchExpr< ArgT, MatcherT > (*Template Class MatchExpr*)
- public Catch::UnaryExpr< LhsT > (*Template Class UnaryExpr*)

Struct Documentation

struct Catch::**ITransientExpression**

Subclassed by *Catch::BinaryExpr< LhsT, RhsT >*, *Catch::MatchExpr< ArgT, MatcherT >*, *Catch::UnaryExpr< LhsT >*

Public Functions

auto **isBinaryExpression**() **const** -> bool

auto **getResult**() **const** -> bool

void **streamReconstructedExpression**(std::ostream &*os*) **const** = 0

ITransientExpression(bool *isBinaryExpression*, bool *result*)

~ITransientExpression()

Public Members

bool **m_isBinaryExpression**

bool **m_result**

Struct WithinAbsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< double >` (*Template Struct MatcherBase*)

Struct Documentation

struct `Catch::Matchers::Floating::`**WithinAbsMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<double>

Public Functions

WithinAbsMatcher(double *target*, double *margin*)

bool **match**(double **const** &*matchee*) **const override**

std::string **describe**() **const override**

Struct WithinRelMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< double >` (*Template Struct MatcherBase*)

Struct Documentation

struct `Catch::Matchers::Floating::`**WithinRelMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<double>

Public Functions

WithinRelMatcher(double *target*, double *epsilon*)

bool **match**(double **const** &*matchee*) **const override**

std::string **describe**() **const override**

Struct WithinUlpsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< double >` (*Template Struct MatcherBase*)

Struct Documentation

struct `Catch::Matchers::Floating::`**WithinUlpsMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<double>

Public Functions

WithinUlpsMatcher(double *target*, uint64_t *ulps*, FloatingPointKind *baseType*)

bool **match**(double **const** &*matchee*) **const override**

std::string **describe**() **const override**

Template Struct MatchAllOf

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< ArgT >` (*Template Struct MatcherBase*)

Struct Documentation

template<typename **ArgT**>
struct Catch::Matchers::Impl::**MatchAllOf** : **public** Catch::Matchers::Impl::*MatcherBase*<*ArgT*>

Public Functions

bool **match**(*ArgT* **const** &*arg*) **const override**

std::string **describe**() **const override**

MatchAllOf<*ArgT*> **operator&&**(*MatcherBase*<*ArgT*> **const** &*other*)

Public Members

std::vector<*MatcherBase*<*ArgT*> **const***> **m_matchers**

Template Struct MatchAnyOf

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Matchers::Impl::MatcherBase< ArgT >(*Template Struct MatcherBase*)

Struct Documentation

template<typename **ArgT**>
struct Catch::Matchers::Impl::**MatchAnyOf** : **public** Catch::Matchers::Impl::*MatcherBase*<*ArgT*>

Public Functions

bool **match**(*ArgT* **const** &*arg*) **const override**

std::string **describe**() **const override**

MatchAnyOf<*ArgT*> **operator||**(*MatcherBase*<*ArgT*> **const** &*other*)

Public Members

std::vector<*MatcherBase*<*ArgT*> **const***> **m_matchers**

Template Struct MatcherBase

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Types

- `public Catch::Matchers::Impl::MatcherUntypedBase` (*Class MatcherUntypedBase*)
- `public Catch::Matchers::Impl::MatcherMethod< T >` (*Template Struct MatcherMethod*)

Derived Types

- `public Catch::Matchers::Exception::ExceptionMessageMatcher` (*Class ExceptionMessageMatcher*)
- `public Catch::Matchers::Floating::WithinAbsMatcher` (*Struct WithinAbsMatcher*)
- `public Catch::Matchers::Floating::WithinRelMatcher` (*Struct WithinRelMatcher*)
- `public Catch::Matchers::Floating::WithinUlpsMatcher` (*Struct WithinUlpsMatcher*)
- `public Catch::Matchers::Generic::PredicateMatcher< T >` (*Template Class PredicateMatcher*)
- `public Catch::Matchers::StdString::RegexMatcher` (*Struct RegexMatcher*)
- `public Catch::Matchers::StdString::StringMatcherBase` (*Struct StringMatcherBase*)
- `public Catch::Matchers::Vector::ApproxMatcher< T >` (*Template Struct ApproxMatcher*)
- `public Catch::Matchers::Vector::ContainsElementMatcher< T >` (*Template Struct ContainsElementMatcher*)
- `public Catch::Matchers::Vector::ContainsMatcher< T >` (*Template Struct ContainsMatcher*)
- `public Catch::Matchers::Vector::EqualsMatcher< T >` (*Template Struct EqualsMatcher*)
- `public Catch::Matchers::Vector::UnorderedEqualsMatcher< T >` (*Template Struct UnorderedEqualsMatcher*)

Struct Documentation

template<typename **T**>
struct Catch::Matchers::Impl::**MatcherBase** : **public** Catch::Matchers::Impl::*MatcherUntypedBase*, **public** Catch
Subclassed by *Catch::Matchers::Exception::ExceptionMessageMatcher*, *Catch::Matchers::Floating::WithinAbsMatcher*, *Catch::Matchers::Floating::WithinRelMatcher*, *Catch::Matchers::Floating::WithinUlpsMatcher*, *Catch::Matchers::Generic::PredicateMatcher< T >*, *Catch::Matchers::StdString::RegexMatcher*, *Catch::Matchers::StdString::StringMatcherBase*, *Catch::Matchers::Vector::ApproxMatcher< T >*, *Catch::Matchers::Vector::ContainsElementMatcher< T >*, *Catch::Matchers::Vector::ContainsMatcher< T >*, *Catch::Matchers::Vector::EqualsMatcher< T >*, *Catch::Matchers::Vector::UnorderedEqualsMatcher< T >*

Public Functions

MatchAllOf<*T*> **operator&&**(*MatcherBase* **const** &*other*) **const**

MatchAnyOf<*T*> **operator||**(*MatcherBase* **const** &*other*) **const**

MatchNotOf<*T*> **operator!**() **const**

Template Struct MatcherMethod

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **ObjectT**>
struct Catch::Matchers::Impl::**MatcherMethod**

Public Functions

bool **match**(*ObjectT* **const** &*arg*) **const** = 0

Template Struct MatchNotOf

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Matchers::Impl::MatcherBase< ArgT > (*Template Struct MatcherBase*)

Struct Documentation

template<typename **ArgT**>
struct Catch::Matchers::Impl::**MatchNotOf** : **public** Catch::Matchers::Impl::*MatcherBase*<*ArgT*>

Public Functions

MatchNotOf(*MatcherBase*<*ArgT*> **const** &*underlyingMatcher*)

bool **match**(*ArgT* **const** &*arg*) **const override**

std::string **describe**() **const override**

Public Members

MatcherBase<*ArgT*> **const** &**m_underlyingMatcher**

Struct CasedString

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::Matchers::StdString::**CasedString**

Public Functions

CasedString(std::string **const** &*str*, *CaseSensitive*::*Choice* *caseSensitivity*)

std::string **adjustString**(std::string **const** &*str*) **const**

std::string **caseSensitivitySuffix**() **const**

Public Members

CaseSensitive::*Choice* **m_caseSensitivity**

std::string **m_str**

Struct ContainsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::StdString::StringMatcherBase` (*Struct StringMatcherBase*)

Struct Documentation

struct `Catch::Matchers::StdString::`**ContainsMatcher** : **public** Catch::Matchers::StdString::*StringMatcherBase*

Public Functions

ContainsMatcher(*CasedString* **const** &*comparator*)

bool **match**(std::string **const** &*source*) **const override**

Struct EndsWithMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::StdString::StringMatcherBase` (*Struct StringMatcherBase*)

Struct Documentation

struct `Catch::Matchers::StdString::`**EndsWithMatcher** : **public** Catch::Matchers::StdString::*StringMatcherBase*

Public Functions

EndsWithMatcher(*CasedString* **const** &*comparator*)

bool **match**(std::string **const** &*source*) **const override**

Struct EqualsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::StdString::StringMatcherBase` (*Struct StringMatcherBase*)

Struct Documentation

struct `Catch::Matchers::StdString::`**EqualsMatcher** : **public** Catch::Matchers::StdString::*StringMatcherBase*

Public Functions

EqualsMatcher(*CasedString* **const** &*comparator*)

bool **match**(std::string **const** &*source*) **const override**

Struct RegexMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< std::string >` (*Template Struct MatcherBase*)

Struct Documentation

struct `Catch::Matchers::StdString::`**RegexMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<std::string>

Public Functions

RegexMatcher(std::string *regex*, *CaseSensitive*::*Choice caseSensitivity*)

bool **match**(std::string **const** &*matchee*) **const override**

std::string **describe**() **const override**

Struct StartsWithMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::StdString::StringMatcherBase` (*Struct StringMatcherBase*)

Struct Documentation

struct `Catch::Matchers::StdString::`**StartsWithMatcher** : **public** Catch::Matchers::StdString::*StringMatcherBa*

Public Functions

StartsWithMatcher(*CasedString* **const** &*comparator*)

bool **match**(std::string **const** &*source*) **const override**

Struct StringMatcherBase

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< std::string >` (*Template Struct MatcherBase*)

Derived Types

- `public Catch::Matchers::StdString::ContainsMatcher` (*Struct ContainsMatcher*)
- `public Catch::Matchers::StdString::EndsWithMatcher` (*Struct EndsWithMatcher*)
- `public Catch::Matchers::StdString::EqualsMatcher` (*Struct EqualsMatcher*)
- `public Catch::Matchers::StdString::StartsWithMatcher` (*Struct StartsWithMatcher*)

Struct Documentation

struct Catch::Matchers::StdString::**StringMatcherBase** : **public** Catch::Matchers::Impl::*MatcherBase*<std::stri
Subclassed by *Catch::Matchers::StdString::ContainsMatcher*, *Catch::Matchers::StdString::EndsWithMatcher*, *Catch::Matchers::StdString::EqualsMatcher*, *Catch::Matchers::StdString::StartsWithMatcher*

Public Functions

StringMatcherBase(std::string **const** &*operation*, *CasedString* **const** &*comparator*)

std::string **describe**() **const override**

Public Members

CasedString **m_comparator**

std::string **m_operation**

Template Struct ApproxMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Matchers::Impl::MatcherBase< std::vector< T > > (*Template Struct MatcherBase*)

Struct Documentation

template<typename **T**>
struct Catch::Matchers::Vector::**ApproxMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<std::vector<*T*>>

Public Functions

ApproxMatcher(std::vector<*T*> **const** &*comparator*)

bool **match**(std::vector<*T*> **const** &*v*) **const override**

std::string **describe**() **const override**

template<typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
ApproxMatcher &**epsilon**(*T* **const** &*newEpsilon*)

template<typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
ApproxMatcher &**margin**(*T* **const** &*newMargin*)

template<typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
ApproxMatcher &**scale**(*T* **const** &*newScale*)

Public Members

std::vector<*T*> **const** &**m_comparator**

Catch::Detail::*Approx* **approx** = Catch::Detail::*Approx*::*custom*()

Template Struct ContainsElementMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< std::vector< T > >` (*Template Struct MatcherBase*)

Struct Documentation

template<typename **T**>
struct `Catch::Matchers::Vector::`**ContainsElementMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<std::

Public Functions

ContainsElementMatcher(*T* **const** &*comparator*)

bool **match**(std::vector<*T*> **const** &*v*) **const override**

std::string **describe**() **const override**

Public Members

T **const** &**m_comparator**

Template Struct ContainsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< std::vector< T > >` (*Template Struct MatcherBase*)

Struct Documentation

template<typename **T**>
struct Catch::Matchers::Vector::**ContainsMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<std::vector<*T*>

Public Functions

ContainsMatcher(std::vector<*T*> **const** &*comparator*)

bool **match**(std::vector<*T*> **const** &*v*) **const override**

std::string **describe**() **const override**

Public Members

std::vector<*T*> **const** &**m_comparator**

Template Struct EqualsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Matchers::Impl::MatcherBase< std::vector< T > > (*Template Struct MatcherBase*)

Struct Documentation

template<typename **T**>
struct Catch::Matchers::Vector::**EqualsMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<std::vector<*T*>>

Public Functions

EqualsMatcher(std::vector<*T*> **const** &*comparator*)

bool **match**(std::vector<*T*> **const** &*v*) **const override**

std::string **describe**() **const override**

Public Members

std::vector<*T*> **const** &**m_comparator**

Template Struct UnorderedEqualsMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< std::vector< T > >` (*Template Struct MatcherBase*)

Struct Documentation

template<typename **T**>
struct `Catch::Matchers::Vector::`**UnorderedEqualsMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<std::

Public Functions

UnorderedEqualsMatcher(std::vector<*T*> **const** &*target*)

bool **match**(std::vector<*T*> **const** &*vec*) **const override**

std::string **describe**() **const override**

Struct MessageBuilder

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::MessageStream` (*Struct MessageStream*)

Struct Documentation

struct `Catch::`**MessageBuilder** : **public** Catch::*MessageStream*

Public Functions

MessageBuilder(*StringRef* **const** &*macroName*, *SourceLineInfo* **const** &*lineInfo*, *ResultWas*::*OfType* *type*)

template<typename **T**>
MessageBuilder &**operator<<**(*T* **const** &*value*)

Public Members

MessageInfo **m_info**

Struct MessageInfo

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**MessageInfo**

Public Functions

MessageInfo(*StringRef* **const** &*_macroName*, *SourceLineInfo* **const** &*_lineInfo*, *ResultWas*::*OfType* *_type*)

bool **operator==**(*MessageInfo* **const** &*other*) **const**

bool **operator<**(*MessageInfo* **const** &*other*) **const**

Public Members

StringRef **macroName**

std::string **message**

SourceLineInfo **lineInfo**

ResultWas::*OfType* **type**

unsigned int **sequence**

Struct MessageStream

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Type

- `public Catch::MessageBuilder` (*Struct MessageBuilder*)

Struct Documentation

struct `Catch::`**MessageStream**
Subclassed by *Catch::MessageBuilder*

Public Functions

template<typename **T**>
MessageStream &**operator<<**(*T* **const** &*value*)

Public Members

ReusableStringStream **m_stream**

Struct NameAndTags

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct `Catch::`**NameAndTags**

Public Functions

NameAndTags(*StringRef* **const** &*name_* = *StringRef*(), *StringRef* **const** &*tags_* = *StringRef*())
noexcept

Public Members

StringRef **name**

StringRef **tags**

Struct not_this_one

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct not_this_one

Struct pluralise

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**pluralise**

Public Functions

pluralise(std::size_t *count*, std::string **const** &*label*)

Public Members

std::size_t **m_count**

std::string **m_label**

Friends

friend std::ostream &**operator<<**(std::ostream &*os*, *pluralise* **const** &*pluraliser*)

Struct RegistrarForTagAliases

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**RegistrarForTagAliases**

Public Functions

RegistrarForTagAliases(char **const** *alias, char **const** *tag, *SourceLineInfo* **const** &*lineInfo*)

Struct ResultDisposition

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**ResultDisposition**

Public Types

enum Flags

Values:

enumerator Normal

enumerator ContinueOnFailure

enumerator FalseTest

enumerator SuppressFail

Struct ResultWas

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**ResultWas**

Public Types

enum OfType

Values:

enumerator Unknown

enumerator Ok

enumerator Info

enumerator Warning

enumerator FailureBit

enumerator ExpressionFailed

enumerator ExplicitFailure

enumerator Exception

enumerator ThrewException

enumerator DidntThrowException

enumerator FatalErrorCondition

Struct RunTests

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**RunTests**

Public Types

enum InWhatOrder

Values:

enumerator InDeclarationOrder

enumerator InLexicographicalOrder

enumerator InRandomOrder

Struct SectionEndInfo

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**SectionEndInfo**

Public Members

SectionInfo **sectionInfo**

Counts **prevAssertions**

double **durationInSeconds**

Struct SectionInfo

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**SectionInfo**

Public Functions

SectionInfo(*SourceLineInfo* **const** &*_lineInfo*, std::string **const** &*_name*)

SectionInfo(*SourceLineInfo* **const** &*_lineInfo*, std::string **const** &*_name*, std::string **const**&)

Public Members

std::string **name**

std::string **description**

SourceLineInfo **lineInfo**

Struct ShowDurations

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**ShowDurations**

Public Types

enum OrNot

Values:

enumerator DefaultForReporter

enumerator Always

enumerator Never

Struct SourceLineInfo

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**SourceLineInfo**

Public Functions

SourceLineInfo() = delete

SourceLineInfo(char **const** *_file, std::size_t _line) **noexcept**

SourceLineInfo(*SourceLineInfo* **const** &other) = default

SourceLineInfo &**operator=**(*SourceLineInfo* **const**&) = default

SourceLineInfo(*SourceLineInfo*&&) **noexcept** = default

SourceLineInfo &**operator=**(*SourceLineInfo*&&) **noexcept** = default

bool **empty**() **const noexcept**

bool **operator==**(*SourceLineInfo* **const** &other) **const noexcept**

bool **operator<**(*SourceLineInfo* **const** &other) **const noexcept**

Public Members

char **const** ***file**

std::size_t **line**

Struct StreamEndStop

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**StreamEndStop**

Public Functions

std::string **operator+**() **const**

Template Struct StringMaker

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **T**, typename = void>
struct Catch::**StringMaker**

Public Static Functions

template<typename **Fake** = *T*>
std::enable_if<::Catch::Detail::*IsStreamInsertable*<*Fake*>::value, std::string>::type **convert**(**const** *Fake* &*value*)

template<typename **Fake** = *T*>
std::enable_if<!::Catch::Detail::*IsStreamInsertable*<*Fake*>::value, std::string>::type **convert**(**const** *Fake* &*value*)

Template Struct StringMaker< bool >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<bool>

Public Static Functions

std::string **convert**(bool *b*)

Template Struct StringMaker< Catch::Detail::Approx >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<Catch::Detail::*Approx*>

Public Static Functions

std::string **convert**(Catch::Detail::*Approx* **const** &*value*)

Template Struct StringMaker< char * >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<char*>

Public Static Functions

std::string **convert**(char **str*)

Template Struct StringMaker< char >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<char>

Public Static Functions

std::string **convert**(char *c*)

Template Struct StringMaker< char const * >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<char **const***>

Public Static Functions

std::string **convert**(char **const** *str)

Template Struct StringMaker< char[SZ]>

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<int **SZ**>
struct Catch::**StringMaker**<char[SZ]>

Public Static Functions

std::string **convert**(char **const** **str*)

Template Struct StringMaker< double >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<double>

Public Static Functions

std::string **convert**(double *value*)

Public Static Attributes

int **precision**

Template Struct StringMaker< float >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<float>

Public Static Functions

std::string **convert**(float *value*)

Public Static Attributes

int **precision**

Template Struct StringMaker< int >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<int>

Public Static Functions

std::string **convert** (int *value*)

Template Struct StringMaker< long >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<long>

Public Static Functions

std::string **convert** (long *value*)

Template Struct StringMaker< long long >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<long long>

Public Static Functions

std::string **convert** (long long *value*)

Template Struct StringMaker< R C::* >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **R**, typename **C**>
struct Catch::**StringMaker**<*R* C::*>

Public Static Functions

std::string **convert** (*R C*::* *p*)

Template Struct StringMaker< R, typename std::enable_if< is_range< R >::value &&!::Catch::Detail::IsStreamInsertable< R >::value >::type >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **R**>
struct Catch::**StringMaker**<*R*, **typename** std::enable_if<*is_range*<*R*>::value && !::Catch::Detail::*IsStreamInsertable*<*R*>::

Public Static Functions

std::string **convert** (*R* **const** &*range*)

Template Struct StringMaker< signed char >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<signed char>

Public Static Functions

std::string **convert** (signed char *c*)

Template Struct StringMaker< signed char[SZ]>

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<int **SZ**>
struct Catch::**StringMaker**<signed char[SZ]>

Public Static Functions

std::string **convert** (signed char **const** **str*)

Template Struct StringMaker< std::nullptr_t >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<std::nullptr_t>

Public Static Functions

std::string **convert** (std::nullptr_t)

Template Struct StringMaker< std::string >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<std::string>

Public Static Functions

std::string **convert**(**const** std::string &*str*)

Template Struct StringMaker< std::wstring >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<std::wstring>

Public Static Functions

std::string **convert**(**const** std::wstring &*wstr*)

Template Struct StringMaker< T * >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **T**>
struct Catch::**StringMaker**<*T**>

Public Static Functions

template<typename **U**>
std::string **convert**(*U* **p*)

Template Struct StringMaker< T[SZ]>

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<typename **T**, int **SZ**>
struct Catch::**StringMaker**<*T*[SZ]>

Public Static Functions

std::string **convert** (*T* **const** (&*arr*)[SZ])

Template Struct StringMaker< unsigned char >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<unsigned char>

Public Static Functions

std::string **convert** (unsigned char *c*)

Template Struct StringMaker< unsigned char[SZ]>

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<int **SZ**>
struct Catch::**StringMaker**<unsigned char[SZ]>

Public Static Functions

std::string **convert** (unsigned char **const** **str*)

Template Struct StringMaker< unsigned int >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<unsigned int>

Public Static Functions

std::string **convert** (unsigned int *value*)

Template Struct StringMaker< unsigned long >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<unsigned long>

Public Static Functions

std::string **convert** (unsigned long *value*)

Template Struct StringMaker< unsigned long long >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<unsigned long long>

Public Static Functions

std::string **convert** (unsigned long long *value*)

Template Struct StringMaker< wchar_t * >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<wchar_t*>

Public Static Functions

std::string **convert**(wchar_t *_str_)

Template Struct StringMaker< wchar_t const * >

- Defined in file_source_catch_catch.hpp

Struct Documentation

template<>
struct Catch::**StringMaker**<wchar_t **const***>

Public Static Functions

std::string **convert**(wchar_t **const** *_str_)

Struct TestCaseInfo

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Type

- `public Catch::TestCase` (*Class TestCase*)

Struct Documentation

struct Catch::**TestCaseInfo**
Subclassed by *Catch::TestCase*

Public Types

enum SpecialProperties
Values:

enumerator None

enumerator IsHidden

enumerator ShouldFail

enumerator MayFail

enumerator Throws

enumerator NonPortable

enumerator Benchmark

Public Functions

TestCaseInfo(std::string **const** &*_name*, std::string **const** &*_className*, std::string **const** &*_description*, std::vector<std::string> **const** &*_tags*, *SourceLineInfo* **const** &*_lineInfo*)

bool **isHidden**() **const**

bool **throws**() **const**

bool **okToFail**() **const**

bool **expectedToFail**() **const**

std::string **tagsAsString**() **const**

Public Members

std::string **name**

std::string **className**

std::string **description**

std::vector<std::string> **tags**

std::vector<std::string> **lcaseTags**

SourceLineInfo **lineInfo**

SpecialProperties **properties**

Friends

friend void **setTags**(*TestCaseInfo* &*testCaseInfo*, std::vector<std::string> *tags*)

Struct TestFailureException

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct TestFailureException

Struct Totals

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**Totals**

Public Functions

Totals **operator-**(*Totals* **const** &*other*) **const**

Totals &**operator+=**(*Totals* **const** &*other*)

Totals **delta**(*Totals* **const** &*prevTotals*) **const**

Public Members

int **error** = 0

Counts **assertions**

Counts **testCases**

Template Struct true_given

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public true_type

Struct Documentation

template<typename>
struct true_given : **public** true_type

Struct UseColour

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**UseColour**

Public Types

enum YesOrNo
Values:

enumerator Auto

enumerator Yes

enumerator No

Struct WaitForKeypress

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**WaitForKeypress**

Public Types

enum When
Values:

enumerator Never

enumerator BeforeStart

enumerator BeforeExit

enumerator BeforeStartAndExit

Struct WarnAbout

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch::**WarnAbout**

Public Types

enum What
Values:

enumerator Nothing

enumerator NoAssertions

enumerator NoTests

Struct Catch_global_namespace_dummy

- Defined in file_source_catch_catch.hpp

Struct Documentation

struct Catch_global_namespace_dummy

Class Bacterium

- Defined in file_source_lysis_mode_Bacterium.h

Inheritance Relationships

Base Type

- `public Host` (*Class Host*)

Class Documentation

class Bacterium : **public** *Host*

Public Functions

Bacterium(emp::Ptr<emp::Random> *_random*, emp::Ptr<*LysisWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, emp::vector<emp::Ptr<*Organism*>> *_syms* = {}, emp::vector<emp::Ptr<*Organism*>> *_repro_syms* = {}, std::set<int> *_set* = std::set<int>(), double *_points* = 0.0)

The constructor for the bacterium class

Bacterium(**const** *Bacterium*&) = default

Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

Bacterium(*Bacterium*&&) = default

Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

Bacterium() = default

Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

double **GetIncVal**()
Input: None

Output: The double representing a genome's value.

Purpose: To determine a genome's value.

void **SetIncVal**(double *_in*)
Input: The double to be set as the bacterium's genome value

Output: None

Purpose: To set a bacterium's genome value

emp::Ptr<*Organism*> **makeNew**()
Input: None.

Output: A new bacterium with same properties as this bacterium.

Purpose: To avoid creating an organism via constructor in other methods.

void **mutate**()
Input: None

Output: None

Purpose: To mutate a bacterium's genome. The mutation will be based on a value chosen from a normal distribution centered at 0, with a standard deviation that is equal to the mutation size. *Bacterium* mutation can be turned on or off.

double **ProcessLysogenResources**(double *phage_inc_val*)

Protected Attributes

double **host_incorporation_val** = 0
Purpose: Represents the host's genome. A double with a range from 0 to 1. The host's genome gets compared against the phage's incorporation value.

emp::Ptr<*LysisWorld*> **my_world** = NULL
Purpose: Represents the world that the hosts are living in.

Class AssertionHandler

- Defined in file_source_catch_catch.hpp

Class Documentation

class Catch::**AssertionHandler**

Public Functions

AssertionHandler(*StringRef* **const** &*macroName*, *SourceLineInfo* **const** &*lineInfo*, *StringRef* *capturedExpression*, *ResultDisposition*::*Flags* *resultDisposition*)

~AssertionHandler()

template<typename **T**>
void **handleExpr**(*ExprLhs*<*T*> **const** &*expr*)

void **handleExpr**(*ITransientExpression* **const** &*expr*)

void **handleMessage**(*ResultWas*::*OfType* *resultType*, *StringRef* **const** &*message*)

void **handleExceptionThrownAsExpected**()

void **handleUnexpectedExceptionNotThrown**()

void **handleExceptionNotThrownAsExpected**()

void **handleThrowingCallSkipped**()

void **handleUnexpectedInflightException**()

void **complete**()

void **setCompleted**()

auto **allowThrows**() **const** -> bool

Template Class BinaryExpr

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::ITransientExpression` (*Struct ITransientExpression*)

Class Documentation

template<typename **LhsT**, typename **RhsT**>
class `Catch::`**BinaryExpr** : **public** Catch::*ITransientExpression*

Public Functions

BinaryExpr(bool *comparisonResult*, *LhsT* *lhs*, *StringRef* *op*, *RhsT* *rhs*)

template<typename **T**>
auto **operator&&**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator||**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator==**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator!=**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator>**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator<**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator>=**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **T**>
auto **operator<=**(*T*) **const** -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

Class Capturer

- Defined in file_source_catch_catch.hpp

Class Documentation

class Catch::**Capturer**

Public Functions

Capturer(*StringRef* *macroName*, *SourceLineInfo* **const** &*lineInfo*, *ResultWas*::*OfType* *resultType*, *StringRef* *names*)

~Capturer()

void **captureValue**(size_t *index*, std::string **const** &*value*)

template<typename **T**>
void **captureValues**(size_t *index*, *T* **const** &*value*)

template<typename **T**, typename ...**Ts**>
void **captureValues**(size_t *index*, *T* **const** &*value*, *Ts* **const**&... *values*)

Class Approx

- Defined in file_source_catch_catch.hpp

Class Documentation

class Catch::Detail::**Approx**

Public Functions

Approx(double *value*)

Approx **operator-**() **const**

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
Approx **operator()**(*T* **const** &*value*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
Approx(*T* **const** &*value*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
Approx &**epsilon**(*T* **const** &*newEpsilon*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
Approx &**margin**(*T* **const** &*newMargin*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
Approx &**scale**(*T* **const** &*newScale*)

std::string **toString**() **const**

Public Static Functions

Approx **custom**()

Friends

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator==**(**const** *T* &*lhs*, *Approx* **const** &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator==**(*Approx* **const** &*lhs*, **const** *T* &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator!=**(*T* **const** &*lhs*, *Approx* **const** &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator!=**(*Approx* **const** &*lhs*, *T* **const** &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator<=**(*T* **const** &*lhs*, *Approx* **const** &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator<=**(*Approx* **const** &*lhs*, *T* **const** &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator>=**(*T* **const** &*lhs*, *Approx* **const** &*rhs*)

template<typename **T**, typename = **typename** std::enable_if<std::is_constructible<double, *T*>::value>::type>
friend bool **operator>=**(*Approx* **const** &*lhs*, *T* **const** &*rhs*)

Template Class IsStreamInsertable

- Defined in file_source_catch_catch.hpp

Class Documentation

template<typename **T**>
class Catch::Detail::**IsStreamInsertable**

Public Static Attributes

const bool **value** = decltype(test<std::ostream, const T&>(0))::value

Class ExceptionTranslatorRegistrar

- Defined in file_source_catch_catch.hpp

Nested Relationships

Nested Types

- *Template Class ExceptionTranslatorRegistrar::ExceptionTranslator*

Class Documentation

class Catch::**ExceptionTranslatorRegistrar**

Public Functions

template<typename **T**>
ExceptionTranslatorRegistrar(std::string (**translateFunction*))*T*&

Template Class ExceptionTranslatorRegistrar::ExceptionTranslator

- Defined in file_source_catch_catch.hpp

Nested Relationships

This class is a nested type of *Class ExceptionTranslatorRegistrar*.

Inheritance Relationships

Base Type

- `public Catch::IExceptionTranslator` (*Struct IExceptionTranslator*)

Class Documentation

template<typename **T**>
class `Catch::`*`ExceptionTranslatorRegistrar`*`::ExceptionTranslator` : **public** Catch::*IExceptionTranslator*

Public Functions

ExceptionTranslator(std::string (**translateFunction*)) *T*&

std::string **translate**(*ExceptionTranslators*::const_iterator *it*, *ExceptionTranslators*::const_iterator *itEnd*) **const override**

Protected Attributes

std::string (***m_translateFunction**)(*T*&)

Template Class ExprLhs

- Defined in file_source_catch_catch.hpp

Class Documentation

template<typename **LhsT**>
class `Catch::ExprLhs`

Public Functions

ExprLhs(*LhsT lhs*)

template<typename **RhsT**>
auto **operator==**(*RhsT* **const** &*rhs*) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

auto **operator==**(bool *rhs*) -> *BinaryExpr*<*LhsT*, bool> **const**

template<typename **RhsT**>
auto **operator!=**(*RhsT* **const** &*rhs*) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

auto **operator!=**(bool *rhs*) -> *BinaryExpr*<*LhsT*, bool> **const**

template<typename **RhsT**>
auto **operator>**(*RhsT* **const** &*rhs*) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **RhsT**>
auto **operator<**(*RhsT* **const** &*rhs*) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **RhsT**>

auto **operator>=**(*RhsT* **const** &*rhs*) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **RhsT**>
auto **operator<=**(*RhsT* **const** &*rhs*) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **RhsT**>
auto **operator&&**(*RhsT* **const**&) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

template<typename **RhsT**>
auto **operator||**(*RhsT* **const**&) -> *BinaryExpr*<*LhsT*, *RhsT* **const**&> **const**

auto **makeUnaryExpr**() **const** -> *UnaryExpr*<*LhsT*>

Class GeneratorException

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public exception`

Class Documentation

class `Catch::`**GeneratorException** : **public** exception

Public Functions

GeneratorException(**const** char **msg*)

const char ***what**() **const noexcept final override**

Template Class ChunkGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< std::vector< T > >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class Catch::Generators::**ChunkGenerator** : **public** Catch::Generators::*IGenerator*<std::vector<*T*>>

Public Functions

ChunkGenerator(size_t *size*, *GeneratorWrapper*<*T*> *generator*)

std::vector<*T*> **const** &**get**() **const override**

bool **next**() **override**

Template Class FilterGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Generators::IGenerator< T > (*Template Struct IGenerator*)

Class Documentation

template<typename **T**, typename **Predicate**>
class Catch::Generators::**FilterGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

template<typename **P** = *Predicate*>
FilterGenerator(*P* &&*pred*, *GeneratorWrapper*<*T*> &&*generator*)

T **const** &**get**() **const override**

bool **next**() **override**

Template Class FixedValuesGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class `Catch::Generators::`**FixedValuesGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

FixedValuesGenerator(std::initializer_list<*T*> *values*)

T **const** &**get**() **const override**

bool **next**() **override**

Template Class Generators

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class `Catch::Generators::`**Generators** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

template<typename ...**Gs**>
Generators(*Gs*&&... *moreGenerators*)

T **const** &**get**() **const override**

bool **next**() **override**

Class GeneratorUntypedBase

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Types

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)
- `public Catch::Generators::IGenerator< Float >` (*Template Struct IGenerator*)
- `public Catch::Generators::IGenerator< Integer >` (*Template Struct IGenerator*)
- `public Catch::Generators::IGenerator< std::vector< T > >` (*Template Struct IGenerator*)

Class Documentation

class `Catch::Generators::`**GeneratorUntypedBase**

Subclassed by *Catch::Generators::IGenerator< T >*, *Catch::Generators::IGenerator< Float >*, *Catch::Generators::IGenerator< Integer >*, *Catch::Generators::IGenerator< std::vector< T > >*

Public Functions

GeneratorUntypedBase() = default

~GeneratorUntypedBase()

bool **next**() = 0

Template Class GeneratorWrapper

- Defined in file_source_catch_catch.hpp

Class Documentation

template<typename **T**>
class `Catch::Generators::`**GeneratorWrapper**

Public Functions

GeneratorWrapper(std::unique_ptr<*IGenerator*<*T*>> *generator*)

T **const** &**get**() **const**

bool **next**()

Template Class IteratorGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class `Catch::Generators::`**IteratorGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

template<typename **InputIterator**, typename **InputSentinel**>
IteratorGenerator(*InputIterator* *first*, *InputSentinel* *last*)

T **const** &**get**() **const override**

bool **next**() **override**

Template Class MapGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**, typename **U**, typename **Func**>
class `Catch::Generators::`**MapGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

template<typename **F2** = *Func*>
MapGenerator(*F2* &&*function*, *GeneratorWrapper*<*U*> &&*generator*)

T **const** &**get**() **const override**

bool **next**() **override**

Template Class RandomFloatingGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< Float >` (*Template Struct IGenerator*)

Class Documentation

template<typename **Float**>
class `Catch::Generators::`**RandomFloatingGenerator** : **public** Catch::Generators::*IGenerator*<*Float*>

Public Functions

RandomFloatingGenerator(*Float a*, *Float b*)

Float **const** &**get**() **const override**

bool **next**() **override**

Template Class RandomIntegerGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< Integer >` (*Template Struct IGenerator*)

Class Documentation

template<typename **Integer**>
class Catch::Generators::**RandomIntegerGenerator** : **public** Catch::Generators::*IGenerator*<*Integer*>

Public Functions

RandomIntegerGenerator(*Integer* *a*, *Integer* *b*)

Integer **const** &**get**() **const** **override**

bool **next**() **override**

Template Class RangeGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::Generators::IGenerator< T > (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class Catch::Generators::**RangeGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

RangeGenerator(*T* **const** &*start*, *T* **const** &*end*, *T* **const** &*step*)

RangeGenerator(*T* **const** &*start*, *T* **const** &*end*)

T **const** &**get**() **const** **override**

bool **next**() **override**

Template Class RepeatGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class `Catch::Generators::`**RepeatGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

RepeatGenerator(size_t *repeats*, *GeneratorWrapper*<*T*> &&*generator*)

T **const** &**get**() **const override**

bool **next**() **override**

Template Class SingleValueGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class `Catch::Generators::`**SingleValueGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

SingleValueGenerator(*T* &&*value*)

T **const** &**get**() **const override**

bool **next**() **override**

Template Class TakeGenerator

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Generators::IGenerator< T >` (*Template Struct IGenerator*)

Class Documentation

template<typename **T**>
class `Catch::Generators::`**TakeGenerator** : **public** Catch::Generators::*IGenerator*<*T*>

Public Functions

TakeGenerator(size_t *target*, *GeneratorWrapper*<*T*> &&*generator*)

T **const** &**get**() **const override**

bool **next**() **override**

Class LazyExpression

- Defined in file_source_catch_catch.hpp

Class Documentation

class `Catch::`**LazyExpression**

Public Functions

LazyExpression(bool *isNegated*)

LazyExpression(*LazyExpression* **const** &*other*)

LazyExpression &**operator=**(*LazyExpression* **const**&) = delete

operator bool() **const**

Friends

friend auto **operator<<**(std::ostream &*os*, *LazyExpression* **const** &*lazyExpr*) -> std::ostream&

Class ExceptionMessageMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< std::exception >` (*Template Struct MatcherBase*)

Class Documentation

class `Catch::Matchers::Exception::`**ExceptionMessageMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<

Public Functions

ExceptionMessageMatcher(std::string **const** &*message*)

bool **match**(std::exception **const** &*ex*) **const override**

std::string **describe**() **const override**

Template Class PredicateMatcher

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::Matchers::Impl::MatcherBase< T >` (*Template Struct MatcherBase*)

Class Documentation

template<typename **T**>
class `Catch::Matchers::Generic::`**PredicateMatcher** : **public** Catch::Matchers::Impl::*MatcherBase*<*T*>

Public Functions

PredicateMatcher(std::function<bool(*T* **const**&)> **const** &*elem*, std::string **const** &*descr*

bool **match**(*T* **const** &*item*) **const override**

std::string **describe**() **const override**

Class MatcherUntypedBase

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Types

- public Catch::Matchers::Impl::MatcherBase< T > (*Template Struct MatcherBase*)
- public Catch::Matchers::Impl::MatcherBase< ArgT > (*Template Struct MatcherBase*)
- public Catch::Matchers::Impl::MatcherBase< double > (*Template Struct MatcherBase*)
- public Catch::Matchers::Impl::MatcherBase< std::exception > (*Template Struct MatcherBase*)
- public Catch::Matchers::Impl::MatcherBase< std::string > (*Template Struct MatcherBase*)
- public Catch::Matchers::Impl::MatcherBase< std::vector< T > > (*Template Struct MatcherBase*)

Class Documentation

class Catch::Matchers::Impl::**MatcherUntypedBase**

Subclassed by *Catch::Matchers::Impl::MatcherBase< T >*, *Catch::Matchers::Impl::MatcherBase< ArgT >*, *Catch::Matchers::Impl::MatcherBase< double >*, *Catch::Matchers::Impl::MatcherBase< std::exception >*, *Catch::Matchers::Impl::MatcherBase< std::string >*, *Catch::Matchers::Impl::MatcherBase< std::vector< T > >*

Public Functions

MatcherUntypedBase() = default

MatcherUntypedBase(*MatcherUntypedBase* **const**&) = default

MatcherUntypedBase &**operator=**(*MatcherUntypedBase* **const**&) = delete

std::string **toString**() **const**

Protected Functions

~MatcherUntypedBase()

std::string **describe**() **const** = 0

Protected Attributes

std::string **m_cachedToString**

Template Class MatchExpr

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::ITransientExpression` (*Struct ITransientExpression*)

Class Documentation

template<typename **ArgT**, typename **MatcherT**>
class `Catch::MatchExpr` : **public** Catch::*ITransientExpression*

Public Functions

MatchExpr(*ArgT* **const** &*arg*, *MatcherT* **const** &*matcher*, *StringRef* **const** &*matcherString*)

void **streamReconstructedExpression**(std::ostream &*os*) **const override**

Class NonCopyable

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Derived Types

- `public Catch::AutoReg` (*Struct AutoReg*)
- `public Catch::IConfig` (*Struct IConfig*)
- `private Catch::ReusableStringStream` (*Class ReusableStringStream*)
- `private Catch::Section` (*Class Section*)

Class Documentation

class Catch::**NonCopyable**
Subclassed by *Catch::AutoReg*, *Catch::IConfig*, *Catch::ReusableStringStream*, *Catch::Section*

Protected Functions

NonCopyable()

~NonCopyable()

Template Class Option

- Defined in file_source_catch_catch.hpp

Class Documentation

template<typename **T**>
class Catch::**Option**

Public Functions

Option()

Option(*T* **const** &*_value*)

Option(*Option* **const** &*_other*)

~Option()

Option &**operator=**(*Option* **const** &*_other*)

Option &**operator=**(*T* **const** &*_value*)

void **reset**()

T &**operator***()

T **const** &**operator***() **const**

T ***operator->**()

const *T* ***operator->**() **const**

T **valueOr**(*T* **const** &*defaultValue*) **const**

bool **some**() **const**

bool **none**() **const**

bool **operator!**() **const**

operator bool() **const**

Class ReusableStringStream

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `private Catch::NonCopyable` (*Class NonCopyable*)

Class Documentation

class `Catch::`**ReusableStringStream** : **private** Catch::*NonCopyable*

Public Functions

ReusableStringStream()

~ReusableStringStream()

auto **str**() **const** -> std::string

template<typename **T**>
auto **operator<<**(*T* **const** &*value*) -> *ReusableStringStream*&

auto **get**() -> std::ostream&

Class ScopedMessage

- Defined in file_source_catch_catch.hpp

Class Documentation

class `Catch::`**ScopedMessage**

Public Functions

ScopedMessage(*MessageBuilder* **const** &*builder*)

ScopedMessage(*ScopedMessage* &*duplicate*) = delete

ScopedMessage(*ScopedMessage* &&*old*)

~ScopedMessage()

Public Members

MessageInfo **m_info**

bool **m_moved**

Class Section

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `private Catch::NonCopyable` (*Class NonCopyable*)

Class Documentation

class `Catch::`**Section** : **private** Catch::*NonCopyable*

Public Functions

Section(*SectionInfo* **const** &*info*)

~Section()

operator bool() **const**

Class SimplePcg32

- Defined in file_source_catch_catch.hpp

Class Documentation

class `Catch::`**SimplePcg32**

Public Types

using result_type = std::uint32_t

Public Functions

SimplePcg32()

SimplePcg32(*result_type seed_*)

void **seed**(*result_type seed_*)

void **discard**(uint64_t *skip*)

result_type **operator()**()

Public Static Functions

constexpr result_type() min ()

constexpr result_type() max ()

Class StringRef

- Defined in file_source_catch_catch.hpp

Class Documentation

class Catch::**StringRef**

A non-owning string class (similar to the forthcoming std::string_view) Note that, because a *StringRef* may be a substring of another string, it may not be null terminated.

Public Types

using size_type = std::size_t

using const_iterator = **const** char*

Public Functions

constexpr StringRef() **noexcept** = default

StringRef(char **const** **rawChars*) **noexcept**

constexpr StringRef(char **const** **rawChars*, *size_type size*) **noexcept**

StringRef(std::string **const** &*stdString*) **noexcept**

operator std::string() **const**

auto **operator==**(*StringRef* **const** &*other*) **const noexcept** -> bool

auto **operator!=**(*StringRef* **const** &*other*) **const noexcept** -> bool

auto **operator[]**(*size_type index*) **const noexcept** -> char

constexpr auto **empty**() **const noexcept** -> bool

constexpr auto **size**() **const noexcept** -> *size_type*

auto **c_str**() **const** -> char **const***

auto **substr**(*size_type start*, *size_type length*) **const noexcept** -> *StringRef*

auto **data**() **const noexcept** -> char **const***

constexpr auto **isNullTerminated**() **const noexcept** -> bool

constexpr *const_iterator* **begin**() **const**

constexpr *const_iterator* **end**() **const**

Class TestCase

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::TestCaseInfo` (*Struct TestCaseInfo*)

Class Documentation

class `Catch::`**TestCase** : **public** Catch::*TestCaseInfo*

Public Functions

TestCase(*ITestInvoker *testCase*, *TestCaseInfo &&info*)

TestCase **withName**(std::string **const** &*_newName*) **const**

void **invoke**() **const**

TestCaseInfo **const** &**getTestCaseInfo**() **const**

bool **operator==**(*TestCase* **const** &*other*) **const**

bool **operator<**(*TestCase* **const** &*other*) **const**

Template Class TestInvokerAsMethod

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- `public Catch::ITestInvoker` (*Struct ITestInvoker*)

Class Documentation

template<typename **C**>
class Catch::**TestInvokerAsMethod** : **public** Catch::*ITestInvoker*

Public Functions

TestInvokerAsMethod(void (*C*::* *testAsMethod*))
noexcept

void **invoke**() **const override**

Class Timer

- Defined in file_source_catch_catch.hpp

Class Documentation

class Catch::**Timer**

Public Functions

void **start**()

auto **getElapsedNanoseconds**() **const** -> uint64_t

auto **getElapsedMicroseconds**() **const** -> uint64_t

auto **getElapsedMilliseconds**() **const** -> unsigned int

auto **getElapsedSeconds**() **const** -> double

Template Class UnaryExpr

- Defined in file_source_catch_catch.hpp

Inheritance Relationships

Base Type

- public Catch::ITransientExpression (*Struct ITransientExpression*)

Class Documentation

template<typename **LhsT**>
class Catch**::UnaryExpr** : **public** Catch::*ITransientExpression*

Public Functions

UnaryExpr(*LhsT lhs*)

Class EfficientHost

- Defined in file_source_efficient_mode_EfficientHost.h

Inheritance Relationships

Base Type

- public Host (*Class Host*)

Class Documentation

class EfficientHost : **public** *Host*

Public Functions

EfficientHost(emp::Ptr<emp::Random> *_random*, emp::Ptr<*EfficientWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, emp::vector<emp::Ptr<*Organism*>> *_syms* = { }, emp::vector<emp::Ptr<*Organism*>> *_repro_syms* = { }, std::set<int> *_set* = std::set<int>(), double *_points* = 0.0, double *_efficient* = 0.1)

The constructor for efficient host

EfficientHost(**const** *EfficientHost*&) = default

Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

EfficientHost(*EfficientHost*&&) = default

Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

EfficientHost() = default

Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

void **SetEfficiency**(double *_in*)
Input: Efficiency value

Output: None

Purpose: Setting an efficient symbiont's efficiency value.

double **GetEfficiency**()
Input: None

Output: A double representing the symbiont's efficiency.

Purpose: Getting an efficient symbiont's efficiency value.

emp::Ptr<*Organism*> **makeNew**()
Input: None.

Output: A new host with same properties as this host.

Purpose: To avoid creating an organism via constructor in other methods.

Protected Attributes

double **efficiency**
Purpose: Represents the efficiency of a host.

emp::Ptr<*EfficientWorld*> **my_world** = NULL
Purpose: Represents the world that the efficient hosts are living in.

Class EfficientSymbiont

- Defined in file_source_efficient_mode_EfficientSymbiont.h

Inheritance Relationships

Base Type

- `public Symbiont` (*Class Symbiont*)

Class Documentation

class EfficientSymbiont : **public** *Symbiont*

Public Functions

EfficientSymbiont(emp::Ptr<emp::Random> *_random*, emp::Ptr<*EfficientWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, double *_points* = 0.0, double *_efficient* = 0.1)
The constructor for efficient symbiont

EfficientSymbiont(**const** *EfficientSymbiont*&) = default
Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

EfficientSymbiont(*EfficientSymbiont*&&) = default
Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

EfficientSymbiont() = default
Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

void **SetEfficiency**(double *_in*)
Input: Efficiency value

Output: None

Purpose: Setting an efficient symbiont's efficiency value.

double **GetEfficiency**()
Input: None

Output: A double representing the symbiont's efficiency.

Purpose: Getting an efficient symbiont's efficiency value.

void **AddPoints**(double *_in*)
Input: A double representing the amount to be incremented to a symbiont's points.

Output: None

Purpose: Incrementing an efficient symbiont's points. The points are adjusted by the efficiency of the symbiont.

void **mutate**(std::string *mode*)
Input: String indicating mode, either "vertical" or "horizontal"

Output: None

Purpose: Mutating the efficiency of an efficient symbiont based upon the config setting for mutation size.

emp::Ptr<*Organism*> **makeNew**()
Input: None.

Output: A new bacterium with same properties as this bacterium.

Purpose: To avoid creating an organism via constructor in other methods.

emp::Ptr<*Organism*> **reproduce**(std::string *mode*)
Input: String to indicate the mode of transmission, either vertical or horizontal

Output: The pointer to the newly created organism

Purpose: To produce a new symbiont

void **VerticalTransmission**(emp::Ptr<*Organism*> *host_baby*)
Input: The pointer to the organism that is the new host baby

Output: None

Purpose: To allow for vertical transmission to occur

void **HorizontalTransmission**(emp::WorldPosition *location*)
Input: The location of the organism as a WorldPosition

Output: None

Purpose: To check and allow for horizontal transmission to occur

Protected Attributes

double **efficiency**
Purpose: Represents the efficiency of a symbiont. This has a multiplicable impact on a symbiont's resource collection.

double **ht_mut_size** = 0.002
Purpose: Represents the standard deviation of the values chosen as mutations of a symbiont's interaction value when horizontal transmission is occuring. .

double **ht_mut_rate** = 0
Purpose: Represents the probability (0-1) of mutation occurring during horizontal transmission.

double **eff_mut_rate** = 0
Purpose: Represents the probability (0-1) of mutation occurring during horizontal transmission for the efficiency trait.

emp::Ptr<*EfficientWorld*> **my_world** = NULL
Purpose: Represents the world that the efficient symbionts are living in.

Class EfficientWorld

- Defined in file_source_efficient_mode_EfficientWorld.h

Inheritance Relationships

Base Type

- `public SymWorld` (*Class SymWorld*)

Class Documentation

class EfficientWorld : **public** *SymWorld*

Public Functions

~EfficientWorld()
Input: None

Output: None

Purpose: To destruct the data nodes belonging to *EfficientWorld* to conserve memory.

emp::DataFile &**SetupEfficiencyFile**(**const** std::string &*filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track mean efficiency

emp::DataMonitor<double> &**GetEfficiencyDataNode**()
Input: None

Output: The DataMonitor<double>& that has the information representing the symbiont's efficiency.

Purpose: To collect data on the lysis burst size to be saved to the data file that is tracking lysis burst size.

Class Host

- Defined in file_source_default_mode_Host.h

Inheritance Relationships

Base Type

- `public Organism` (*Class Organism*)

Derived Types

- `public Bacterium` (*Class Bacterium*)
- `public EfficientHost` (*Class EfficientHost*)
- `public PggHost` (*Class PggHost*)

Class Documentation

class Host : **public** *Organism*
Subclassed by *Bacterium*, *EfficientHost*, *PggHost*

Public Functions

Host(emp::Ptr<emp::Random> *_random*, emp::Ptr<*SymWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, emp::vector<emp::Ptr<*Organism*>> *_syms* = {}, emp::vector<emp::Ptr<*Organism*>> *_repro_syms* = {}, std::set<int> *_set* = std::set<int>(), double *_points* = 0.0)
The constructor for the host class

~Host()
Input: None

Output: None

Purpose: To delete the memory used by a host's symbionts when the host is deleted.

Host(**const** *Host*&) = default
Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

Host(*Host*&&) = default
Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

Host() = default
Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

Host &**operator=**(**const** *Host*&) = default
Input: None

Output: None

Purpose: To force a copy assignment operator to be generated by the compiler.

Host &**operator=**(*Host*&&) = default
Input: None

Output: None

Purpose: To force a move assignment operator to be generated by the compiler.

bool **operator==**(**const** *Host* &*other*) **const**
Input: An object of host to be compared to the current host.

Output: To boolean representing if thing1 == &thing2

Purpose: To override the bool operator == to return (thing1 == &thing2)

bool **operator!=**(**const** *Host* &*other*) **const**
Input: An object of host, and the address of the thing it is being compared to.

Output: To boolean representing if *thing1 == thing2

Purpose: To override the bool operator != to return !(*thing1 == thing2)

double **GetIntVal**() **const**
Input: None

Output: The double representing host's interaction value

Purpose: To get the double representing host's interaction value

emp::vector<emp::Ptr<*Organism*>> &**GetSymbionts**()
Input: None

Output: A vector of pointers to the organisms that are the host's syms.

Purpose: To get the vector containing pointers to the host's symbionts.

emp::vector<emp::Ptr<*Organism*>> &**GetReproSymbionts**()
Input: None

Output: A vector of pointers to the organisms that are the host's repro syms.

Purpose: To get the vector containing pointers to the host's repro syms.

std::set<int> **GetResTypes**() **const**
Input: None

Output: The set of ints representing a host's res type.

Purpose: To get the set of ints representing the host's res type.

double **GetPoints**()
Input: None

Output: The double representing a host's points.

Purpose: To get the host's points.

double **GetResInProcess**()
Input: None

Output: The double representing res_in_process

Purpose: To get the value of res_in_process

bool **IsHost**()
Input: None

Output: The bool representing if an organism is a host.

Purpose: To determine if an organism is a host.

void **SetIntVal**(double *_in*)
Input: A double representing the host's new interaction value.

Output: None

Purpose: To set a host's interaction value.

void **SetSymbionts**(emp::vector<emp::Ptr<*Organism*>> *_in*)
Input: A vector of pointers to organisms that will become a host's symbionts.

Output: None

Purpose: To set a host's symbionts to the input vector of organisms.

void **SetResTypes**(std::set<int> *_in*)
Input: A set of ints representing a host's resource type.

Output: None

Purpose: To set a host's resource types to the input.

void **SetPoints**(double *_in*)
Input: A double representing a host's new point value.

Output: None

Purpose: To set a host's points.

void **ClearSyms**()
Input: None

Output: None

Purpose: To clear a host's symbionts.

void **ClearReproSyms**()
Input: None

Output: None

Purpose: To clear a host's repro symbionts.

void **SetDead**()
Input: None

Output: None

Purpose: To kill a host.

void **SetResInProcess**(double *_in*)
Input: The double to be set as res_in_process

Output: None

Purpose: To set the value of res_in_process

bool **GetDead**()
Input: None

Output: boolean

Purpose: To determine if a host is dead.

int **GetAge**()
Input: None

Output: an int representing the current age of the *Host*

Purpose: To get the *Host*'s age.

void **SetAge**(int *_in*)
Input: An int of what age the *Host* should be set to

Output: None

Purpose: To set the *Host*'s age for testing purposes.

void **GrowOlder**()
Input: None

Output: None

Purpose: Increments age by one and kills it if too old.

double **StealResources**(double *_intval*)
Input: The interaction value of the symbiont that is eligible to steal resources from the host.

Output: The double representing the amount of resources that are actually stolen from the host.

Purpose: To determine if a host's symbiont is eligible to steal resources from the host.

void **AddPoints**(double *_in*)
Input: The double representing the number of points to be incremented onto a host's points.

Output: None

Purpose: To increment a host's points by the input value.

void **AddSymbiont**(emp::Ptr<*Organism*> *_in*)
Input: The pointer to the organism that is to be added to the host's symbionts.

Output: None

Purpose: To add a symbionts to a host's symbionts

bool **SymAllowedIn**()
Input: None

Output: A bool representing if a symbiont will be allowed to enter a host.

Purpose: To determine if a symbiont will be allowed into a host. If phage exclusion is off, this function will always return true. If phage exclusion is on, then there is a 1/2^n chance of a new phage being allowed in, where n is the number of existing phage.

void **AddReproSym**(emp::Ptr<*Organism*> *_in*)
Input: A pointer to the organism to be added to the host's symbionts.

Output: None

Purpose: To add a repro sym to the host's symbionts.

bool **HasSym**()
Input: None

Output: A bool representing if a host has any symbionts.

Purpose: To determine if a host has any symbionts, though they might be corpses that haven't been removed yet.

emp::Ptr<*Organism*> **makeNew**()
Input: None.

Output: A new host with same properties as this host.

Purpose: To avoid creating an organism via constructor in other methods.

emp::Ptr<*Organism*> **reproduce**()
Input: None.

Output: A new host baby of the current host, mutated.

Purpose: To create a new baby host and reset this host's points to 0.

void **mutate**()
Input: None

Output: None

Purpose: To mutate a host's interaction value. This is called on newly generated hosts to allow for evolution to occur.

void **DistribResources**(double *resources*)
Input: The double representing the number of resources to be distributed to the host and its symbionts and the position of the host in the world.

Output: None

Purpose: To distribute resources to a host and its symbionts. In the event that the host has no symbionts, the host gets all resources not allocated to defense or given to absent partner. Otherwise, the resource is split into equal chunks for each symbiont

double **HandleEctosymbiosis**(double *resources*, size_t *location*)
Input: The total resources recieved by the host and its location in the world.

Output: The resources remaining after the host maybe does ectosymbiosis.

Purpose: To handle ectosymbiosis.

bool **GetDoEctosymbiosis**(size_t *location*)
Input: The location of this host in the world.

Output: A bool value representing whether this host should interact with a parallel sym

Purpose: To determine whether a host should interact with a parallel sym

void **DistribResToSym**(emp::Ptr<*Organism*> *sym*, double *sym_piece*)
Input: The sym to whom resources are distributed and the resources it might recieve.

Output: None

Purpose: To distribute resources between sym and host depending on their interaction values.

void **Process** (emp::WorldPosition *pos*)
Input: The size_t value representing the location of the host.

Output: None

Purpose: To process the host, meaning determining eligibility for reproduction, checking for vertical transmission, removing dead syms, and processing alive syms.

Protected Attributes

double **interaction_val** = 0
Purpose: Represents the interaction value between the host and symbiont. A negative interaction value represent antagonism, while a positive one represents mutualism. Zero is a neutral value.

int **age** = 0
Purpose: Represents the number of updates the host has lived through; at birth is set to 0.

emp::vector<emp::Ptr<*Organism*>> **syms** = { }
Purpose: Represents the set of symbionts belonging to a host. This can be set with *SetSymbionts()*, and symbionts can be added with *AddSymbiont()*. This can be cleared with *ClearSyms()*

emp::vector<emp::Ptr<*Organism*>> **repro_syms** = { }
Purpose: Represents the set of repro symbionts belonging to a host. Symbionts can be added with AddReproSymb(). This can be cleared with *ClearSyms()*

std::set<int> **res_types** = { }
Purpose: Represents the resource type available to hosts. This is currently not implemented fully.

double **points** = 0
Purpose: Represents the resource points possessed by a host. This is what hosts must collect to reproduce.

double **res_in_process** = 0
Purpose: Represents the resources that could be in the process of being stolen.

emp::Ptr<emp::Random> **random** = NULL
Purpose: Represents an instance of random.

emp::Ptr<*SymWorld*> **my_world** = NULL
Purpose: Represents the world that the hosts are living in.

emp::Ptr<SymConfigBase> **my_config** = NULL
Purpose: Represents the configuration settings for a particular run.

bool **dead** = false
Purpose: Represents if a host is alive. This is set to true when a host is killed.

Class LysisWorld

- Defined in file_source_lysis_mode_LysisWorld.h

Inheritance Relationships

Base Type

- `public SymWorld` (*Class SymWorld*)

Class Documentation

class LysisWorld : **public** *SymWorld*

Public Functions

~LysisWorld()
Input: None

Output: None

Purpose: To destruct the data nodes belonging to *LysisWorld* to conserve memory.

emp::DataFile &**SetupLysisChanceFile**(**const** std::string &*filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track mean lysis chance, the number of symbionts, and the histogram of the mean lysis chance.

emp::DataFile &**SetupInductionChanceFile**(**const** std::string &*filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track mean induction chance, the number of symbionts, and the histogram of the mean induction chance.

emp::DataFile &**SetupIncorporationDifferenceFile**(**const** std::string &*filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track the difference between bacterium and phage incorporation values and the histogram of the difference between the incorporation vals.

emp::DataMonitor<double, emp::data::Histogram> &**GetLysisChanceDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the chance of lysis for each symbiont.

Purpose: To collect data on the chance of lysis for each symbiont to be saved to the data file that is tracking the chance of lysis for each symbiont.

emp::DataMonitor<double> &**GetBurstSizeDataNode**()
Input: None

Output: The DataMonitor<double>& that has the information representing the lysis burst size.

Purpose: To collect data on the lysis burst size to be saved to the data file that is tracking lysis burst size.

emp::DataMonitor<int> &**GetBurstCountDataNode**()
Input: None

Output: The DataMonitor<int>& that has the information representing the lysis burst count.

Purpose: To collect data on the lysis burst count to be saved to the data file that is tracking lysis burst count.

emp::DataMonitor<double, emp::data::Histogram> &**GetInductionChanceDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the chance of induction for each symbionts.

Purpose: To collect data on the chance of induction for each symbiont to be saved to the data file that is tracking chance of induction for each symbiont.

emp::DataMonitor<double, emp::data::Histogram> &**GetIncorporationDifferenceDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the difference between incorporation vals for bacteriums and their phage

Purpose: To collect data on the difference between incorporation vals for each bacteria and their phage to be saved to the data file that is tracking incorporation val differences.

Class Organism

- Defined in file_source_Organism.h

Inheritance Relationships

Derived Types

- `public Host` (*Class Host*)
- `public Symbiont` (*Class Symbiont*)

Class Documentation

class Organism
Subclassed by *Host*, *Symbiont*

Public Functions

Organism() = default

Organism(**const** *Organism*&) = default

Organism(*Organism*&&) = default

~Organism()

Organism &**operator=**(**const** *Organism*&) = default

Organism &**operator=**(*Organism*&&) = default

bool **operator==**(**const** *Organism* &*other*) **const**

bool **operator!=**(**const** *Organism* &*other*) **const**

double **GetIntVal**() **const**

double **GetPoints**()

void **SetIntVal**(double *_in*)

void **SetPoints**(double *_in*)

void **AddPoints**(double *_in*)

void **SetHost**(emp::Ptr<*Organism*> *_in*)

void **SetDead**()

bool **GetDead**()

void **Process**(emp::WorldPosition *location*)

double **GetIncVal**()

void **SetIncVal**(double *_in*)

int **GetAge**()

void **SetAge**(int *_in*)

emp::Ptr<*Organism*> **makeNew**()

void **mutate**()

emp::Ptr<*Organism*> **reproduce**()

void **VerticalTransmission**(emp::Ptr<*Organism*> *host_baby*)

void **HorizontalTransmission**(emp::WorldPosition *location*)

double **ProcessResources**(double *sym_piece*)

double **ProcessResources**(double *sym_piece*, emp::Ptr<*Organism*> *host*)

bool **IsPhage**()

emp::Ptr<*Organism*> **GetHost**()

bool **WantsToInfect**()

double **GetInfectionChance**()

void **SetInfectionChance**(double *_in*)

bool **InfectionFails**()

emp::Ptr<emp::Taxon<int>> **GetTaxon**()

void **SetTaxon**(emp::Ptr<emp::Taxon<int>> *_in*)

double **GetEfficiency**()

void **SetEfficiency**(double *_in*)

emp::Ptr<*Organism*> **reproduce**(std::string *mode*)

void **mutate**(std::string *mode*)

emp::vector<emp::Ptr<*Organism*>> &**GetSymbionts**()

emp::vector<emp::Ptr<*Organism*>> &**GetReproSymbionts**()

std::set<int> **GetResTypes** () **const**

void **SetResInProcess** (double *_in*)

double **GetResInProcess** ()

double **StealResources** (double *_intval*)

void **SetSymbionts** (emp::vector<emp::Ptr<*Organism*>> *_in*)

void **SetResTypes** (std::set<int> *_in*)

void **AddSymbiont** (emp::Ptr<*Organism*> *_in*)

void **AddReproSym** (emp::Ptr<*Organism*> *_in*)

bool **HasSym** ()

bool **IsHost** ()

void **DistribResources** (double *resources*)

void **ClearSyms** ()

void **ClearReproSyms** ()

double **ProcessLysogenResources** (double *phage_inc_val*)

double **GetBurstTimer** ()

void **IncBurstTimer** ()

void **SetBurstTimer** (int *_in*)

double **GetLysisChance** ()

double **GetInductionChance** ()

void **SetLysisChance** (double *_in*)

void **SetInductionChance** (double *_in*)

void **uponInjection** ()

void **LysisBurst** (emp::WorldPosition *location*)

void **LysisStep** ()

bool **GetLysogeny** ()

void **SetPool** ()

void **AddPool** ()

void **DistribPool** ()

double **GetDonation** ()

void **Setdonation** ()

double **ProcessPool** ()

Class PggHost

- Defined in file_source_pgg_mode_Pgghost.h

Inheritance Relationships

Base Type

- `public Host` (*Class Host*)

Class Documentation

class PggHost : **public** *Host*

Public Functions

PggHost(emp::Ptr<emp::Random> *_random*, emp::Ptr<*PggWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, emp::vector<emp::Ptr<*Organism*>> *_syms* = {}, emp::vector<emp::Ptr<*Organism*>> *_repro_syms* = {}, std::set<int> *_set* = std::set<int>(), double *_points* = 0.0)

PggHost(**const** *PggHost*&) = default
Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

PggHost(*PggHost*&&) = default
Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

PggHost() = default
Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

double **GetPool**()
Input: None

Output: #

Purpose:

void **SetPool**(double *_in*)
Input: #

Output: None

Purpose:

void **AddPool**(double *_in*)
Input: #

Output: None

Purpose:

void **DistribResources**(double *resources*)
Input: #

Output: None

Purpose:

void **DistribPool**()
Input: None

Output: #

Purpose:

emp::Ptr<*Organism*> **makeNew**()
Input: None.

Output: A new pgghost with same properties as this pgghost.

Purpose: To avoid creating an organism via constructor in other methods.

Protected Attributes

double **sourcepool** = 0
Purpose:#

emp::Ptr<*PggWorld*> **my_world** = NULL
Purpose: Represents the world that the pgg hosts are living in.

Class PGGSymbiont

- Defined in file_source_pgg_mode_Pggsym.h

Inheritance Relationships

Base Type

- `public Symbiont` (*Class Symbiont*)

Class Documentation

class PGGSymbiont : **public** *Symbiont*

Public Functions

PGGSymbiont(emp::Ptr<emp::Random> *_random*, emp::Ptr<*PggWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, double *_donation* = 0.0, double *_points* = 0.0)

PGGSymbiont(**const** *PGGSymbiont*&) = default

Input: #

Output:

Purpose:

PGGSymbiont(*PGGSymbiont*&&) = default

Input: #

Output:

Purpose:

PGGSymbiont() = default

Input: #

Output:

Purpose:

PGGSymbiont &**operator=**(**const** *PGGSymbiont*&) = default

Input: #

Output:

Purpose:

PGGSymbiont &**operator=**(*PGGSymbiont*&&) = default

Input: #

Output:

Purpose:

double **GetDonation**()

Input: #

Output:

Purpose:

void **Setdonation**(double *_in*)

Input: #

Output:

Purpose:

void **mutate**()

Input: #

Output:

Purpose:

double **ProcessPool**()

Input: #

Output:

Purpose:

emp::Ptr<*Organism*> **makeNew**()
Input: None

Output: The pointer to the newly created organism

Purpose: To produce a new phage, identical to the original

std::string **PrintSym**(emp::Ptr<*PGGSymbiont*> *org*)
Input: #

Output:

Purpose:

Protected Attributes

double **Pgg_donate** = 0
Purpose: #

emp::Ptr<*PggWorld*> **my_world** = NULL
Purpose: Represents the world that the pgg symbionts are living in.

Class PggWorld

- Defined in file_source_pgg_mode_PggWorld.h

Inheritance Relationships

Base Type

- `public SymWorld` (*Class SymWorld*)

Class Documentation

class PggWorld : **public** *SymWorld*

Public Functions

~PggWorld()
Input: None

Output: None

Purpose: To destruct the data nodes belonging to *PggWorld* to conserve memory.

emp::DataFile &**SetupPGGSymIntValFile**(**const** std::string &*filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track information about the PGG symbiont's interaction values

emp::DataMonitor<double, emp::data::Histogram> &**GetPGGDataNode**()
Input: The double representing the number of resources each host gets in each update.

Output: None

Purpose: To set the resources that each host gets per update.

Class Phage

- Defined in file_source_lysis_mode_Phage.h

Inheritance Relationships

Base Type

- `public Symbiont` (*Class Symbiont*)

Class Documentation

class Phage : **public** *Symbiont*

Public Functions

Phage(emp::Ptr<emp::Random> *_random*, emp::Ptr<*LysisWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, double *_points* = 0.0)
The constructor for phage

Phage(**const** *Phage*&) = default
Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

Phage(*Phage*&&) = default
Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

Phage() = default
Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

double **GetBurstTimer**()
Input: None

Output: The double representing the phage's burst timer.

Purpose: To get a phage's burst timer.

void **IncBurstTimer**()
Input: None

Output: None

Purpose: To increment a phage's burst timer.

void **SetBurstTimer**(double *_in*)
Input: The double to be set as the phage's burst timer

Output: None

Purpose: To set a phage's burst timer.

double **GetLysisChance**()
Input: None

Output: The double representing a phage's change of lysis.

Purpose: To determine a phage's chance of lysis.

void **SetLysisChance**(double *_in*)
Input: The double to be set as the phage's chance of lysis.

Output: None

Purpose: To set a phage's chance of lysis

double **GetIncVal**()
Input: None

Output: The double representing a phage's incorporation value.

Purpose: To determine a phage's incorporation value.

void **SetIncVal**(double *_in*)
Input: The double to be set as the phage's incorporation value.

Output: None

Purpose: To set a phage's incorporation value.

double **GetInductionChance**()
Input: None

Output: The double representing a prophage's chance of induction.

Purpose: To determine a lysogenic phage's chance of inducing

void **SetInductionChance**(double *_in*)
Input: The double to be set as the phage's chance of induction

Output:None

Purpose: To set a phage's chance of inducing

bool **GetLysogeny**()
Input: None

Output: The bool representing if a phage will do lysogeny.

Purpose: To determine if a phage is capable of lysogeny

bool **IsPhage**()
Input: None

Output: The bool representing if an organism is a phage, always true.

Purpose: To determine if an organism is a phage.

void **uponInjection**()
Input: None

Output: None

Purpose: To determine if a phage will choose lysis or lysogeny. If a phage chooses to be lytic, their interaction value will be -1 to represent them being antagonistic. If a phage chooses to be lysogenic, their interaction value will be 0 to represent them being neutral.

void **mutate**()
Input: Optional string that indicates mode of reproduction and therefore which mutation rate and size should be used. Options are vertical (for prophage) or horizontal (for lytic phage)

Output: None

Purpose: To mutate a phage's chance of lysis. The mutation will be based on a value chosen from a normal distribution centered at 0, with a standard deviation that is equal to the mutation size. *Phage* mutation can be on or off.

emp::Ptr<*Organism*> **makeNew**()
Input: None

Output: The pointer to the newly created organism

Purpose: To produce a new symbiont, identical to the original

void **LysisBurst**(emp::WorldPosition *location*)
Input: location of the phage attempting to horizontally transmit

Output: None

Purpose: To burst host and release offspring

void **LysisStep**()
Input: None

Output: None

Purpose: To allow lytic phage to produce offspring and increment the burst timer

void **VerticalTransmission**(emp::Ptr<*Organism*> *host_baby*)
Input: A pointer to the baby host to have symbionts added.

Output: None

Purpose: To allow for vertical transmission to occur. lysogenic phage have 100% chance of vertical transmission, lytic phage have 0% chance

double **ProcessResources**(double *hostDonation*)
Input: The double representing the resources that will be given to a phage.

Output: The double representing the resources that are left over from what was distributed to the phage.

Purpose: To allow a phage to steal or use donated resources from their host.

void **Process**(emp::WorldPosition *location*)
Input: The worldposition representing the location of the phage being processed.

Output: None

Purpose: To process a phage, meaning check for reproduction, check for lysis, and move the phage.

Protected Attributes

double **burst_timer** = 0
Purpose: Represents the time until lysis will be triggered.

bool **lysogeny** = false
Purpose: Represents if lysogeny is on.

double **incorporation_val** = 0.0
Purpose: Represents the compatibility of the prophage to it's placement within the host's genome.

double **chance_of_lysis** = 1
Purpose: Represents the chance of lysis

double **induction_chance** = 1
Purpose: Represents the chance of a prophage inducing to the lytic process

emp::Ptr<*LysisWorld*> **my_world** = NULL
Purpose: Represents the world that the phage are living in.

Class SymAnimate

- Defined in file_source_SymAnimate.h

Inheritance Relationships

Base Type

- `public Animate`

Class Documentation

class SymAnimate : **public** Animate

Public Functions

SymAnimate()
The contructor for *SymAnimate*

void **initializeWorld**()
Input: None

Output: None

Purpose: To initialize the world based upon the config setting given

void **setButtonStyle**(std::string *but_id*)
Input: The string representing the button identification.

Output: None

Purpose: To add style to the buttons displayed.

void **drawPetriDish**(UI::Canvas &*can*)
Input: The canvas being used.

Output: None

Purpose: To draw the petri dish of basteria and phage.

std::string **matchColor**(double *intVal*)
Input: The double representing symbiont or host's interaction value

Output: The string representing the hex value for the color of the organism.

Purpose: To determine the color that an organism should be, given its interaction value.

void **DoFrame**()
Input: None

Output: None

Purpose: To update the frame displayed of the current world state.

Class Symbiont

- Defined in file_source_default_mode_Symbiont.h

Inheritance Relationships

Base Type

- `public Organism` (*Class Organism*)

Derived Types

- `public EfficientSymbiont` (*Class EfficientSymbiont*)
- `public PGGSymbiont` (*Class PGGSymbiont*)
- `public Phage` (*Class Phage*)

Class Documentation

class Symbiont : **public** *Organism*
Subclassed by *EfficientSymbiont*, *PGGSymbiont*, *Phage*

Public Functions

Symbiont(emp::Ptr<emp::Random> *_random*, emp::Ptr<*SymWorld*> *_world*, emp::Ptr<SymConfigBase> *_config*, double *_intval* = 0.0, double *_points* = 0.0)
The constructor for symbiont

Symbiont(**const** *Symbiont*&) = default
Input: None

Output: None

Purpose: To force a copy constructor to be generated by the compiler.

Symbiont(*Symbiont*&&) = default
Input: None

Output: None

Purpose: To force a move constructor to be generated by the compiler

Symbiont() = default
Input: None

Output: None

Purpose: To tell the compiler to use its default generated variants of the constructor

Symbiont &**operator=**(**const** *Symbiont*&) = default
Input: None

Output: None

Purpose: To force a copy assignment operator to be generated by the compiler.

Symbiont &**operator=**(*Symbiont*&&) = default
Input: None

Output: None

Purpose: To force a move assignment operator to be generated by the compiler.

~Symbiont()
Input: None

Output: None

Purpose: To destruct the symbiont and remove the symbiont from the systematic.

double **GetIntVal**() **const**
Input: None

Output: The double representing the symbiont's interaction value

Purpose: To get a symbiont's interaction value.

double **GetPoints**()
Input: None

Output: The double representing the symbiont's points

Purpose: To get a symbiont's points.

bool **IsPhage**()
Input: None

Output: The bool representing if a symbiont is a phage

Purpose: To determine if a symbiont is a phage

bool **IsHost**()
Input: None

Output: The bool representing if a symbiont is a host

Purpose: To determine if a symbiont is a host

double **GetInfectionChance**()
Input: None

Output: The chance of a symbiont infecting a parallel host during process

Purpose: To determine a symbiont's infection chance

emp::Ptr<*Organism*> **GetHost**()
Input: None

Output: The pointer to a symbiont's host

Purpose: To retrieve a symbiont's host

emp::Ptr<emp::Taxon<int>> **GetTaxon**()
Input: None

Output: The pointer to the symbiont's taxon

Purpose: To retrieve the symbiont's taxon

void **SetTaxon**(emp::Ptr<emp::Taxon<int>> *_in*)
Input: A pointer to the taxon that this organism should belong to.

Output: None

Purpose: To set the symbiont's taxon

void **SetDead**()
Input: None

Output: None

Purpose: To set a symbiont to dead

bool **GetDead**()
Input: None

Output: The bool representing if a symbiont is dead

Purpose: To determine if a symbiont is dead

void **SetIntVal**(double *_in*)
Input: The double representing the new interaction value of a symbiont

Output: None

Purpose: To set a symbiont's interaction value

void **SetPoints**(double *_in*)
Input: The double representing the points to be set as a symbinot's points

Output: None

Purpose: To set a symbiont's points

void **AddPoints**(double *_in*)
Input: The double representing the points to be added to a symbinot's points

Output: None

Purpose: To increment a symbiont's points

int **GetAge**()
Input: None

Output: an int representing the current age of the *Symbiont*

Purpose: To get the *Symbiont*'s age.

void **SetAge**(int *_in*)
Input: An int of what age the *Symbiont* should be set to

Output: None

Purpose: To set the *Symbiont*'s age for testing purposes.

void **SetHost**(emp::Ptr<*Organism*> *_in*)
Input: The pointer to an organism that will be set as the symbinot's host

Output: None

Purpose: To set a symbiont's host

void **SetInfectionChance**(double *_in*)

void **uponInjection**()
Input: None

Output: None

Purpose: Does nothing for now, added for backwards compatibility from phage to symbiont

void **GrowOlder**()
Input: None

Output: None

Purpose: Increments age by one and kills it if too old.

void **mutate**()
Input: None

Output: None

Purpose: To mutate a symbiont's interaction value. The mutation value is chosen from a normal distribution centered on 0 with the mutation size as the standard deviation.

double **ProcessResources**(double *hostDonation*)
Input: The double representing the resources to be distributed to the symbionts

Output: The double representing the host's resources

Purpose: To process and distribute resources.

double **ProcessResources**(double *hostDonation*, emp::Ptr<*Organism*> *host*)
Input: The double representing the resources to be distributed to the symbionts and the host from whom it comes

Output: The double representing the host's resources

Purpose: To process and distribute resources.

bool **WantsToInfect**()
Input: None

Output: The boolean representing if a symbiont will seek out to infect a host.

Purpose: To determine if a symbiont wants to infect a host based upon its infection chance

bool **InfectionFails**()
Input: None

Output: The boolean representing if a symbiont will survive crossing over into the host world.

Purpose: To determine if a symbiont will survive crossing over into the host world based on infection risk.

void **LoseResources**(double *resources*)
Input: The double representing the resources given by the world.

Output: None

Purpose: Free living symbionts specialized to interact with hosts (extreme interaction value in either direction) lose some of the resources that they get from the world.

void **Process**(emp::WorldPosition *location*)
Input: The size_t representing the location of the symbiont, and the size_t representation of the symbiont's position in the host (default -1 if it doesn't have a host)

Output: None

Purpose: To process a symbiont, meaning to check for reproduction, distribute resources, and to allow for movement

emp::Ptr<*Organism*> **makeNew**()
Input: None

Output: The pointer to the newly created organism

Purpose: To produce a new symbiont, identical to the original

emp::Ptr<*Organism*> **reproduce**()
Input: None

Output: The pointer to the newly created organism

Purpose: To produce a new symbiont; does not remove resources from the parent, assumes that is handled by calling function

void **VerticalTransmission**(emp::Ptr<*Organism*> *host_baby*)
Input: The pointer to the organism that is the new host baby

Output: None

Purpose: To allow for vertical transmission to occur

void **HorizontalTransmission**(emp::WorldPosition *location*)
Input: The location of the organism (and it's *Host*) as a size_t

Output: None

Purpose: To check and allow for horizontal transmission to occur

Protected Attributes

double **interaction_val** = 0
Purpose: Represents the interaction value between the host and symbiont. A negative interaction value represent antagonism, while a positive one represents mutualism. Zero is a neutral value.

double **points** = 0
Purpose: Represents the resource points possessed by a symbiont. This is what host's must collect to reproduce.

bool **dead** = false
Purpose: Represents if a symbiont is alive. This is set to true when a symbiont is killed.

double **infection_chance** = 0.0
Purpose: Represents the chance (between 0 and 1) that a free-living sym will infect a parallel host on process

int **age** = 0
Purpose: Represents the number of updates the symbiont has lived through; at birth is set to 0.

emp::Ptr<emp::Random> **random** = NULL
Purpose: Represents an instance of random.

emp::Ptr<*SymWorld*> **my_world** = NULL
Purpose: Represents the world that the hosts are living in.

emp::Ptr<*Organism*> **my_host** = NULL
Purpose: Represents the symbiont's host.

emp::Ptr<SymConfigBase> **my_config** = NULL
Purpose: Represents the configuration settings for a particular run.

emp::Ptr<emp::Taxon<int>> **my_taxon** = NULL
Purpose: Tracks the taxon of this organism.

Class SymWorld

- Defined in file_source_default_mode_SymWorld.h

Inheritance Relationships

Base Type

- `public emp::World< Organism >`

Derived Types

- `public EfficientWorld` (*Class EfficientWorld*)
- `public LysisWorld` (*Class LysisWorld*)
- `public PggWorld` (*Class PggWorld*)

Class Documentation

class SymWorld : **public** emp::World<*Organism*>
Subclassed by *EfficientWorld*, *LysisWorld*, *PggWorld*

Public Functions

SymWorld(emp::Random &*_random*)
Input: The world's random seed

Output: None

Purpose: To construct an instance of *SymWorld*

~SymWorld()
Input: None

Output: None

Purpose: To destruct the data nodes belonging to *SymWorld* to conserve memory.

void **SetVertTrans**(double *vt*)
Input: The double representing the vertical transmission rate

Output: None

Purpose: To set the vertical transmission rate

void **SetResPerUpdate**(double *val*)
Input: The double representing the number of resources each host gets in each update.

Output: None

Purpose: To set the resources that each host gets per update.

void **SetLimitedRes**(bool *val*)
Input: To boolean representing if resources are limited or not.

Output: None

Purpose: To allow for resources to be limited or unlimited.

void **SetFreeLivingSyms**(bool *flp*)
Input: The boolean representing if symbionts are allowed to be free living.

Output: None

Purpose: To allow for free-living symbionts

void **SetMoveFreeSyms**(bool *mfs*)
Input: The bool representing if free living symbionts are permitted to move around in the world.

Output: None

Purpose: To set the value representing if FLS are permitted to move around.

void **SetNumPhyloBins**(size_t *_in*)
Input: The size_t number of bins that organisms should be placed into when phylogeny tracking is on.

Output: None

Purpose: To set the number of bins used by phylogenies.

void **SetTrackPhylogeny**(bool *_in*)
Input: The bool representing whether phylogenies should be tracked.

Output: None

Purpose: To set the value representing whether phylogenies should be tracked.

void **SetTotalRes**(int *val*)
Input: The int representing the total number of resources for the world.

Output: None

Purpose: To set the total number of resources in the world. If limited resources is off, then the total resource value is of no consequence.

emp::World<*Organism*>::pop_t **GetPop**()
Input: None

Output: The pop_t value that represents the world's population.

Purpose: To get the world's population of organisms.

emp::World<*Organism*>::pop_t **GetSymPop**()
Input: None

Output: The pop_t value that represent the world's symbiont population.

Purpose: To get the world's symbiont population.

bool **WillTransmit**()
Input: None

Output: The boolean representing if vertical transmission will occur

Purpose: To determine if vertical transmission will occur

emp::Ptr<emp::Systematics<*Organism*, int>> **GetHostSys**()
Input: None

Output: The systematic object tracking hosts

Purpose: To retrieve the host systematic

emp::Ptr<emp::Systematics<*Organism*, int>> **GetSymSys**()
Input: None

Output: The systematic object tracking hosts

Purpose: To retrieve the symbiont systematic

fun_calc_info_t **GetCalcInfoFun**()
Input: None

Output: The standard function object that determines which bin organisms should belong to depending on their interaction value

Purpose: To classify organsims based on their interaction value.

emp::Ptr<emp::Taxon<int>> **AddSymToSystematic**(emp::Ptr<*Organism*> *sym*, emp::Ptr<emp::Taxon<int>> *parent_taxon* = nullptr)
Input: The symbiont to be added to the systematic

Output: the taxon the symbiont is added to.

Purpose: To add a symbiont to the systematic and to set it to track its taxon

int **PullResources**(int *desired_resources*)
Input: The amount of resourcces an organism wants from the world.

Output: If there are unlimited resources or the total resources are greater than those requested, returns the amount of desired resources. If total_res is less than the desired resources, but greater than 0, then total_res will be returned. If none of these are true, then 0 will be returned.

Purpose: To determine how many resources to distribute to each organism.

void **Resize**(size_t *new_width*, size_t *new_height*)
Input: The size_t representing the world's new width; the size_t representing the world's new height.

Output: None

Purpose: To overwrite the Empirical resize so that sym_pop is also resized

void **Resize**(size_t *new_size*)
Input: The size_t representing the new size of the world

Output: None

Purpose: To override the Empirical Resize function with a single-arg method that can be used for AddOrgAt vector expansions

void **AddOrgAt**(emp::Ptr<*Organism*> *new_org*, emp::WorldPosition *pos*, emp::WorldPosition *p_pos* = emp::WorldPosition())
Input: The pointer to the new organism; the world position of the location to add the new organism.

Output: None

Purpose: To overwrite the empirical AddOrgAt function to permit syms to be added into sym_pop

emp::WorldPosition **DoBirth** (emp::Ptr<*Organism*> *new_org*, emp::WorldPosition *p_pos*)
Input: (1) The pointer to the organism that is being birthed; (2) The size_t location of the parent organism.

Output: The WorldPosition of the position of the new organism.

Purpose: To introduce new organisms to the world.

int **GetNeighborHost** (size_t *i*)
Input: The size_t value representing the location whose neighbors are being searched.

Output: If there are no occupied neighboring positions, -1 will be returned. If there are occupied neighboring positions, then the location of one occupied position will be returned.

Purpose: To determine the location of a valid occupied neighboring position.

void **InjectSymbiont** (emp::Ptr<*Organism*> *new_sym*)
Input: The pointer to an organism that will be injected into a host.

Output: None

Purpose: To add a symbiont to a host's symbionts.

void **WritePhylogenyFile** (**const** std::string *&filename*)
Definitions of data node functions, expanded in DataNodes.h

Input: The address of the string representing the suffixes for the files to be created.

Output: None.

Purpose: To setup and write to the files that track the symbiont systematic information and the host systematic information

void **WriteDominantPhylogenyFiles** (**const** std::string *&filename*)

emp::Ptr<emp::Taxon<int>> **GetDominantSymTaxon** ()

emp::Ptr<emp::Taxon<int>> **GetDominantHostTaxon** ()

emp::vector<emp::Ptr<emp::Taxon<int>>> **GetDominantFreeHostedSymTaxon** ()

emp::DataFile &**SetupSymIntValFile** (**const** std::string *&filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track the average symbiont interaction value, the total number of symbionts, the total number of symbionts in a host, the total number of free syms and set up a histogram of the symbiont's interaction values.

emp::DataFile &**SetupHostIntValFile** (**const** std::string *&filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track host's interaction values, the total number of hosts, the total number of colony forming units, and the histogram of the host's interaction values

emp::DataFile &**SetUpFreeLivingSymFile** (**const** std::string *&filename*)
Input: The address of the string representing the file to be created's name

Output: The address of the DataFile that has been created.

Purpose: To set up the file that will be used to track mean information about the free living symbionts in the world. This includes: (1) their total count, (2) the counts of the free and hosted symbionts, (3) the

interaction values for the free and hosted symbionts, and (4) the infection chances from the total population, free symbionts, and hosted symbionts.

emp::DataMonitor<int> &**GetHostCountDataNode**()
Input: None

Output: The DataMonitor<int>& that has the information representing the host count.

Purpose: To collect data on the host count to be saved to the data file that is tracking host count

emp::DataMonitor<int> &**GetSymCountDataNode**()
Input: None

Output: The DataMonitor<int>& that has the information representing the symbiont count.

Purpose: To collect data on the symbiont count to be saved to the data file that is tracking symbiont count

emp::DataMonitor<int> &**GetCountHostedSymsDataNode**()
Input: None

Output: The DataMonitor<double>& that has the information representing the count of the hosted symbionts.

Purpose: To collect data on the count of the hosted symbionts to be saved to the data file that is tracking the count of the hosted symbionts.

emp::DataMonitor<int> &**GetCountFreeSymsDataNode**()
Input: None

Output: The DataMonitor<double>& that has the information representing the count of the free symbionts.

Purpose: To collect data on the count of the free symbionts to be saved to the data file that is tracking the count of the free symbionts.

emp::DataMonitor<int> &**GetUninfectedHostsDataNode**()
Input: None

Output: The DataMonitor<int>& that has the information representing the count of the uninfected hosts

Purpose: To collect data on the count of the uninfected hosts to be saved to the data file that is tracking the count of the uninfected hosts.

emp::DataMonitor<int> &**GetCFUDataNode**()
Input: None

Output: The DataMonitor<int>& that has the information representing the number of colony forming units.

Purpose: To collect data on the CFU count to be saved to the data file that is tracking CFU

emp::DataMonitor<double, emp::data::Histogram> &**GetHostIntValDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the host interaction value.

Purpose: To collect data on the host interaction value to be saved to the data file that is tracking host interaction value.

emp::DataMonitor<double, emp::data::Histogram> &**GetSymIntValDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the symbiont interaction value.

Purpose: To collect data on the symbiont interaction value to be saved to the data file that is tracking symbionts interaction value.

emp::DataMonitor<double, emp::data::Histogram> &**GetFreeSymIntValDataNode**()
Input: None

Output: The DataMonitor<double>& that has the information representing the free symbiont's interaction value.

Purpose: To collect data on the interaction value of the free symbionts to be saved to the data file that is tracking the interaction value of the free symbionts.

emp::DataMonitor<double, emp::data::Histogram> &**GetHostedSymIntValDataNode**()
Input:None

Output:

Purpose: To access the data node that is tracking the hosted symbiont interaction value

emp::DataMonitor<double, emp::data::Histogram> &**GetSymInfectChanceDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the infection chance for each symbionts.

Purpose: To access the data node that is tracking the symbiont infection chance

emp::DataMonitor<double, emp::data::Histogram> &**GetFreeSymInfectChanceDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the free symbionts' chance of infection

Purpose: To access the data node that is tracking the infection chance within the free symbionts.

emp::DataMonitor<double, emp::data::Histogram> &**GetHostedSymInfectChanceDataNode**()
Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has the information representing the infection chance for the hosted symbionts

Purpose: To retrieve the data nodes that is tracking the infection chance within the hosted symbionts.

void **MoveIntoNewFreeWorldPos**(emp::Ptr<*Organism*> *sym*, emp::WorldPosition *parent_pos*)
Input: The pointer to the symbiont that is moving, the WorldPosition of its current location.

Output: None

Purpose: To move a symbiont into a new world position.

void **SymDoBirth**(emp::Ptr<*Organism*> *sym_baby*, emp::WorldPosition *parent_pos*)
Input: The pointer to the organism that is being birthed, and the WorldPosition location of the parent symbiont.

Output: None

Purpose: To birth a new symbiont. If free living symbionts is on, the new symbiont can be put into an unoccupied place in the world. If not, then it will be placed in a host near its parent's location, or deleted if the parent's location has no eligible near-by hosts.

void **MoveFreeSym**(emp::WorldPosition *pos*)
Input: The WorldPosition location of the symbiont to be moved.

Output: None

Purpose: To move a symbiont, either into a host, or into a free world position

emp::Ptr<*Organism*> **GetSymAt** (size_t *location*)

emp::Ptr<*Organism*> **ExtractSym** (size_t *i*)
Input: The size_t representing the location of the symbiont to be extracted from the world.

Output: The pointer to the organism that was removed from the world

Purpose: To remove a symbiont from the world

void **DoSymDeath** (size_t *i*)
Input: The size_t representing the location of the symbiont to be deleted from the world.

Output: None

Purpose: To delete a symbiont from the world.

void **Update** ()
Input: None

Output: None

Purpose: To call the process functions for hosts and symbionts.

Protected Types

using fun_calc_info_t = std::function<int (*Organism*&)>

Protected Attributes

double **vertTrans** = 0
Purpose: Represents the vertical transmission rate. This can be set with *SetVertTrans()*

int **total_res** = -1
Purpose: Represents the total resources in the world. This can be set with *SetTotalRes()*

bool **limited_res** = false
Purpose: Represents if resources are limited or not. This can be set with *SetLimitedRes()*

bool **do_free_living_syms** = false
Purpose: Represents if free living symbionts are allowed. This can be set with *SetFreeLivingSyms()*

double **resources_per_host_per_update** = 0
Purpose: Represents how many resources each host gets per update. This can be set with *SetResPerUpdate()*

bool **move_free_syms** = false
Purpose: Represents if free living symbionts are permitted to move around the world.

bool **track_phylogeny** = false
Purpose: Represents if phylogeneis should be tracked. This can be set with *SetTrackPhylogeny()*

size_t **num_phylo_bins**
Purpose: Represents how many bins to place organisms into when tracking phylogenies.

pop_t **sym_pop**
Purpose: Represents the free living sym environment, parallel to "pop" for hosts

fun_calc_info_t **calc_info_fun**
Purpose: Represents a standard function object which determines which taxon an organism belongs to.

emp::Ptr<emp::Systematics<*Organism*, int>> **host_sys**
Purpose: Represents the systematics object tracking hosts.

emp::Ptr<emp::Systematics<*Organism*, int>> **sym_sys**
Purpose: Represents the systematics object tracking symbionts.

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_hostintval**

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_symintval**

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_freesymintval**

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_hostedsymintval**

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_syminfectchance**

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_freesyminfectchance**

emp::Ptr<emp::DataMonitor<double, emp::data::Histogram>> **data_node_hostedsyminfectchance**

emp::Ptr<emp::DataMonitor<int>> **data_node_hostcount**

emp::Ptr<emp::DataMonitor<int>> **data_node_symcount**

emp::Ptr<emp::DataMonitor<int>> **data_node_freesymcount**

emp::Ptr<emp::DataMonitor<int>> **data_node_hostedsymcount**

emp::Ptr<emp::DataMonitor<int>> **data_node_cfu**

emp::Ptr<emp::DataMonitor<int>> **data_node_uninf_hosts**

Enums

Enum Verbosity

- Defined in file_source_catch_catch.hpp

Enum Documentation

enum Catch::**Verbosity**
Values:

enumerator Quiet

enumerator Normal

enumerator High

Functions

Function Catch::cerr

- Defined in file_source_catch_catch.hpp

Function Documentation

std::ostream &Catch::**cerr**()

Function Catch::cleanUp

- Defined in file_source_catch_catch.hpp

Function Documentation

void Catch::**cleanUp**()

Function Catch::cleanUpContext

- Defined in file_source_catch_catch.hpp

Function Documentation

void Catch::**cleanUpContext**()

Function Catch::clog

- Defined in file_source_catch_catch.hpp

Function Documentation

std::ostream &Catch::**clog**()

Template Function Catch::compareEqual(LhsT const&, RhsT const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareEqual" with arguments (LhsT const&, RhsT const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareEqual(LhsT const &lhs, RhsT␣
↪const &rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareEqual(T *const&, int)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareEqual" with arguments (T *const&, int) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareEqual(LhsT const &lhs, RhsT␣
↪const &rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareEqual(T *const&, long)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareEqual" with arguments (T *const&, long) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareEqual(LhsT const &lhs, RhsT␣
↪const &rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareEqual(int, T *const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareEqual" with arguments (int, T *const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareEqual(LhsT const &lhs, RhsT␣
↪const &rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareEqual(long, T *const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareEqual" with arguments (long, T *const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareEqual(LhsT const &lhs, RhsT␣
↪const &rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareNotEqual(LhsT const&, RhsT&&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareNotEqual" with arguments (LhsT const&, RhsT&&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT␣
↪&&rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareNotEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareNotEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareNotEqual(T *const&, int)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareNotEqual" with arguments (T *const&, int) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT␣
↪&&rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareNotEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareNotEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareNotEqual(T *const&, long)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareNotEqual" with arguments (T *const&, long) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT␣
↪&&rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareNotEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareNotEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareNotEqual(int, T *const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareNotEqual" with arguments (int, T *const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT␣
↪&&rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareNotEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareNotEqual(long lhs, T *const &rhs) -> bool
```

Template Function Catch::compareNotEqual(long, T *const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::compareNotEqual" with arguments (long, T *const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT␣
↪&&rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, int rhs) -> bool
- template<typename T> auto compareNotEqual(T *const &lhs, long rhs) -> bool
- template<typename T> auto compareNotEqual(int lhs, T *const &rhs) -> bool
- template<typename T> auto compareNotEqual(long lhs, T *const &rhs) -> bool
```

Function Catch::contains

- Defined in file_source_catch_catch.hpp

Function Documentation

bool `Catch::contains` (std::string **const** &*s*, std::string **const** &*infix*)

Function Catch::cout

- Defined in file_source_catch_catch.hpp

Function Documentation

std::ostream &Catch::**cout**()

Template Function Catch::Detail::convertUnknownEnumToString

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **E**>
std::string Catch::Detail::**convertUnknownEnumToString**(*E* *e*)

Template Function Catch::Detail::convertUnstreamable(T const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
std::enable_if<!std::is_enum<*T*>::value && !std::is_base_of<std::exception, *T*>::value, std::string>::type Catch::Detail::**conve**

Template Function Catch::Detail::convertUnstreamable(T const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
std::enable_if<!std::is_enum<*T*>::value && !std::is_base_of<std::exception, *T*>::value, std::string>::type Catch::Detail::**conve**

Template Function Catch::Detail::convertUnstreamable(T const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
std::enable_if<!std::is_enum<*T*>::value && !std::is_base_of<std::exception, *T*>::value, std::string>::type `Catch::Detail::conve`

Template Function Catch::Detail::rangeToString

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **InputIterator**>
std::string `Catch::Detail::`**`rangeToString`**(*InputIterator* *first*, *InputIterator* *last*)

Function Catch::Detail::rawMemoryToString(const void *, std::size_t)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Detail::rawMemoryToString" with arguments (const void *, std::size_t) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- std::string rawMemoryToString(const void *object, std::size_t size)
- template<typename T> std::string rawMemoryToString(const T &object)
```

Template Function Catch::Detail::rawMemoryToString(const T&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Detail::rawMemoryToString" with arguments (const T&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- std::string rawMemoryToString(const void *object, std::size_t size)
- template<typename T> std::string rawMemoryToString(const T &object)
```

Template Function Catch::Detail::stringify

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
std::string `Catch::Detail::`**`stringify`**`(`**`const`** *T* *&e*`)`

Function Catch::endsWith(std::string const&, std::string const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::endsWith" with arguments (std::string const&, std::string const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- bool endsWith(std::string const &s, char suffix)
- bool endsWith(std::string const &s, std::string const &suffix)
```

Function Catch::endsWith(std::string const&, char)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::endsWith" with arguments (std::string const&, char) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- bool endsWith(std::string const &s, char suffix)
- bool endsWith(std::string const &s, std::string const &suffix)
```

Function Catch::filterTests

- Defined in file_source_catch_catch.hpp

Function Documentation

std::vector<*TestCase*> `Catch::`**`filterTests`**(std::vector<*TestCase*> **const** &*testCases*, TestSpec **const** &*testSpec*, *IConfig* **const** &*config*)

Function Catch::formatReconstructedExpression

- Defined in file_source_catch_catch.hpp

Function Documentation

void `Catch::`**`formatReconstructedExpression`**(std::ostream &*os*, std::string **const** &*lhs*, *StringRef* *op*, std::string **const** &*rhs*)

Function Catch::Generators::acquireGeneratorTracker

- Defined in file_source_catch_catch.hpp

Function Documentation

auto `Catch::Generators::`**`acquireGeneratorTracker`**(*SourceLineInfo* **const** &*lineInfo*) -> *IGeneratorTracker*&

Template Function Catch::Generators::chunk

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
GeneratorWrapper<std::vector<*T*>> `Catch::Generators::`**chunk**(size_t *size*, *GeneratorWrapper*<*T*> *&&generator*)

Template Function Catch::Generators::filter

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**, typename **Predicate**>
GeneratorWrapper<*T*> `Catch::Generators::`**filter**(*Predicate* *&&pred*, *GeneratorWrapper*<*T*> *&&generator*)

Template Function Catch::Generators::from_range(InputIterator, InputSentinel)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::from_range" with arguments (InputIterator, InputSentinel) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Container, typename ResultType = typename Container::value_type>
↪ GeneratorWrapper<ResultType> from_range(Container const &cnt)
- template<typename InputIterator, typename InputSentinel, typename ResultType =␣
↪typename std::iterator_traits<InputIterator>::value_type> GeneratorWrapper
↪<ResultType> from_range(InputIterator from, InputSentinel to)
```

Template Function Catch::Generators::from_range(Container const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::from_range" with arguments (Container const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Container, typename ResultType = typename Container::value_type>
↪ GeneratorWrapper<ResultType> from_range(Container const &cnt)
- template<typename InputIterator, typename InputSentinel, typename ResultType =␣
↪typename std::iterator_traits<InputIterator>::value_type> GeneratorWrapper
↪<ResultType> from_range(InputIterator from, InputSentinel to)
```

Template Function Catch::Generators::generate

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **L**>
auto `Catch::Generators::`**`generate`**(*SourceLineInfo* **const** *&lineInfo*, *L* **const** *&generatorExpression*) -> decltype(std::declval<decltype(*generatorExpression*())>().get())

Template Function Catch::Generators::makeGenerators(GeneratorWrapper<T>&&, Gs&&...)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::makeGenerators" with arguments (GeneratorWrapper<T>&&, Gs&&...) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T, typename ...Gs> auto makeGenerators(GeneratorWrapper<T> &&
↪generator, Gs&&... moreGenerators) -> Generators<T>
- template<typename T, typename ...Gs> auto makeGenerators(T &&val, Gs&&...␣
↪moreGenerators) -> Generators<T>
- template<typename T, typename U, typename ...Gs> auto makeGenerators(as<T>, U &&
↪val, Gs&&... moreGenerators) -> Generators<T>
- template<typename T> auto makeGenerators(GeneratorWrapper<T> &&generator) ->␣
↪Generators<T>
```

Template Function Catch::Generators::makeGenerators(GeneratorWrapper<T>&&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::makeGenerators" with arguments (GeneratorWrapper<T>&&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T, typename ...Gs> auto makeGenerators(GeneratorWrapper<T> &&
↪generator, Gs&&... moreGenerators) -> Generators<T>
- template<typename T, typename ...Gs> auto makeGenerators(T &&val, Gs&&...␣
↪moreGenerators) -> Generators<T>
```

```
- template<typename T, typename U, typename ...Gs> auto makeGenerators(as<T>, U &&
↪val, Gs&&... moreGenerators) -> Generators<T>
- template<typename T> auto makeGenerators(GeneratorWrapper<T> &&generator) ->␣
↪Generators<T>
```

Template Function Catch::Generators::makeGenerators(T&&, Gs&&...)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::makeGenerators" with arguments (T&&, Gs&&...) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T, typename ...Gs> auto makeGenerators(GeneratorWrapper<T> &&
↪generator, Gs&&... moreGenerators) -> Generators<T>
- template<typename T, typename ...Gs> auto makeGenerators(T &&val, Gs&&...␣
↪moreGenerators) -> Generators<T>
- template<typename T, typename U, typename ...Gs> auto makeGenerators(as<T>, U &&
↪val, Gs&&... moreGenerators) -> Generators<T>
- template<typename T> auto makeGenerators(GeneratorWrapper<T> &&generator) ->␣
↪Generators<T>
```

Template Function Catch::Generators::makeGenerators(as<T>, U&&, Gs&&...)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::makeGenerators" with arguments (as<T>, U&&, Gs&&...) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T, typename ...Gs> auto makeGenerators(GeneratorWrapper<T> &&
↪generator, Gs&&... moreGenerators) -> Generators<T>
- template<typename T, typename ...Gs> auto makeGenerators(T &&val, Gs&&...␣
↪moreGenerators) -> Generators<T>
- template<typename T, typename U, typename ...Gs> auto makeGenerators(as<T>, U &&
↪val, Gs&&... moreGenerators) -> Generators<T>
- template<typename T> auto makeGenerators(GeneratorWrapper<T> &&generator) ->␣
↪Generators<T>
```

Template Function Catch::Generators::map

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **Func**, typename **U**, typename **T** = *FunctionReturnType*<*Func*, *U*>>
GeneratorWrapper<*T*> Catch::Generators::**map**(*Func* &&*function*, *GeneratorWrapper*<*U*> &&*generator*)

Template Function Catch::Generators::pf::make_unique

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**, typename ...**Args**>
std::unique_ptr<*T*> Catch::Generators::pf::**make_unique**(*Args*&&... *args*)

Template Function Catch::Generators::random(T, T)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::random" with arguments (T, T) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T> std::enable_if<std::is_floating_point<T>::value,␣
↪GeneratorWrapper<T>>::type random(T a, T b)
- template<typename T> std::enable_if<std::is_integral<T>::value && !std::is_same<T,
↪ bool>::value, GeneratorWrapper<T>>::type random(T a, T b)
```

Template Function Catch::Generators::random(T, T)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::random" with arguments (T, T) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T> std::enable_if<std::is_floating_point<T>::value,␣
↪GeneratorWrapper<T>>::type random(T a, T b)
- template<typename T> std::enable_if<std::is_integral<T>::value && !std::is_same<T,
↪ bool>::value, GeneratorWrapper<T>>::type random(T a, T b)
```

Template Function Catch::Generators::range(T const&, T const&, T const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::range" with arguments (T const&, T const&, T const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T> GeneratorWrapper<T> range(T const &start, T const &end)
- template<typename T> GeneratorWrapper<T> range(T const &start, T const &end, T␣
↪const &step)
```

Template Function Catch::Generators::range(T const&, T const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Generators::range" with arguments (T const&, T const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T> GeneratorWrapper<T> range(T const &start, T const &end)
- template<typename T> GeneratorWrapper<T> range(T const &start, T const &end, T␣
↪const &step)
```

Template Function Catch::Generators::repeat

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
GeneratorWrapper<*T*> Catch::Generators::**repeat**(size_t *repeats*, *GeneratorWrapper*<*T*> &&*generator*)

Template Function Catch::Generators::table

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename ...**Ts**>
GeneratorWrapper<std::tuple<*Ts*...>> Catch::Generators::**table**(std::initializer_list<std::tuple<**typename** std::decay<*Ts*>::type...>> *tuples*)

Template Function Catch::Generators::take

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
GeneratorWrapper<*T*> Catch::Generators::**take**(size_t *target*, *GeneratorWrapper*<*T*> &&*generator*)

Template Function Catch::Generators::value

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
GeneratorWrapper<*T*> Catch::Generators::**value**(*T* &&*value*)

Template Function Catch::Generators::values

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
GeneratorWrapper<*T*> `Catch::Generators::`**`values`**(std::initializer_list<*T*> *values*)

Function Catch::getAllTestCasesSorted

- Defined in file_source_catch_catch.hpp

Function Documentation

std::vector<*TestCase*> **const** &`Catch::`**`getAllTestCasesSorted`**(*IConfig* **const** &*config*)

Function Catch::getCurrentContext

- Defined in file_source_catch_catch.hpp

Function Documentation

IContext &`Catch::`**`getCurrentContext`**()

Function Catch::getCurrentMutableContext

- Defined in file_source_catch_catch.hpp

Function Documentation

IMutableContext &`Catch::`**`getCurrentMutableContext`**()

Function Catch::getCurrentNanosecondsSinceEpoch

- Defined in file_source_catch_catch.hpp

Function Documentation

auto Catch::**getCurrentNanosecondsSinceEpoch**() -> uint64_t

Function Catch::getEstimatedClockResolution

- Defined in file_source_catch_catch.hpp

Function Documentation

auto Catch::**getEstimatedClockResolution**() -> uint64_t

Function Catch::getMutableRegistryHub

- Defined in file_source_catch_catch.hpp

Function Documentation

IMutableRegistryHub &Catch::**getMutableRegistryHub**()

Function Catch::getRegistryHub

- Defined in file_source_catch_catch.hpp

Function Documentation

IRegistryHub **const** &Catch::**getRegistryHub**()

Function Catch::getResultCapture

- Defined in file_source_catch_catch.hpp

Function Documentation

IResultCapture &Catch::**getResultCapture**()

Function Catch::handleExceptionMatchExpr(AssertionHandler&, std::string const&, StringRef const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::handleExceptionMatchExpr" with arguments (AssertionHandler&, std::string const&, StringRef const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- void handleExceptionMatchExpr(AssertionHandler &handler, StringMatcher const &
↪matcher, StringRef const &matcherString)
- void handleExceptionMatchExpr(AssertionHandler &handler, std::string const &str,␣
↪StringRef const &matcherString)
```

Function Catch::handleExceptionMatchExpr(AssertionHandler&, StringMatcher const&, StringRef const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::handleExceptionMatchExpr" with arguments (AssertionHandler&, StringMatcher const&, StringRef const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- void handleExceptionMatchExpr(AssertionHandler &handler, StringMatcher const &
↪matcher, StringRef const &matcherString)
- void handleExceptionMatchExpr(AssertionHandler &handler, std::string const &str,␣
↪StringRef const &matcherString)
```

Function Catch::handleExpression(ITransientExpression const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::handleExpression" with arguments (ITransientExpression const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T> void handleExpression(ExprLhs<T> const &expr)
- void handleExpression(ITransientExpression const &expr)
```

Template Function Catch::handleExpression(ExprLhs<T> const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::handleExpression" with arguments (ExprLhs<T> const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename T> void handleExpression(ExprLhs<T> const &expr)
- void handleExpression(ITransientExpression const &expr)
```

Function Catch::isFalseTest

- Defined in file_source_catch_catch.hpp

Function Documentation

bool `Catch::isFalseTest` (int *flags*)

Function Catch::isJustInfo

- Defined in file_source_catch_catch.hpp

Function Documentation

bool `Catch::isJustInfo` (int *flags*)

Function Catch::isOk

- Defined in file_source_catch_catch.hpp

Function Documentation

bool `Catch::isOk` (*ResultWas*::*OfType* *resultType*)

Function Catch::isThrowSafe

- Defined in file_source_catch_catch.hpp

Function Documentation

bool `Catch::isThrowSafe` (*TestCase* **const** &*testCase*, *IConfig* **const** &*config*)

Function Catch::literals::operator"" _a(long double)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::literals::operator"" _a" with arguments (long double) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Detail::Approx operator""_a(long double val)
- Detail::Approx operator""_a(unsigned long long val)
```

Function Catch::literals::operator"" _a(unsigned long long)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::literals::operator"" _a" with arguments (unsigned long long) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Detail::Approx operator""_a(long double val)
- Detail::Approx operator""_a(unsigned long long val)
```

Template Function Catch::makeMatchExpr

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **ArgT**, typename **MatcherT**>
auto `Catch::makeMatchExpr`(*ArgT* **const** &*arg*, *MatcherT* **const** &*matcher*, *StringRef* **const** &*matcherString*) -> *MatchExpr*<*ArgT*, *MatcherT*>

Function Catch::makeStream

- Defined in file_source_catch_catch.hpp

Function Documentation

auto `Catch::makeStream`(*StringRef* **const** &*filename*) -> *IStream* **const***

Function Catch::makeTestCase

- Defined in file_source_catch_catch.hpp

Function Documentation

TestCase `Catch::makeTestCase`(*ITestInvoker* **testCase*, std::string **const** &*className*, *NameAndTags* **const** &*nameAndTags*, *SourceLineInfo* **const** &*lineInfo*)

Function Catch::makeTestInvoker(void(*)())

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::makeTestInvoker" with arguments (void(*)()) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- auto makeTestInvoker(void (*testAsFunction)()) noexcept -> ITestInvoker*
- template<typename C> auto makeTestInvoker(void (C::* testAsMethod)()) noexcept ->␣
↪ITestInvoker*
```

Template Function Catch::makeTestInvoker(void(C::*)())

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::makeTestInvoker" with arguments (void(C::*)()) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- auto makeTestInvoker(void (*testAsFunction)()) noexcept -> ITestInvoker*
- template<typename C> auto makeTestInvoker(void (C::* testAsMethod)()) noexcept ->␣
↪ITestInvoker*
```

Template Function Catch::Matchers::Approx

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
Vector::*ApproxMatcher*<*T*> `Catch::Matchers::`**`Approx`**(std::vector<*T*> **const** &*comparator*)

Function Catch::Matchers::Contains(std::string const&, CaseSensitive::Choice)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::Contains" with arguments (std::string const&, CaseSensitive::Choice) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- StdString::ContainsMatcher Contains(std::string const &str, CaseSensitive::Choice␣
↪caseSensitivity = CaseSensitive::Yes)
- template<typename T> Vector::ContainsMatcher<T> Contains(std::vector<T> const &
↪comparator)
```

Template Function Catch::Matchers::Contains(std::vector<T> const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::Contains" with arguments (std::vector<T> const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- StdString::ContainsMatcher Contains(std::string const &str, CaseSensitive::Choice␣
↪caseSensitivity = CaseSensitive::Yes)
- template<typename T> Vector::ContainsMatcher<T> Contains(std::vector<T> const &
↪comparator)
```

Function Catch::Matchers::EndsWith

- Defined in file_source_catch_catch.hpp

Function Documentation

StdString::*EndsWithMatcher* Catch::Matchers::**EndsWith**(std::string **const** &*str*, *CaseSensitive*::*Choice* *caseSensitivity* = *CaseSensitive*::Yes)

Function Catch::Matchers::Equals(std::string const&, CaseSensitive::Choice)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::Equals" with arguments (std::string const&, CaseSensitive::Choice) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- StdString::EqualsMatcher Equals(std::string const &str, CaseSensitive::Choice␣
↪caseSensitivity = CaseSensitive::Yes)
- template<typename T> Vector::EqualsMatcher<T> Equals(std::vector<T> const &
↪comparator)
```

Template Function Catch::Matchers::Equals(std::vector<T> const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::Equals" with arguments (std::vector<T> const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- StdString::EqualsMatcher Equals(std::string const &str, CaseSensitive::Choice␣
↪caseSensitivity = CaseSensitive::Yes)
- template<typename T> Vector::EqualsMatcher<T> Equals(std::vector<T> const &
↪comparator)
```

Function Catch::Matchers::Generic::Detail::finalizeDescription

- Defined in file_source_catch_catch.hpp

Function Documentation

std::string Catch::Matchers::Generic::Detail::**finalizeDescription**(**const** std::string &*desc*)

Function Catch::Matchers::Matches

- Defined in file_source_catch_catch.hpp

Function Documentation

StdString::*RegexMatcher* Catch::Matchers::**Matches**(std::string **const** &*regex*, *CaseSensitive*::*Choice* *caseSensitivity* = *CaseSensitive*::Yes)

Function Catch::Matchers::Message

- Defined in file_source_catch_catch.hpp

Function Documentation

Exception::*ExceptionMessageMatcher* `Catch::Matchers::`**`Message`**(std::string **`const`** &*message*)

Template Function Catch::Matchers::Predicate

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **`T`**>
Generic::*PredicateMatcher*<*T*> `Catch::Matchers::`**`Predicate`**(std::function<bool) *T* **`const`**& > **`const`** &*predicate*, std::string **`const`** &*description* = ""

Function Catch::Matchers::StartsWith

- Defined in file_source_catch_catch.hpp

Function Documentation

StdString::*StartsWithMatcher* `Catch::Matchers::`**`StartsWith`**(std::string **`const`** &*str*, *CaseSensitive*::*Choice* *caseSensitivity* = *CaseSensitive*::Yes)

Template Function Catch::Matchers::UnorderedEquals

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **`T`**>
Vector::*UnorderedEqualsMatcher*<*T*> `Catch::Matchers::`**`UnorderedEquals`**(std::vector<*T*> **`const`** &*target*)

Template Function Catch::Matchers::VectorContains

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
Vector::*ContainsElementMatcher*<*T*> `Catch::Matchers::`**`VectorContains`**(*T* **const** &*comparator*)

Function Catch::Matchers::WithinAbs

- Defined in file_source_catch_catch.hpp

Function Documentation

Floating::*WithinAbsMatcher* `Catch::Matchers::`**`WithinAbs`**(double *target*, double *margin*)

Function Catch::Matchers::WithinRel(double, double)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::WithinRel" with arguments (double, double) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Floating::WithinRelMatcher WithinRel(double target)
- Floating::WithinRelMatcher WithinRel(double target, double eps)
- Floating::WithinRelMatcher WithinRel(float target)
- Floating::WithinRelMatcher WithinRel(float target, float eps)
```

Function Catch::Matchers::WithinRel(double)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::WithinRel" with arguments (double) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Floating::WithinRelMatcher WithinRel(double target)
- Floating::WithinRelMatcher WithinRel(double target, double eps)
- Floating::WithinRelMatcher WithinRel(float target)
- Floating::WithinRelMatcher WithinRel(float target, float eps)
```

Function Catch::Matchers::WithinRel(float, float)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::WithinRel" with arguments (float, float) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Floating::WithinRelMatcher WithinRel(double target)
- Floating::WithinRelMatcher WithinRel(double target, double eps)
- Floating::WithinRelMatcher WithinRel(float target)
- Floating::WithinRelMatcher WithinRel(float target, float eps)
```

Function Catch::Matchers::WithinRel(float)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::WithinRel" with arguments (float) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Floating::WithinRelMatcher WithinRel(double target)
- Floating::WithinRelMatcher WithinRel(double target, double eps)
- Floating::WithinRelMatcher WithinRel(float target)
- Floating::WithinRelMatcher WithinRel(float target, float eps)
```

Function Catch::Matchers::WithinULP(double, uint64_t)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::WithinULP" with arguments (double, uint64_t) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Floating::WithinUlpsMatcher WithinULP(double target, uint64_t maxUlpDiff)
- Floating::WithinUlpsMatcher WithinULP(float target, uint64_t maxUlpDiff)
```

Function Catch::Matchers::WithinULP(float, uint64_t)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::Matchers::WithinULP" with arguments (float, uint64_t) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- Floating::WithinUlpsMatcher WithinULP(double target, uint64_t maxUlpDiff)
- Floating::WithinUlpsMatcher WithinULP(float target, uint64_t maxUlpDiff)
```

Function Catch::matchTest

- Defined in file_source_catch_catch.hpp

Function Documentation

bool `Catch::matchTest`(*TestCase* **const** &*testCase*, TestSpec **const** &*testSpec*, *IConfig* **const** &*config*)

Function Catch::operator"" _sr

- Defined in file_source_catch_catch.hpp

Function Documentation

constexpr auto `Catch::operator""_sr`(char **const** **rawChars*, std::size_t *size*) **noexcept** -> *StringRef*

Template Function Catch::operator+

- Defined in file_source_catch_catch.hpp

Function Documentation

template<typename **T**>
T **const** &`Catch::operator+`(*T* **const** &*value*, *StreamEndStop*)

Function Catch::operator+=

- Defined in file_source_catch_catch.hpp

Function Documentation

auto `Catch::operator+=` (std::string &*lhs*, *StringRef* **const** &*sr*) -> std::string&

Function Catch::operator<<(std::ostream&, SourceLineInfo const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::operator<<" with arguments (std::ostream&, SourceLineInfo const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- auto operator<<(std::ostream &os, StringRef const &sr) -> std::ostream&
- std::ostream &operator<<(std::ostream &os, SourceLineInfo const &info)
```

Function Catch::operator<<(std::ostream&, StringRef const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::operator<<" with arguments (std::ostream&, StringRef const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- auto operator<<(std::ostream &os, StringRef const &sr) -> std::ostream&
- std::ostream &operator<<(std::ostream &os, SourceLineInfo const &info)
```

Function Catch::operator|

- Defined in file_source_catch_catch.hpp

Function Documentation

ResultDisposition::*Flags* `Catch::operator|` (*ResultDisposition*::*Flags* *lhs*, *ResultDisposition*::*Flags* *rhs*)

Template Function Catch::rangeToString(Range const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::rangeToString" with arguments (Range const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Allocator> std::string rangeToString(std::vector<bool,␣
↪Allocator> const &v)
- template<typename Range> std::string rangeToString(Range const &range)
```

Template Function Catch::rangeToString(std::vector<bool, Allocator> const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::rangeToString" with arguments (std::vector<bool, Allocator> const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename Allocator> std::string rangeToString(std::vector<bool,␣
↪Allocator> const &v)
- template<typename Range> std::string rangeToString(Range const &range)
```

Function Catch::replaceInPlace

- Defined in file_source_catch_catch.hpp

Function Documentation

bool Catch::**replaceInPlace**(std::string &*str*, std::string **const** &*replaceThis*, std::string **const** &*withThis*)

Function Catch::rng

- Defined in file_source_catch_catch.hpp

Function Documentation

SimplePcg32 &Catch::**rng**()

Function Catch::rngSeed

- Defined in file_source_catch_catch.hpp

Function Documentation

unsigned int Catch::**rngSeed**()

Function Catch::shouldContinueOnFailure

- Defined in file_source_catch_catch.hpp

Function Documentation

bool Catch::**shouldContinueOnFailure**(int *flags*)

Function Catch::shouldSuppressFailure

- Defined in file_source_catch_catch.hpp

Function Documentation

bool Catch::**shouldSuppressFailure**(int *flags*)

Function Catch::splitStringRef

- Defined in file_source_catch_catch.hpp

Function Documentation

std::vector<*StringRef*> `Catch::`**`splitStringRef`**(*StringRef* *str*, char *delimiter*)

Function Catch::startsWith(std::string const&, std::string const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::startsWith" with arguments (std::string const&, std::string const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- bool startsWith(std::string const &s, char prefix)
- bool startsWith(std::string const &s, std::string const &prefix)
```

Function Catch::startsWith(std::string const&, char)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::startsWith" with arguments (std::string const&, char) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- bool startsWith(std::string const &s, char prefix)
- bool startsWith(std::string const &s, std::string const &prefix)
```

Function Catch::throw_domain_error

- Defined in file_source_catch_catch.hpp

Function Documentation

void Catch::**throw_domain_error**(std::string **const** &*msg*)

Function Catch::throw_exception

- Defined in file_source_catch_catch.hpp

Function Documentation

void Catch::**throw_exception**(std::exception **const** &*e*)

Function Catch::throw_logic_error

- Defined in file_source_catch_catch.hpp

Function Documentation

void Catch::**throw_logic_error**(std::string **const** &*msg*)

Function Catch::throw_runtime_error

- Defined in file_source_catch_catch.hpp

Function Documentation

void Catch::**throw_runtime_error**(std::string **const** &*msg*)

Function Catch::toLower

- Defined in file_source_catch_catch.hpp

Function Documentation

std::string Catch::**toLower**(std::string **const** &*s*)

Function Catch::toLowerInPlace

- Defined in file_source_catch_catch.hpp

Function Documentation

void `Catch::toLowerInPlace`(std::string &*s*)

Function Catch::translateActiveException

- Defined in file_source_catch_catch.hpp

Function Documentation

std::string `Catch::translateActiveException`()

Function Catch::trim(std::string const&)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::trim" with arguments (std::string const&) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- StringRef trim(StringRef ref)
- std::string trim(std::string const &str)
```

Function Catch::trim(StringRef)

- Defined in file_source_catch_catch.hpp

Function Documentation

Warning: doxygenfunction: Unable to resolve multiple matches for function "Catch::trim" with arguments (StringRef) in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml. Potential matches:

```
- StringRef trim(StringRef ref)
- std::string trim(std::string const &str)
```

Function Cube

- Defined in file_source_catch_sanity_check.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “Cube” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function main(int, char *)

- Defined in file_source_native_symbulation_default.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function main(int, char *)

- Defined in file_source_native_symbulation_efficient.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function main(int, char *)

- Defined in file_source_native_symbulation_lysis.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function main(int, char *)

- Defined in file_source_native_symbulation_pgg.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function main(int, char *)

- Defined in file_source_web_symbulation-web.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function operator”” _catch_sr

- Defined in file_source_catch_catch.hpp

Function Documentation

constexpr auto **operator""_catch_sr**(char **const** *rawChars, std::size_t *size*) **noexcept** -> Catch::*StringRef*

Function operator<<

- Defined in file_source_catch_catch.hpp

Function Documentation

std::ostream &**operator<<**(std::ostream&, *Catch_global_namespace_dummy*)

Function symbulation_main(int, char *)

- Defined in file_source_native_symbulation_default.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “symbulation_main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function symbulation_main(int, char *)

- Defined in file_source_native_symbulation_efficient.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “symbulation_main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function symbulation_main(int, char *)

- Defined in file_source_native_symbulation_lysis.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “symbulation_main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function symbulation_main(int, char *)

- Defined in file_source_native_symbulation_pgg.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “symbulation_main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function symbulation_main(int, char *)

- Defined in file_source_web_symbulation-web.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “symbulation_main” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function Test

- Defined in file_source_test_end_to_end.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “Test” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE()

- Defined in file_source_catch_sanity_check.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Cubes are computed”)

- Defined in file_source_catch_sanity_check.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetHostCountDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetSymCountDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetCountHostedSymsDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetCountFreeSymsDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetUninfectedHostsDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetSymIntValDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetFreeSymIntValDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetHostedSymIntValDataNode”, “”)

- Defined in file_source_test_default_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetCFUDataNode”, “”)

- Defined in file_source_test_lysis_mode_test_DataNodes.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host Constructor”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host, GetIntVal”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“, AddPoints, GetPoints”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“, GetResTypes”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“HasSym”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host Mutate”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“DistributeResources”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“, GetResInProcess”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Steal resources unit test", "")

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("GetDoEctosymbiosis", "")

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host GrowOlder", "")

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host makeNew", "")

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host reproduce”, “”)

- Defined in file_source_test_default_mode_test_Host.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host-Symbiont interactions”, “”)

- Defined in file_source_test_default_mode_test_HostSymbiontInteraction.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont, GetHost”, “”)

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host SetSymbionts”, “”)

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host SymLimit", "")

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host AddSymbiont", "")

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host, ClearReproSym, GetReproSymbionts", "")

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host DistribResources", "")

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Vertical Transmission of Symbiont”, “”)

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“HandleEctosymbiosis”)

- Defined in file_source_test_default_mode_test_HostSymbiontUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont Constructor”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“, GetIntVal”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“, GetInfectionChance”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“, GetPoints”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont, GetDead”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“WantsToInfect”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“InfectionFails”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“mutate”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“reproduce”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Process”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont ProcessResources”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont GrowOlder”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont makeNew”, “”)

- Defined in file_source_test_default_mode_test_Symbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PullResources”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Vertical Transmission”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“World Capacity”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Interaction Patterns”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Hosts injected correctly”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“InjectSymbiont”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“DoBirth”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“SymDoBirth”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Update”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“MoveFreeSym”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“ExtractSym”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“MoveIntoNewFreeWorldPos”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Resize”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“AddOrgAt”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“GetSymAt”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“DoSymDeath”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Host Phylogeny”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Symbiont Phylogeny”, “”)

- Defined in file_source_test_default_mode_test_SymWorld.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientHost Constructor”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientHost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientHost makeNew”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientHost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientHost SetEfficiency and GetEfficiency”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientHost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont Constructor”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont mutate”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont AddPoints”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“INT_VAL_MUT_RATE”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont reproduce”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont HorizMutate”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont mutate with horizontal transmission”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont mutate with vertical transmission”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont’s Process called from Host when mutation rate and size are zero”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont makeNew”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“EfficientSymbiont SetEfficiency and GetEfficiency”, “”)

- Defined in file_source_test_efficient_mode_test_EfficientSymbiont.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“End To End”)

- Defined in file_source_test_end_to_end.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Bacterium, host_incorporation_val”, “”)

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Bacterium, GetIncVal”, “”)

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Bacterium mutate”, “”)

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“ProcessLysogenResources”, “”)

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Bacterium Process", "")

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage Exclude", "")

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Bacterium makeNew", "")

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Bacterium reproduce", "")

- Defined in file_source_test_lysis_mode_test_Bacterium.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage Process", "")

- Defined in file_source_test_lysis_mode_test_BacteriumPhageUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage Vertical Transmission", "")

- Defined in file_source_test_lysis_mode_test_BacteriumPhageUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Host phage death and removal from syms list", "")

- Defined in file_source_test_lysis_mode_test_BacteriumPhageUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage LysisBurst", "")

- Defined in file_source_test_lysis_mode_test_BacteriumPhageUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage LysisStep", "")

- Defined in file_source_test_lysis_mode_test_BacteriumPhageUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage, GetIntVal", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage reproduce", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE(", IncBurstTimer", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage, GetLysisChance", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage, GetInductionChance", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage, GetIncVal", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Phage uponInjection", "")

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“phage_mutate”, “”)

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Phage process”, “”)

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Phage ProcessResources”, “”)

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Phage makeNew”, “”)

- Defined in file_source_test_lysis_mode_test_Phage.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PggHost constructor”, “”)

- Defined in file_source_test_pgg_mode_test_Pgghost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PggHost get pool”, “”)

- Defined in file_source_test_pgg_mode_test_Pgghost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“Pgghost DistributeResources”, “”)

- Defined in file_source_test_pgg_mode_test_Pgghost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PggHost makeNew”, “”)

- Defined in file_source_test_pgg_mode_test_Pgghost.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PggSymbiont, GetHost", "")

- Defined in file_source_test_pgg_mode_test_PggHostPggSymUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PGGHost DistribResources", "")

- Defined in file_source_test_pgg_mode_test_PggHostPggSymUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PGGVertical Transmission of Symbiont", "")

- Defined in file_source_test_pgg_mode_test_PggHostPggSymUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PGGSymbiont PGGHost Pool Interaction", "")

- Defined in file_source_test_pgg_mode_test_PggHostPggSymUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PGGSYM Dead and Removal", "")

- Defined in file_source_test_pgg_mode_test_PggHostPggSymUnitTest.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PGGSymbiont Constructor", "")

- Defined in file_source_test_pgg_mode_test_Pggsym.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("Pggmutate", "")

- Defined in file_source_test_pgg_mode_test_Pggsym.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "TEST_CASE" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function TEST_CASE("PGGSymbiont ProcessPool", "")

- Defined in file_source_test_pgg_mode_test_Pggsym.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PGGProcess”, “”)

- Defined in file_source_test_pgg_mode_test_Pggsym.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PGGSymbiont ProcessResources”, “”)

- Defined in file_source_test_pgg_mode_test_Pggsym.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function TEST_CASE(“PggSymbiont makeNew”, “”)

- Defined in file_source_test_pgg_mode_test_Pggsym.test.cc

Function Documentation

Warning: doxygenfunction: Cannot find function “TEST_CASE” in doxygen xml output for project “Symbulation” from directory: ./doxyoutput/xml

Function worldSetup(emp::Ptr<SymWorld>, emp::Ptr<SymConfigBase>)

- Defined in file_source_default_mode_WorldSetup.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "worldSetup" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function worldSetup(emp::Ptr<EfficientWorld>, emp::Ptr<SymConfigBase>)

- Defined in file_source_efficient_mode_EfficientWorldSetup.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "worldSetup" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function worldSetup(emp::Ptr<LysisWorld>, emp::Ptr<SymConfigBase>)

- Defined in file_source_lysis_mode_LysisWorldSetup.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "worldSetup" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Function worldSetup(emp::Ptr<PggWorld>, emp::Ptr<SymConfigBase>)

- Defined in file_source_pgg_mode_PggWorldSetup.cc

Function Documentation

Warning: doxygenfunction: Cannot find function "worldSetup" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Variables

Variable anim

- Defined in file_source_web_symbulation-web.cc

Variable Documentation

Warning: doxygenvariable: Cannot find variable "anim" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Variable Catch::begin

- Defined in file_source_catch_catch.hpp

Variable Documentation

not_this_one `Catch::begin(...)`

Variable Catch::Detail::unprintableString

- Defined in file_source_catch_catch.hpp

Variable Documentation

const std::string `Catch::Detail::unprintableString`

Variable Catch::end

- Defined in file_source_catch_catch.hpp

Variable Documentation

not_this_one `Catch::end(...)`

Variable config

- Defined in file_source_SymAnimate.h

Variable Documentation

SymConfigBase **config**

Defines

Define AND_GIVEN

- Defined in file_source_catch_catch.hpp

Define Documentation

AND_GIVEN (*desc*)

Define AND_THEN

- Defined in file_source_catch_catch.hpp

Define Documentation

AND_THEN (*desc*)

Define AND_WHEN

- Defined in file_source_catch_catch.hpp

Define Documentation

AND_WHEN (*desc*)

Define ANON_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

ANON_TEST_CASE ( )

Define CAPTURE

- Defined in file_source_catch_catch.hpp

Define Documentation

CAPTURE(...)

Define CATCH_CATCH_ALL

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CATCH_ALL

Define CATCH_CATCH_ANON

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CATCH_ANON(*type*)

Define CATCH_CONFIG_COUNTER

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CONFIG_COUNTER

Define CATCH_CONFIG_CPP11_TO_STRING

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CONFIG_CPP11_TO_STRING

Define CATCH_CONFIG_DISABLE_EXCEPTIONS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CONFIG_DISABLE_EXCEPTIONS

Define CATCH_CONFIG_GLOBAL_NEXTAFTER

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CONFIG_GLOBAL_NEXTAFTER

Define CATCH_CONFIG_MAIN

- Defined in file_source_catch_main.cc

Define Documentation

Warning: doxygendefine: Cannot find define "CATCH_CONFIG_MAIN" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Define CATCH_CONFIG_POSIX_SIGNALS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CONFIG_POSIX_SIGNALS

Define CATCH_CONFIG_WCHAR

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_CONFIG_WCHAR

Define CATCH_DEFER

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_DEFER(*id*)

Define CATCH_EMPTY

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_EMPTY()

Define CATCH_ENFORCE

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_ENFORCE(*condition*, ...)

Define CATCH_ERROR

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_ERROR(...)

Define CATCH_INTERNAL_CONFIG_COUNTER

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_CONFIG_COUNTER

Define CATCH_INTERNAL_CONFIG_GLOBAL_NEXTAFTER

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_CONFIG_GLOBAL_NEXTAFTER

Define CATCH_INTERNAL_CONFIG_POSIX_SIGNALS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_CONFIG_POSIX_SIGNALS

Define CATCH_INTERNAL_ERROR

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_ERROR(...)

Define CATCH_INTERNAL_IGNORE_BUT_WARN

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_IGNORE_BUT_WARN(...)

Define CATCH_INTERNAL_LINEINFO

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_LINEINFO

Define CATCH_INTERNAL_START_WARNINGS_SUPPRESSION

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_START_WARNINGS_SUPPRESSION

Define CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION

Define CATCH_INTERNAL_STRINGIFY

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_STRINGIFY(...)

Define CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGS

Define CATCH_INTERNAL_SUPPRESS_PARENTHESES_WARNINGS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_SUPPRESS_PARENTHESES_WARNINGS

Define CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS

Define CATCH_INTERNAL_SUPPRESS_UNUSED_WARNINGS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_SUPPRESS_UNUSED_WARNINGS

Define CATCH_INTERNAL_SUPPRESS_ZERO_VARIADIC_WARNINGS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_INTERNAL_SUPPRESS_ZERO_VARIADIC_WARNINGS

Define CATCH_MAKE_MSG

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_MAKE_MSG(...)

Define CATCH_REC_END

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_END(...)

Define CATCH_REC_GET_END

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_GET_END(...)

Define CATCH_REC_GET_END1

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_GET_END1(...)

Define CATCH_REC_GET_END2

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_GET_END2 ( )

Define CATCH_REC_LIST

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST (*f*, ...)

Define CATCH_REC_LIST0

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST0 (*f*, *x*, *peek*, ...)

Define CATCH_REC_LIST0_UD

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST0_UD (*f*, *userdata*, *x*, *peek*, ...)

Define CATCH_REC_LIST1

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST1 (*f*, *x*, *peek*, ...)

Define CATCH_REC_LIST1_UD

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST1_UD (*f*, *userdata*, *x*, *peek*, ...)

Define CATCH_REC_LIST2

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST2 (*f*, *x*, *peek*, ...)

Define CATCH_REC_LIST2_UD

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST2_UD (*f*, *userdata*, *x*, *peek*, ...)

Define CATCH_REC_LIST_UD

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_LIST_UD (*f*, *userdata*, ...)

Define CATCH_REC_NEXT

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_NEXT (*test*, *next*)

Define CATCH_REC_NEXT0

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_NEXT0 (*test*, *next*, ...)

Define CATCH_REC_NEXT1

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_NEXT1 (*test*, *next*)

Define CATCH_REC_OUT

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REC_OUT

Define CATCH_RECURSE

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSE(...)

Define CATCH_RECURSION_LEVEL0

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSION_LEVEL0(...)

Define CATCH_RECURSION_LEVEL1

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSION_LEVEL1(...)

Define CATCH_RECURSION_LEVEL2

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSION_LEVEL2(...)

Define CATCH_RECURSION_LEVEL3

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSION_LEVEL3(...)

Define CATCH_RECURSION_LEVEL4

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSION_LEVEL4(...)

Define CATCH_RECURSION_LEVEL5

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RECURSION_LEVEL5(...)

Define CATCH_REGISTER_ENUM

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REGISTER_ENUM(*enumName*, ...)

Define CATCH_REGISTER_TAG_ALIAS

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_REGISTER_TAG_ALIAS(*alias*, *spec*)

Define CATCH_RUNTIME_ERROR

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_RUNTIME_ERROR(...)

Define CATCH_TRANSLATE_EXCEPTION

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_TRANSLATE_EXCEPTION(*signature*)

Define CATCH_TRY

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_TRY

Define CATCH_VERSION_MAJOR

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_VERSION_MAJOR

Define CATCH_VERSION_MINOR

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_VERSION_MINOR

Define CATCH_VERSION_PATCH

- Defined in file_source_catch_catch.hpp

Define Documentation

CATCH_VERSION_PATCH

Define CHECK

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK(...)

Define CHECK_FALSE

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_FALSE(...)

Define CHECK_NOFAIL

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_NOFAIL(...)

Define CHECK_NOTHROW

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_NOTHROW(...)

Define CHECK_THAT

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_THAT(*arg*, *matcher*)

Define CHECK_THROWS

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_THROWS(...)

Define CHECK_THROWS_AS

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_THROWS_AS(*expr*, *exceptionType*)

Define CHECK_THROWS_MATCHES

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_THROWS_MATCHES(*expr*, *exceptionType*, *matcher*)

Define CHECK_THROWS_WITH

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECK_THROWS_WITH(*expr*, *matcher*)

Define CHECKED_ELSE

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECKED_ELSE(...)

Define CHECKED_IF

- Defined in file_source_catch_catch.hpp

Define Documentation

CHECKED_IF(...)

Define DYNAMIC_SECTION

- Defined in file_source_catch_catch.hpp

Define Documentation

DYNAMIC_SECTION(...)

Define EFFWORLD_SETUP_C

- Defined in file_source_efficient_mode_EfficientWorldSetup.cc

Define Documentation

Warning: doxygendefine: Cannot find define "EFFWORLD_SETUP_C" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Define FAIL

- Defined in file_source_catch_catch.hpp

Define Documentation

FAIL(...)

Define FAIL_CHECK

- Defined in file_source_catch_catch.hpp

Define Documentation

FAIL_CHECK(...)

Define GENERATE

- Defined in file_source_catch_catch.hpp

Define Documentation

GENERATE(...)

Define GENERATE_COPY

- Defined in file_source_catch_catch.hpp

Define Documentation

GENERATE_COPY(...)

Define GENERATE_REF

- Defined in file_source_catch_catch.hpp

Define Documentation

GENERATE_REF(...)

Define GIVEN

- Defined in file_source_catch_catch.hpp

Define Documentation

GIVEN(*desc*)

Define INFO

- Defined in file_source_catch_catch.hpp

Define Documentation

INFO(*msg*)

Define INTERNAL_CATCH_CAPTURE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_CAPTURE(*varName*, *macroName*, ...)

Define INTERNAL_CATCH_CATCH

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_CATCH (*capturer*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST (*TestName*, *...*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST0 (*TestName*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST1

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST1 (*TestName*, *signature*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD (*TestName*, *ClassName*, *...*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD0 (*TestName*, *ClassName*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD1

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD1 (*TestName*, *ClassName*, *signature*)

Define INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD_X

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD_X (*TestName*, *ClassName*, *signature*, ...)

Define INTERNAL_CATCH_DECLARE_SIG_TEST_X

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DECLARE_SIG_TEST_X (*TestName*, *signature*, ...)

Define INTERNAL_CATCH_DEF

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEF(...)

Define INTERNAL_CATCH_DEFINE_SIG_TEST

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST(*TestName*, ...)

Define INTERNAL_CATCH_DEFINE_SIG_TEST0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST0(*TestName*)

Define INTERNAL_CATCH_DEFINE_SIG_TEST1

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST1(*TestName*, *signature*)

Define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD(*TestName*, ...)

Define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD0 (*TestName*)

Define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD1

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD1 (*TestName*, *signature*)

Define INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X (*TestName*, *signature*, ...)

Define INTERNAL_CATCH_DEFINE_SIG_TEST_X

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DEFINE_SIG_TEST_X (*TestName*, *signature*, ...)

Define INTERNAL_CATCH_DYNAMIC_SECTION

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_DYNAMIC_SECTION(...)

Define INTERNAL_CATCH_ELSE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_ELSE(*macroName*, *resultDisposition*, ...)

Define INTERNAL_CATCH_EXPAND1

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_EXPAND1(*param*)

Define INTERNAL_CATCH_EXPAND2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_EXPAND2(...)

Define INTERNAL_CATCH_IF

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_IF(*macroName*, *resultDisposition*, ...)

Define INTERNAL_CATCH_INFO

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_INFO(*macroName*, *log*)

Define INTERNAL_CATCH_MAKE_NAMESPACE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_MAKE_NAMESPACE(*name*)

Define INTERNAL_CATCH_MAKE_NAMESPACE2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_MAKE_NAMESPACE2(...)

Define INTERNAL_CATCH_MAKE_TYPE_LIST

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_MAKE_TYPE_LIST(...)

Define INTERNAL_CATCH_MAKE_TYPE_LIST2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_MAKE_TYPE_LIST2(...)

Define INTERNAL_CATCH_MAKE_TYPE_LISTS_FROM_TYPES

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_MAKE_TYPE_LISTS_FROM_TYPES(...)

Define INTERNAL_CATCH_METHOD_AS_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_METHOD_AS_TEST_CASE(*QualifiedMethod*, ...)

Define INTERNAL_CATCH_MSG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_MSG(*macroName*, *messageType*, *resultDisposition*, ...)

Define INTERNAL_CATCH_NO_THROW

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NO_THROW(*macroName*, *resultDisposition*, ...)

Define INTERNAL_CATCH_NOINTERNAL_CATCH_DEF

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NOINTERNAL_CATCH_DEF

Define INTERNAL_CATCH_NTTP_0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_0

Define INTERNAL_CATCH_NTTP_1

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_1(*signature*, ...)

Define INTERNAL_CATCH_NTTP_GEN

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_GEN(...)

Define INTERNAL_CATCH_NTTP_REG_GEN

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_REG_GEN(*TestFunc*, ...)

Define INTERNAL_CATCH_NTTP_REG_METHOD_GEN

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_REG_METHOD_GEN(*TestName*, ...)

Define INTERNAL_CATCH_NTTP_REGISTER

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_REGISTER(*TestFunc*, *signature*, ...)

Define INTERNAL_CATCH_NTTP_REGISTER0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_REGISTER0(*TestFunc*, *signature*)

Define INTERNAL_CATCH_NTTP_REGISTER_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_REGISTER_METHOD(*TestName*, *signature*, ...)

Define INTERNAL_CATCH_NTTP_REGISTER_METHOD0

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_NTTP_REGISTER_METHOD0(*TestName*, *signature*, ...)

Define INTERNAL_CATCH_REACT

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REACT(*handler*)

Define INTERNAL_CATCH_REGISTER_ENUM

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REGISTER_ENUM(*enumName*, ...)

Define INTERNAL_CATCH_REGISTER_TESTCASE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REGISTER_TESTCASE(*Function*, ...)

Define INTERNAL_CATCH_REMOVE_PARENS

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS(...)

Define INTERNAL_CATCH_REMOVE_PARENS_10_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_10_ARG(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*, *_6*, *_7*, *_8*, *_9*)

Define INTERNAL_CATCH_REMOVE_PARENS_11_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_11_ARG(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*, *_6*, *_7*, *_8*, *_9*, *_10*)

Define INTERNAL_CATCH_REMOVE_PARENS_1_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_1_ARG(*_0*)

Define INTERNAL_CATCH_REMOVE_PARENS_2_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_2_ARG(*_0*, *_1*)

Define INTERNAL_CATCH_REMOVE_PARENS_3_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_3_ARG(*_0*, *_1*, *_2*)

Define INTERNAL_CATCH_REMOVE_PARENS_4_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_4_ARG(*_0*, *_1*, *_2*, *_3*)

Define INTERNAL_CATCH_REMOVE_PARENS_5_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_5_ARG(*_0*, *_1*, *_2*, *_3*, *_4*)

Define INTERNAL_CATCH_REMOVE_PARENS_6_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_6_ARG(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*)

Define INTERNAL_CATCH_REMOVE_PARENS_7_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_7_ARG(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*, *_6*)

Define INTERNAL_CATCH_REMOVE_PARENS_8_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_8_ARG(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*, *_6*, *_7*)

Define INTERNAL_CATCH_REMOVE_PARENS_9_ARG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_9_ARG(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*, *_6*, *_7*, *_8*)

Define INTERNAL_CATCH_REMOVE_PARENS_GEN

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_REMOVE_PARENS_GEN(...)

Define INTERNAL_CATCH_SECTION

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_SECTION(...)

Define INTERNAL_CATCH_STRINGIZE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_STRINGIZE(...)

Define INTERNAL_CATCH_STRINGIZE2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_STRINGIZE2(...)

Define INTERNAL_CATCH_STRINGIZE_WITHOUT_PARENS

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_STRINGIZE_WITHOUT_PARENS(*param*)

Define INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE(*Name*, *Tags*, *TmplList*)

Define INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_2 (*TestName*, *TestFunc*, *Name*, *Tags*, *TmplList*)

Define INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD (*ClassName*, *Name*, *Tags*, *TmplList*)

Define INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD_2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD_2 (*TestNameClass*, *TestName*, *ClassName*, *Name*, *Tags*, *TmplList*)

Define INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE (*Name*, *Tags*, ...)

Define INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE2 (*TestName*, *TestFuncName*, *Name*, *Tags*, *Signature*, *TmplTypes*, *TypesList*)

Define INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_METHOD(*ClassName*, *Name*, *Tags*, ...)

Define INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_METHOD_2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_METHOD_2(*TestNameClass*, *TestName*, *ClassName*, *Name*, *Tags*, *Signature*, *TmplTypes*, *TypesList*)

Define INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(*ClassName*, *Name*, *Tags*, *Signature*, ...)

Define INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_SIG(*Name*, *Tags*, *Signature*, ...)

Define INTERNAL_CATCH_TEMPLATE_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_TEST_CASE (*Name*, *Tags*, ...)

Define INTERNAL_CATCH_TEMPLATE_TEST_CASE_2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_TEST_CASE_2 (*TestName*, *TestFunc*, *Name*, *Tags*, *Signature*, ...)

Define INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD (*ClassName*, *Name*, *Tags*, ...)

Define INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD_2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD_2 (*TestNameClass*, *TestName*, *ClassName*, *Name*, *Tags*, *Signature*, ...)

Define INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_TEST_CASE_METHOD_SIG (*ClassName*, *Name*, *Tags*, *Signature*, ...)

Define INTERNAL_CATCH_TEMPLATE_TEST_CASE_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEMPLATE_TEST_CASE_SIG (*Name*, *Tags*, *Signature*, ...)

Define INTERNAL_CATCH_TEST

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEST (*macroName*, *resultDisposition*, ...)

Define INTERNAL_CATCH_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEST_CASE_METHOD (*ClassName*, ...)

Define INTERNAL_CATCH_TEST_CASE_METHOD2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TEST_CASE_METHOD2 (*TestName*, *ClassName*, ...)

Define INTERNAL_CATCH_TESTCASE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TESTCASE(...)

Define INTERNAL_CATCH_TESTCASE2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TESTCASE2(*TestName*, ...)

Define INTERNAL_CATCH_THROWS

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_THROWS(*macroName*, *resultDisposition*, ...)

Define INTERNAL_CATCH_THROWS_AS

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_THROWS_AS(*macroName*, *exceptionType*, *resultDisposition*, *expr*)

Define INTERNAL_CATCH_THROWS_MATCHES

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_THROWS_MATCHES(*macroName*, *exceptionType*, *resultDisposition*, *matcher*, ...)

Define INTERNAL_CATCH_THROWS_STR_MATCHES

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_THROWS_STR_MATCHES(*macroName*, *resultDisposition*, *matcher*, ...)

Define INTERNAL_CATCH_TRANSLATE_EXCEPTION

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TRANSLATE_EXCEPTION(*signature*)

Define INTERNAL_CATCH_TRANSLATE_EXCEPTION2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TRANSLATE_EXCEPTION2(*translatorName*, *signature*)

Define INTERNAL_CATCH_TRY

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TRY

Define INTERNAL_CATCH_TYPE_GEN

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_TYPE_GEN

Define INTERNAL_CATCH_UNIQUE_NAME

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_UNIQUE_NAME(*name*)

Define INTERNAL_CATCH_UNIQUE_NAME_LINE

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_UNIQUE_NAME_LINE(*name*, *line*)

Define INTERNAL_CATCH_UNIQUE_NAME_LINE2

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_UNIQUE_NAME_LINE2(*name*, *line*)

Define INTERNAL_CATCH_UNSCOPED_INFO

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_UNSCOPED_INFO(*macroName*, *log*)

Define INTERNAL_CATCH_VA_NARGS_IMPL

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CATCH_VA_NARGS_IMPL(*_0*, *_1*, *_2*, *_3*, *_4*, *_5*, *_6*, *_7*, *_8*, *_9*, *_10*, *N*, *...*)

Define INTERNAL_CHECK_THAT

- Defined in file_source_catch_catch.hpp

Define Documentation

INTERNAL_CHECK_THAT(*macroName*, *matcher*, *resultDisposition*, *arg*)

Define LYSIS_WORLD_SETUP_C

- Defined in file_source_lysis_mode_LysisWorldSetup.cc

Define Documentation

Warning: doxygendefine: Cannot find define "LYSIS_WORLD_SETUP_C" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Define METHOD_AS_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

METHOD_AS_TEST_CASE(*method*, *...*)

Define PGG_WORLD_SETUP_C

- Defined in file_source_pgg_mode_PggWorldSetup.cc

Define Documentation

Warning: doxygendefine: Cannot find define "PGG_WORLD_SETUP_C" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Define REGISTER_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

REGISTER_TEST_CASE(*Function*, ...)

Define REQUIRE

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE(...)

Define REQUIRE_FALSE

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_FALSE(...)

Define REQUIRE_NOTHROW

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_NOTHROW(...)

Define REQUIRE_THAT

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_THAT(*arg*, *matcher*)

Define REQUIRE_THROWS

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_THROWS(...)

Define REQUIRE_THROWS_AS

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_THROWS_AS(*expr*, *exceptionType*)

Define REQUIRE_THROWS_MATCHES

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_THROWS_MATCHES(*expr*, *exceptionType*, *matcher*)

Define REQUIRE_THROWS_WITH

- Defined in file_source_catch_catch.hpp

Define Documentation

REQUIRE_THROWS_WITH(*expr*, *matcher*)

Define SCENARIO

- Defined in file_source_catch_catch.hpp

Define Documentation

SCENARIO(...)

Define SCENARIO_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

SCENARIO_METHOD(*className*, ...)

Define SECTION

- Defined in file_source_catch_catch.hpp

Define Documentation

SECTION(...)

Define STATIC_REQUIRE

- Defined in file_source_catch_catch.hpp

Define Documentation

STATIC_REQUIRE(...)

Define STATIC_REQUIRE_FALSE

- Defined in file_source_catch_catch.hpp

Define Documentation

STATIC_REQUIRE_FALSE(...)

Define SUCCEED

- Defined in file_source_catch_catch.hpp

Define Documentation

SUCCEED(...)

Define TEMPLATE_LIST_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_LIST_TEST_CASE(...)

Define TEMPLATE_LIST_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_LIST_TEST_CASE_METHOD(*className*, ...)

Define TEMPLATE_PRODUCT_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_PRODUCT_TEST_CASE(...)

Define TEMPLATE_PRODUCT_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_PRODUCT_TEST_CASE_METHOD(*className*, ...)

Define TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_PRODUCT_TEST_CASE_METHOD_SIG(*className*, ...)

Define TEMPLATE_PRODUCT_TEST_CASE_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_PRODUCT_TEST_CASE_SIG(...)

Define TEMPLATE_TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_TEST_CASE(...)

Define TEMPLATE_TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_TEST_CASE_METHOD (*className*, ...)

Define TEMPLATE_TEST_CASE_METHOD_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_TEST_CASE_METHOD_SIG (*className*, ...)

Define TEMPLATE_TEST_CASE_SIG

- Defined in file_source_catch_catch.hpp

Define Documentation

TEMPLATE_TEST_CASE_SIG (...)

Define TEST_CASE

- Defined in file_source_catch_catch.hpp

Define Documentation

TEST_CASE (...)

Define TEST_CASE_METHOD

- Defined in file_source_catch_catch.hpp

Define Documentation

TEST_CASE_METHOD(*className*, ...)

Define THEN

- Defined in file_source_catch_catch.hpp

Define Documentation

THEN(*desc*)

Define UNSCOPED_INFO

- Defined in file_source_catch_catch.hpp

Define Documentation

UNSCOPED_INFO(*msg*)

Define WARN

- Defined in file_source_catch_catch.hpp

Define Documentation

WARN(*msg*)

Define WHEN

- Defined in file_source_catch_catch.hpp

Define Documentation

WHEN(*desc*)

Define WORLD_SETUP_C

- Defined in file_source_default_mode_WorldSetup.cc

Define Documentation

Warning: doxygendefine: Cannot find define "WORLD_SETUP_C" in doxygen xml output for project "Symbulation" from directory: ./doxyoutput/xml

Typedefs

Typedef Catch::exceptionTranslateFunction

- Defined in file_source_catch_catch.hpp

Typedef Documentation

using Catch::**exceptionTranslateFunction** = std::string (*) ()

Typedef Catch::ExceptionTranslators

- Defined in file_source_catch_catch.hpp

Typedef Documentation

using Catch::**ExceptionTranslators** = std::vector<std::unique_ptr<*IExceptionTranslator* **const**>>

Typedef Catch::FunctionReturnType

- Defined in file_source_catch_catch.hpp

Typedef Documentation

using Catch::**FunctionReturnType** = **typename** std::remove_reference<**typename** std::remove_cv<**typename** std::resu

Typedef Catch::Generators::GeneratorBasePtr

- Defined in file_source_catch_catch.hpp

Typedef Documentation

using Catch::Generators::**GeneratorBasePtr** = std::unique_ptr<*GeneratorUntypedBase*>

Typedef Catch::IConfigPtr

- Defined in file_source_catch_catch.hpp

Typedef Documentation

typedef std::shared_ptr<*IConfig* **const**> Catch::**IConfigPtr**

Typedef Catch::IReporterFactoryPtr

- Defined in file_source_catch_catch.hpp

Typedef Documentation

using Catch::**IReporterFactoryPtr** = std::shared_ptr<IReporterFactory>

Typedef Catch::StringMatcher

- Defined in file_source_catch_catch.hpp

Typedef Documentation

using Catch::**StringMatcher** = Matchers::Impl::*MatcherBase*<std::string>

1.5 Getting started with Symbulation development

This document is intended to help those just getting started with Symbulation development. It details the initial one-time dependency installs and any similar routines necessary to get started with development.

Start by making your own copy of Symbulation and setting yourself up for development; then, build Symbulation and run the tests; and finally, claim an issue and start developing!

If you're unfamiliar with git and branching in particular, check out the git-scm book.

1.5.1 One-time Preparation

1. Get a GitHub account.

   (We use GitHub to manage Symbulation contributions.)

2. Fork github.com/anyaevostiar/SymbulationEmp.

   Visit that page, and then click on the 'fork' button (upper right).

   This makes a copy of the Symbulation source code in your own GitHub account. If you have contributor permissions to the main Symbulation library, this step is optional (you can instead develop on a branch within the main repo).

3. Clone your copy of Symbulation and Empirical to your local development environment.

   Your clone URLs should look something like this:

   ```
   https://github.com/yourusername/SymbulationEmp.git
   https://github.com/devosoft/Empirical.git
   ```

   and the UNIX shell command should be:

   ```
   git clone https://github.com/yourusername/SymbulationEmp.git
   git clone https://github.com/devosoft/Empirical.git
   ```

   (This makes a local copy of Symbulation on your development machine.)

4. Add a git reference to the Symbulation repository:

   ```
   cd SymbulationEmp
   git remote add upstream https://github.com/anyaevostinar/SymbulationEmp.git
   cd ../
   ```

   (This makes it easy for you to pull down the latest changes in the main repository.)

Building Symbulation and running the tests

To run the tests:

```
    make test
```

Congratulations! You're ready to develop!

1.5.2 Claiming an issue and starting to develop

1. Find an open issue and claim it.

   Once you've found an issue you like, make sure that no one has been assigned to it (see "assignee", bottom right near "notifications"). Then, add a comment "I am working on this issue." You've staked your claim!

   (We're trying to avoid having multiple people working on the same issue.)

2. In your local copy of the source code, update your main branch from the main Symbulation main branch:

   ```
   git checkout main
   git pull upstream main
   ```

(This pulls in all of the latest changes from whatever we've been doing on Symbulation.)

It is possible that when you do a [git pull]{.title-ref} you will get a "merge conflict" – This is what happens when something changed in the branch you're pulling in in the same place you made a change in your local copy.

Git will complain loudly about merges and tell you specifically in which files they occurred. If you open the file, you'll see something vaguely like this in the place where the merge occurred:

```
<<<<<<< HEAD
Changes made on the branch that is being merged into. In most cases,
this is the branch that you have currently checked out
=======
Changes made on the branch that is being merged in, almost certainly
main.
>>>>>>> abcde1234
```

Though there are a variety of tools to assist with resolving merge conflicts they can be quite complicated at first glance and it is usually easy enough to manually resolve the conflict.

To resolve the conflict you simply have to manually 'meld' the changes together and remove the merge markers. Make sure that you don't remove someone else's changes while resolving the merge conflict!

After this you'll have to add and commit the merge just like any other set of changes. You should also run the tests to make sure that everything works as expected.

3. Create a new branch and link it to your fork on GitHub:

```
git checkout -b fix/brief_issue_description
git push -u origin fix/brief_issue_description
```

where you replace "brief_issue_description" with 2-3 words, separated by underscores, describing the issue.

(This is the set of changes you're going to ask to be merged into Symbulation.)

4. Make some changes and commit them.

Though this will largely be issue-dependent the basics of committing are simple. After you've made a cohesive set of changes, run the command [git status]{.title-ref}. This will display a list of all the files git has noticed you changed. A file in the 'untracked' section are files that haven't existed previously in the repository but git has noticed.

To commit changes you have to 'stage' them–this is done by issuing the following command:

```
git add path/to/file
```

If you have a large quantity of changes and you don't want to add each file manually you can do `git add --patch` which will display each set of changes to you before staging them for commit.

Once you have staged your changes, it's time to make a commit:

```
git commit -m "added x change"
```

Git will then open your default console text editor to write a commit message – this is a short (typically 1-3 sentence) description of the changes you've made. Please make your commit message informative but concise – these messages become part of the 'official' history of the project.

Once your changes have been committed, push them up to the remote branch:

```
git push
```

If this is your first commit on a new branch git will error out, telling you the remote branch doesn't exist – This is fine, as it will also provide the command to create the branch. Copy/paste/run and you should be set.

5. Periodically update your branch from the Symbulation main branch:

```
git pull upstream main
```

(This pulls in all of the latest changes from whatever we've been doing on the upstream branch- important especially during periods of fast change or for long-running pull requests.)

6. Run the tests *before* pushing to GitHub:

```
make test
```

Make sure they all pass!

7. Push your branch to your own GitHub fork:

```
git push origin
```

(This pushes all of your changes to your own fork.)

8. Repeat until you're ready to merge your changes into "official" Symbulation.

9. Set up a Pull Request asking to merge things into the central Symbulation repository.

In a Web browser, go to your GitHub fork of Symbulation, e.g.:

```
https://github.com/yourusername/SymbulationEmp
```

and you will see a list of "recently pushed branches" just above the source code listing. On the right side of that should be a "Compare & pull request" green button. Click on it!

Now:

- add a descriptive title ("updated tests for XXX")
- put the issue number in the comment ("fixes issue #532")

then click "Create pull request."

(This creates a new issue where we can all discuss your proposed changes; the Symbulation team will be automatically notified and you will receive e-mail notifications as we add comments. See GitHub flow for more info.)

10. As you add new commits to address bugs or formatting issues, you can keep pushing your changes to the pull request by doing:

```
git push origin
```

11. If we request changes, return to the step "Make some changes and commit them" and go from there. Any additional commits you make and push to your branch will automatically be added to the pull request (which is pretty dang cool.)

1.5.3 After your first issue is successfully merged...

You're now an experienced GitHub user! Go ahead and take some more tasks; you can broaden out beyond the low hanging fruit if you like.

1.5.4 Your second contribution...

Here are a few pointers on getting started on your second (or third, or fourth, or nth contribution).

So, assuming you've found an issue you'd like to work on there are a couple things to do to make sure your local copy of the repository is ready for a new issue–specifically, we need to make sure it's in sync with the remote repository so you aren't working on an old copy. So:

```
git checkout main
git fetch --all
git pull
```

This puts you on the latest main branch and pulls down updates from GitHub with any changes that may have been made since your last contribution (usually including the merge of your last contribution). Then we merge those changes into your local copy of the main branch.

Now, you can go back to *Claiming an issue and starting to develop*.

1.5.5 Pull request cleanup (commit squashing)

Submitters are invited to reduce the numbers of commits in their pull requests either via [git rebase -i upstream/main]{.title-ref} or this recipe:

```
git pull ## make sure the local is up to date
git pull upstream main ## get up to date
## fix any merge conflicts
git status ## sanity check
git diff upstream/main ## does the diff look correct? (no merge markers)
git reset --soft upstream/main ## un-commit the differences from dib/main
git status ## sanity check
git commit --all ## package all differences in one commit
git status ## sanity check
git push ## should fail
git push --force ## override what's in GitHub's copy of the branch/pull request
```

1.6 Coding guidelines and review checklist

This document is for those who want to contribute code or documentation fixes to the Symbulation project and describes our coding standards as well as our code review process.

This document has been adapted from the khmer project

1.6.1 C++ standards

We use C++17 features throughout the project and currently that is the de-facto standard version to use.

All code should be in header files for ease of inclusion into Emscripten projects.

Files that define a single class should be named after that class. Files that define sets of functions or multiple classes should have an all-lowercase name that describes its contents.

All files and all directories must be levelized. This is partly enforced through all files being header files (and thus we cannot have circular dependencies), but for clean coding practices (and easy of unit testing) whole directories should not refer to each other bidirectionally either. See Large-Scale C++ Software Design by John Lakos for a strong pro-levelization argument.

In-code identifier formatting is always hard to settle upon. The guidelines below are for consistency.

- Variable names should be all_lowercase, with words separated by underscores
- Function names should be CamelCase() unless they are meant to mimic a function from the C++ standard library, at which point they can be all_lowercase to fit in.
- User-defined types should be CamelCase
- Constants should be ALL_UPPERCASE, with words separated by underscores
- Template parameters should be ALL_UPPERCASE.
- Typedefs should match the casing of the types they are aliasing. For example, a typedef on a template parameter might be all uppercase, while a typedef on a user-defined type should be CamelCase.

1.6.2 Guidelines based on Emscripten Limitations

- Try to avoid use of 64-bit integers (that is, the "long long" type). Emscripten has to emulate these and they can cause a notable slowdown.
- Do not rely on exceptions when possible. Emscripten is slow at dealing with them and they can slow down code even when not triggered.
- Do not write multithreaded code that uses shared state. Javascript cannot (yet) handle such code and as such Emscripten cannot compile it. Note that Emscripten does have experimental support of pthreads.
- Obviously, do not use any architecture-specific tricks, such as assuming endianness, doing unaligned reads or writes, directly accessing registers, etc.

Please see the Emscripten doc page for a full list.

1.6.3 General Standards

All plain-text files should have line widths of 100 characters or less unless that is unsupported for the particular file format.

All contributions should have their spelling checked before being committed to the codebase.

Vim users can run:

```
:setlocal spell spelllang=en_us
```

to automagically check the spelling within the file being edited.

It's expected that before requesting a code review the author of the PR will have checked the code. It's also expected that whomever reviews the PR will check the code individually as well. Though the CI runs most of these and will

pass/fail the PR accordingly it is not infallible and the whole point of having a code review process is to have human eyes go over the changes to the codebase.

1.7 Documentation for Symbulation Documentation

This is a quick primer on how to document things within Symbulation.

Symbulation makes use of the Sphinx documentation system based off of XML information gathered from Doxygen via a plugin named Breathe. This means that Doxygen will automatically build documentation for anything written in a C++ source file and Sphinx will be used to organize how that documentation is displayed.

1.7.1 Dependencies

You'll need to install some additional things to build the documentation locally. If you make additions to the documentation, please build it locally to make sure that it is formatted well before making a pull request.

You need to install

- Doxygen
- Sphinx
- pip/pip3

For Homebrew users, that looks like this:

```
brew install sphinx
brew install doxygen
brew install pip3
```

You can then use `pip3` to install the rest of the requirements:

```
cd SymbulationEmp/docs
pip3 install -r requirements.txt
```

1.7.2 Building Documentation Locally

You are then ready to make your local documentation and run it:

```
make html
cd _build/html/
python3 -m http.server
```

1.7.3 How to Comment for Doxygen Autodoc

Doxygen has an entire documentation section on how to comment your code. We'll provide a trimmed version here so that you can get started quickly.

Doxygen will examine all comments to determine if they are documentation comments or just code comments. To make a documentation comment you must add either an extra * or /, depending on the kind of comment:

```
/**
 * This is a documentation comment
 * across several lines
 *
 * This comment will be associated with the function immediately following.
*/
void somefunc(sometype param)
{

}

// this is a comment that doxygen will ignore
// note how it only has two leading slashes, like a normal comment
/// This is a comment that will be included in the documentation
/// Note the extra leading slash
/// Huzzah, documentation
```

Note: Doxygen requires a minimum of *three* triple slash'd lines before a block is considered documentation:

```
/// This line will be ignored
int somefunc() { return 5; }

///
/// This line will be included
///
void otherfunc() { ; }
```

If you wish to make a more visible comment block, e.g. a header for a class, then you may do something like the following:

```
/*********************************************//**
* Here is some text inside a visible block
***********************************************/
```

Note: Doxygen will view this as any other documentation comment and will not render it any differently than a 'normal' documentation comment–it is simply more visible within the source code.

1.7.4 How to include Doxygen's autodocs within Sphinx files

Through the use of the Breathe extension it is incredibly easy to include Doxygen autodocs within a Sphinx documentation file.

Suppose we have a C++ implementation file name `lily.h` that has inline comment documentation as detailed above and that `lily.h` is a component of a module named `flowers` that was just created.

To document them, you must create a file within the Symbulation Library documentation source to hold the module's documentation:

```
    touch doc/library/flowers.md
```

Within `flowers.md` you can make an introduction to the module, etc., and then add in the sphinx directives to include auto-documentation. Your `flowers.md` file should look something like the following:

```
    # This is the flowers documentation!

    This is a very short introduction.

    ## lily.h

    ```{eval-rst}
    .. doxygenfile:: lily.h
       :project: Symbulation
    ```
```

When the docs are built Sphinx will automatically pull the available documentation from Doxygen's XML files to construct the docs.

Additional directives exist to include auto-documentaiton from different levels, the full breakdown of which is available within the Breathe Documentation.

1.7.5 How to add docs to the Sphinx documentation

Sphinx is the system used to generate the developer guide and similar reference documentation. A primer to using ReStructured Text, the markup language used by Sphinx, can be found here. You can also look at any of the [.rst]{.title-ref} files in the [doc/]{.title-ref} directory to get a feel for how thinks work.

New documents must be included in the `toctree` in the `index.md` file for the directory the added file lives in. For example, if you add `CowFacts.md` to the `CoolFacts/` directory you must add `CowFacts.md` to the toctree found in `CoolFacts/CowFacts.md`:

```
    # Cool Facts

    A bunch of cool facts!

    ```{toctree}
    AnteaterFacts
    BirdFacts
    CowFacts
    ```
```

1.8 Guide to Testing in Symbulation

This document details how testing works in Symbulation, both for writing and understanding tests.

1.8.1 Running Tests

In the root directory of Symbulation, use the maketarget `test`, like so:

```
make test
```

The tests will compile and execute automatically, and you should see output that looks something like this:

```
g++ -O3 -DNDEBUG -Wall -Wno-unused-function -std=c++17 -I../Empirical/include/ source/
↪catch/main.cc -o symbulation.test
# Execute tests
```

(continues on next page)

(continued from previous page)

```
./symbulation.test
===============================================================================
All tests passed (592 assertions in 70 test cases)
```

1.8.2 Writing Tests

It is required that contributions to the Symbulation library have test coverage. Though writing tests can be a complex task in some cases, it can also be easy to do.

In general the best way to understand how to write tests is to look at the existing tests. Each header file in source/ has a file full of tests ending with "class_name.test.cc". We recommend browsing through those files.

The test cases should have the following layout:

```
TEST_CASE("Test name goes here", "[test classification here]")
{
        // body of test
}
```

Within a test case you can use the `REQUIRE` macro like an assert, to require certain conditions within the test:

```
REQUIRE(1==1); // will pass, obviously
REQUIRE(1==0); // will fail, and Catch will complain
```

If a `REQUIRE` fails, the compiler will give an error when you run "make test".

genindex

INDEX

D

E

F

G

H

I

L

M

O

P

R

S

T

U

W