

---

# Symbulation

*Release 0.0.1*

**Anya Vostinar**

**Dec 05, 2022**



# USING SYMBULATION

<b>1</b>	<b>Usage</b>	<b>3</b>
<b>2</b>	<b>Table of Contents</b>	<b>5</b>
2.1	Projects using Symbulation . . . . .	5
2.2	Quick Start Guides . . . . .	5
2.3	ALIFE 2022 Symbulation Tutorial . . . . .	21
2.4	ALIFE Talks Featuring Symbulation . . . . .	23
2.5	Library API . . . . .	24
2.6	Getting started with Symbulation development . . . . .	277
2.7	Coding guidelines and review checklist . . . . .	281
2.8	Documentation for Symbulation Documentation . . . . .	282
2.9	Guide to Testing in Symbulation . . . . .	285
<b>Index</b>		<b>287</b>



**Authors** Anya Vostinar and contributors.

**GitHub** <https://github.com/anyaevostinar/SymbulationEmp>

# Symbulation

Agent-based modeling of symbiont ecology and evolution

Symbulation is an artificial life software tool for agent-based modeling of the evolution of biological symbiosis, which can occur along the spectrum between parasitism and mutualism.

Try out our [browser-based GUI](#). See our documentation at <https://symbulation.readthedocs.io>.

Interested in starting a new project with Symbulation? We have a cookiecutter template here: <https://github.com/anyaevostinar/SymbulationProjectTemplate>.

Symbulation is built with the [Empirical platform](#) and the cookiecutter includes our recommended directory layout and example analysis files.



---

## CHAPTER

# ONE

---

## USAGE

Install the latest Cookiecutter:

```
pip install -U cookiecutter
```

Generate an Symbulation project:

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

Move into the new project directory and the SymbulationEmp directory and make:

```
cd SymbulationProject/SymbulationEmp  
make
```

Make any changes that you wish to the file SymSettings.cfg and then run:

```
./symbulation_default
```

By default, your data will be output to the files HostVals\_data\_\_SEED10.data and SymVals\_data\_\_SEED10.data.

We recommend that you copy your symbulation executable to your Data folders:

```
cp symbulation_default ../Data/sample_treatment  
cd ../Data/sample_treatment
```

You can then use the provided Python script to run several replicates:

```
python3 simple_repeat.py
```

You can also then use the provided Python script to transform your data into a format more easily used by R:

```
cd ../../Analysis/sample_treatment  
python3 munge_data.py
```

And then open our provided R script SampleAnalysis.R in R, set your working directory to the Analysis directory and then run all lines to observe the effect of vertical transmission rate on the interaction value evolved.

Created part of my free logo at LogoMakr.com.

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!



## TABLE OF CONTENTS

### 2.1 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 your 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.

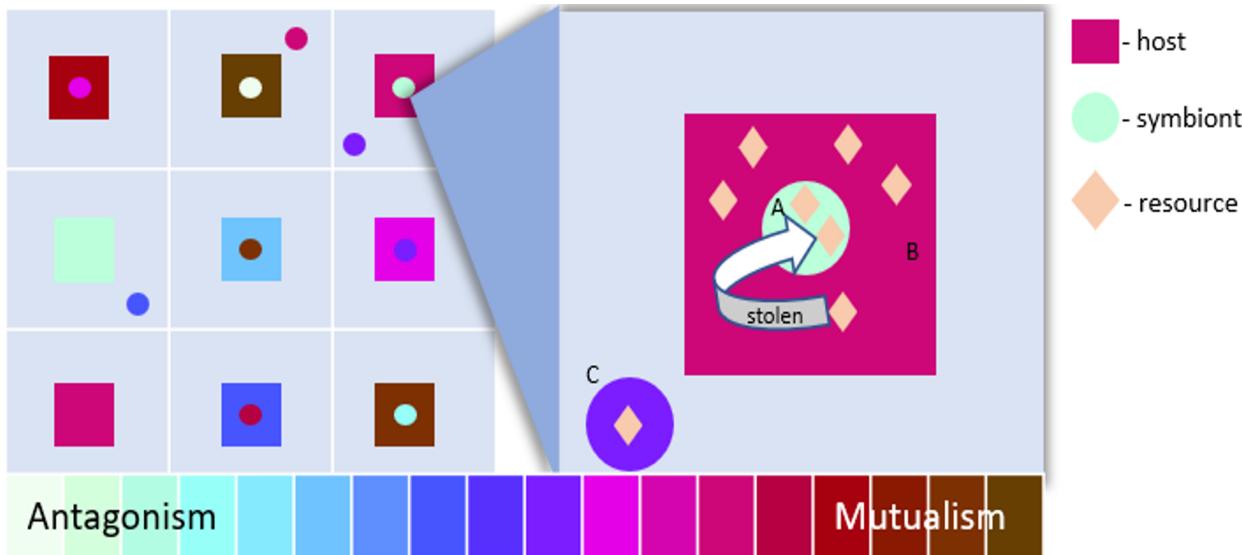
- An investigation into the conditions for the evolution of endosymbiosis
  - Symbulation: v0.2
  - Git: <https://github.com/anyaevostinar/Evolution-of-Endosymbiosis-Paper>
- An investigation into the evolution of lysis in bacteriophage
  - Symbulation: v0.2
  - Git: <https://github.com/anyaevostinar/Evolution-of-Lysogeny-Paper>
- 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>

### 2.2 Quick Start Guides

Contents:

## 2.2.1 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.

## 2.2.2 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.

If you aren't able to install everything locally, you can try out editing the command-line version of the code on [this Repl](#). The web version can't be rebuilt on Replit at the moment due to missing packages on the Replit platform.

### 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. Cookiecutter will attempt to push your new directory to a git repo; establish a git repo with the appropriate project name if you would like to use this functionality.

- Install the latest Cookiecutter. Depending on your Python, you may need to use `pip3` in place of `pip`:

```
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
```

## Compiling

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

```
cd SymbulationEmp  
make
```

Then run:

```
./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 -GRID_X 50 -GRID_Y 50
```

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 to set up Emscripten, which is included in the SymbulationEmp folder:

```
cd emsdk  
./emsdk install 1.38.48  
./emsdk activate 1.38.48
```

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's top folder and build the web version:

```
cd ..  
make web
```

Now you have the website all built and in the web folder, so just run:

```
make serve
```

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

### 2.2.3 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_default ../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 (inclusive).

For example, the input

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

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

## 2.2.4 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. We recommend using RStudio for running R scripts. You can find the documentation and information on how to [download RStudio here](#).

## 2.2.5 Creating a New Configuration Setting

Once you start making changes to Symbulation, you'll probably want to be able to control those changes from the command line at run time. It is very easy to do so because Symbulation uses Empirical's configuration management system. This guide walks you through the step-by-step process of adding and using a new configuration setting.

## Add Setting to ConfigSetup.h

The first step for adding a new configuration setting is to navigate to the configuration setup file `SymbulationEmp/source/ConfigSetup.h`. This file is currently shared by all modes, but we're planning to eventually have separate files for each mode because it's already getting rather crowded!

In this file, you'll see that there are several groups. You are probably safe putting your configuration setting into the `MAIN` group, unless it is a mutation rate or size. If it is a mutation rate or size and you want it to be zeroed out during the "no mutation updates" at the end of an experiment, you should put it into the `MUTATION` group. If you are making a lot of new configuration settings that are related to each other, you might want to have a group all to yourself.

The syntax for adding a new value is to just add a new line with the following:

```
VALUE(NAME_OF_SETTING, type, default_value, "English description of setting"),
```

For example, when I was creating a new limited pool of resources for the world, here is the configuration setting that I added:

```
VALUE(LIMITED_RES_TOTAL, int, -1, "Number of total resources available over the  
entire run, -1 for unlimited"),
```

It is common convention in Symbulation's configuration settings for `-1` to be used to turn a feature off (so that we don't have to have two settings for each feature).

Bug alert: Make sure you don't forget the comma at the end!

That's all you have to do to get your new configuration option in the configuration object that gets passed all around the code!

## Using and Changing the Setting

Now that you have your new configuration setting, you can access it pretty easily from most classes. `Host`, `Symbiont`, and `SymWorld` (and therefore their subclasses) all have an instance variable pointing to the configuration object that they all share, generally called `my_config`. The configuration class automatically provides both getters/accessors and setters/mutators for all the configuration settings!

To access your new configuration in `SymWorld`, `Host`, `Symbiont` or any of their subclasses:

```
type name_of_setting = my_config->NAME_OF_SETTING();
```

For example, in `SymWorld`, here is how the amount of total resources is accessed:

```
total_res = my_config->LIMITED_RES_TOTAL();
```

If you want to change the setting from what the configuration file specified, you can use the provided setter:

```
my_config->NAME_OF_SETTING(new_value);
```

This is particularly useful to do in test files, like so:

```
SymConfigBase config;  
config.LIMITED_RES_TOTAL(150);
```

Remember that you can now also change your configuration setting on the command line (also handled for you automatically):

```
./symbulation_default -NAME_OF_SETTING value
```

It will also automatically show up in the configuration panel on the web GUI if it is in the `MAIN` group, though the GUI might not be able to show the settings effect depending on what it does.

### Trying it out

The sky is the limit when it comes to new features that can be controlled with configuration settings. Here are some ideas of things to practice with:

- Expand on the limited resources by having a small amount trickle into the world every update or based on chance or something about the population
- Change something about how hosts and symbionts interact with each other, such as how resources are distributed (in `Host::DistribResources`)
- Make various catastrophic events happen in the world at random or set times, such as a portion of the population dying

## 2.2.6 Creating A New Mode

*Note: this guide is in progress and does not contain all helpful details.*

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. This guide explains how you can add a new *mode* to Symbulation, which is preferred when you have more new functionality that you want to add than can be handled by a couple of configuration settings. The guide assumes that you are familiar with the ideas of inheritance and classes. It gives a fair amount of assistance on how to do those in C++, though familiarity with general syntax of C++ is assumed.

There are a lot of different ways that symbiosis can occur that have overlaps but are functionality different. With Symbulation, we want to both support lots of those different ways of implementing symbiosis while keeping any one approach from getting too cluttered and bogged down with code that isn't relevant. We also don't want to repeat a bunch of code that is shared between those different ways (which we don't always succeed at, but we're constantly improving the codebase!). We achieve these dual goals with *modes*.

A mode generally has its own subclasses of `Symbiont` and/or `Host` and `SymWorld` defined so that they can be highly customized to the particular functionality of the mode. It also has its own compile target and test suite. The existing modes include:

- `default` (`Host` and `Symbiont` in the “classic” Symbulation functionality of interaction or resource behavior values),
- `efficient` (`EfficientHost` and `EfficientSymbiont` for work on the Dirty Transmission Hypothesis),
- `lysis` (`Bacterium` and `Phage` to study bacteriophage/bacterial-specific dynamics),
- and public goods game or pgg (`PGGHost` and `PGGSymbiont` to study ways for symbionts to interact within a host).
- We're also working a new mode `sgp` that uses the SignalGP-Lite library to have hosts and symbionts with computer programs as their genomes! You can check out our progress on the `complex-genomes` branch.

If you want to create new subclasses of `Host`, `Symbiont`, or `SymWorld`, a new mode is the best way to go. There are several steps to creating your own mode, including following conventions for file structure, adding your own organisms, adding a `WorldSetup` file, adding targets to the makefile, designing tests, and more. This guide will walk you through how to properly add most of these features.

### Makefile

First, you'll want to add the necessary targets to the `Makefile` for your new mode, so that you can compile and test your code as you go. This file can be a little overwhelming since there is a lot there already, but the bare minimum that you'll need is a compiling target. Navigate to the section of the file that looks like this:

```
default-mode: source/native/symbulation_default.cc
    $(CXX_nat) $(CFLAGS_nat) source/native/symbulation_default.cc -o symbulation_
→ default

efficient-mode: source/native/symbulation_efficient.cc
    $(CXX_nat) $(CFLAGS_nat) source/native/symbulation_efficient.cc -o_
→ symbulation_efficient

lysis-mode: source/native/symbulation_lysis.cc
    $(CXX_nat) $(CFLAGS_nat) source/native/symbulation_lysis.cc -o symbulation_
→ lysis

pgg-mode: source/native/symbulation_pgg.cc
    $(CXX_nat) $(CFLAGS_nat) source/native/symbulation_pgg.cc -o symbulation_pgg
```

Choose a one word descriptor for your new mode and add a line following the above template with your mode name.

If you wish, you can also add the more advanced targets that you will find helpful if you need to use Empirical's debug mode (great for finding memory leaks!) and running the test suite. These each have their own section with the other modes following the format that you can again copy and adapt:

- a debug target, with the naming convention “`debug-`” and “`-debug`”
- a testing target, with the naming convention “`test-`”
- a debug while testing target, with the naming convention “`test-debug-`”

### Primary Folder

Then make a folder in `SymbulationEmp/source` named `<name>_mode`. Inside of this folder will be a world setup source file (explained in more detail later), as well as any necessary header files such as your new subclasses.

### Organisms

Next, you will need to add new header files for your new organisms. The host organism(s) must extend the `Host` class in the `source/default_mode`, and similarly, the symbiont organism(s) must extend the `Symbiont` class. We'll go through how to make a new `Symbiont` subclass, but the process would be the same for the `Host` subclasses. Specifically, we'll use the `EfficientSymbiont` as an example since it's a fairly simple class.

Each new class must contain the following (which we'll go into depth below):

- `ifndef` macro
- constructor(s)
- `MakeNew()`

If you are adding a new trait or component of the genome, you will also need to:

- Define the relevant genome/instance variables
- Define a custom `Mutate()`
- Overwrite whichever super class methods your new trait is impacting

You'll probably also want to add some *configuration settings* to alter relevant things about your mode at runtime.

## Macros

Because there is a whole lot of `include-ing` going on, you should first put the following into your file to make sure it doesn't get added more than once:

```
#ifndef NAMEOFCCLASS_H
#define NAMEOFCCLASS_H

//All your code will go here

#endif
```

For example, here is the one from `EfficientSymbiont.h`:

```
#ifndef EFFSYM_H
#define EFFSYM_H

...

#endif
```

## Includes

Of course, you'll be referring to some other files, so you'll need to include them. The most obvious will be either `Host.h` or `Symbiont.h` depending on which you are inheriting from. Since `EfficientSymbiont` also knows about the new world type and host type, here are the includes from it:

```
#include "../default_mode/Symbiont.h"
#include "EfficientWorld.h"
#include "EfficientHost.h"
```

## Class definition

Now it's time to define your new class! To declare a class that inherits from another class you use the following syntax:

```
class YOUR_CLASS: public SUPER_CLASS {
    //All your class code here
}
```

For example:

```
class EfficientSymbiont: public Symbiont {
    //All the class code here
}
```

You can then add whatever new instance variables you want your class to have. Remember that your class will inherit the instance variables from the superclass, so you don't need to declare any of those.

For example, here are some of `EfficientSymbiont`'s new instance variables, with documentation removed:

```
protected:  
    double efficiency;  
    double ht_mut_size = 0.002;  
    double ht_mut_rate = 0;  
    double eff_mut_rate = 0;
```

### Constructor(s)

You'll probably want to define a constructor since it's unlikely the default is what you want. Your constructor needs to take parameters that the superclass will need as well as any additional parameters that you want. Here is one way of structuring your constructor for a Symbiont subclass, which calls the Symbiont constructor for you:

```
YourClassName(emp::Ptr<emp::Random> _random, emp::Ptr<SymWorld> _world, emp::Ptr  
    ↵<SymConfigBase> _config, double _intval=0.0, double _points = 0.0) : Symbiont(_  
    ↵random, _world, _config, _intval, _points) {  
    //Your specific code here  
}
```

Alternatively for a Host subclass, which needs a few more things:

```
YourClassName(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) :  
    Host(_random, _world, _config, _intval, _syms, _repro_syms, _set, _points) {  
    //Your specific code here  
}
```

You may also want to specify copy and move constructors, though they generally aren't used in Symbulation and can usually just be the default.

### MakeNew()

Because Symbulation uses a whole lot of inheritance and makes a whole lot of new objects of different subclasses through reproduction events, we rely on every class defining a `MakeNew()` method which can be used to make a new organism of the subclass without needing to know the type of the object. This method **should not** handle mutations, just make a new organism that is a clone of the current organism, wrapped in an Empirical pointer, and return it. It seems very simple, and it is, but it has allowed us to reduced a lot of repeated code between modes!

Here is the basic structure:

```
emp::Ptr<Organism> MakeNew() {  
    emp::Ptr<YourClassName> baby = emp::NewPtr<YourClassName>(random, my_world, my_  
    ↵config, GetIntVal());  
    //Setting your specific traits if needed  
    return baby;  
}
```

Here is an example from EfficientSymbiont:

```

emp::Ptr<Organism> MakeNew() {
    emp::Ptr<EfficientSymbiont> sym_baby = emp::NewPtr<EfficientSymbiont>(random, my_
    ↪world, my_config, GetIntVal());
    sym_baby->SetInfectionChance(GetInfectionChance());
    sym_baby->SetEfficiency(GetEfficiency());
    return sym_baby;
}

```

### (Optional) Creating new traits

If you are interested in adding a new heritable trait to your organism class, you will need to follow the next couple of optional sections. If you aren't adding a new heritable trait, you can skip them!

If you want your organism to have a new trait in its genome, you will need to add an instance variable for it up in the protected section and should probably add a Get and Set method as well.

For example, EfficientSymbionts have a new heritable trait called efficiency that is just how efficient they are at actually getting points. There is an instance variable:

```
double efficiency;
```

That instance variable is set in the constructor:

```
efficiency = _efficient;
```

And they have a getter and setter for it:

```

void SetEfficiency(double _in) {
    if(_in > 1 || _in < 0) throw "Invalid efficiency chance. Must be between 0 and 1,_
    ↪(inclusive)";
    efficiency = _in;
}

double GetEfficiency() {return efficiency;}

```

Then, it is used in the overwritten version of the AddPoints method:

```
void AddPoints(double _in) {points += (_in * efficiency);}
```

If you add a heritable trait, don't forget to make a new Mutate method for it (see below) and include it in the MakeNew method!

### (Optional) Mutate

If you are making a new heritable trait, you need to also specify how it should be mutated during reproduction.

You can either completely toss out the Mutate method of the superclass, as EfficientSymbiont does because it is testing some different mutation approaches, or you can just add on to the existing functionality. Since EfficientSymbiont is a bit complicated in this regard, we're going to switch examples to Phage. The Phage class has a couple of new heritable traits, but it also wants to keep mutating the original Symbiont traits without needing to repeat that code.

Here is the general structure for doing that by first calling the superclass method, and then doing your own logic:

```
void Mutate() {
    Symbiont::Mutate(); //or Host::Mutate(); for those subclasses
    double local_rate = my_config->MUTATION_RATE();
    double local_size = my_config->MUTATION_SIZE();
    //Repeat the below for each new trait that needs mutating
    if (random->GetDouble(0.0, 1.0) <= local_rate) {
        //mutate trait, assuming it should be between 0 and 1
        trait += random->GetRandNormal(0.0, local_size);
        if(trait < 0) trait = 0;
        else if (trait > 1) trait = 1;
    }
}
```

The Phage class has three new traits and has configuration settings for turning mutation for each of those traits on or off, as you can see in the example:

```
void Mutate() {
    Symbiont::Mutate();
    double local_rate = my_config->MUTATION_RATE();
    double local_size = my_config->MUTATION_SIZE();
    if (random->GetDouble(0.0, 1.0) <= local_rate) {
        //mutate chance of lysis/lysogeny, if enabled
        if(my_config->MUTATE_LYSIS_CHANCE()){
            chance_of_lysis += random->GetRandNormal(0.0, local_size);
            if(chance_of_lysis < 0) chance_of_lysis = 0;
            else if (chance_of_lysis > 1) chance_of_lysis = 1;
        }
        if(my_config->MUTATE_INDUCTION_CHANCE()){
            induction_chance += random->GetRandNormal(0.0, local_size);
            if(induction_chance < 0) induction_chance = 0;
            else if (induction_chance > 1) induction_chance = 1;
        }
        if(my_config->MUTATE_INC_VAL()){
            incorporation_val += random->GetRandNormal(0.0, local_size);
            if(incorporation_val < 0) incorporation_val = 0;
            else if (incorporation_val > 1) incorporation_val = 1;
        }
    }
}
```

### (Optional) World Class and Data Nodes

If you have added new evolvable traits to the organisms, you will probably want to find out information about those traits. You may also want to change how the environment impacts the organisms, even if you didn't make new inheritable traits. In either case, you'll need to create a new "world" class that inherits from `SymWorld` or one of its subclasses.

For example, here is the start of the `EfficientWorld` class:

```
#ifndef EFWORLD_H
#define EFWORLD_H

#include "../default_mode/SymWorld.h"
#include "../default_mode/DataNodes.h"

class EfficientWorld : public SymWorld {
```

(continues on next page)

(continued from previous page)

```
//All new code here
}
```

## DataMonitor/Node

Empirical provides a powerful data-tracking framework that works with the world classes, so there is only a bit of setup that you need to do to track and output data from your experiment. We're going to focus on data collection here, but of course if you want to change how the environment interacts with the organisms, you can do that by overwriting SymWorld methods in this class as well.

You'll need to first make a new instance variable for your `DataMonitor`. A data monitor is a special version of an Empirical `DataManager` that is focused on the data that it most recently received and the distribution of values that it has received previously, so it's useful for tracking what the data looks like every so many updates. There is a huge amount of power in the `DataManager` functionality that we are ignoring, so if there are different ways that you'd like to track your data, go take a look at Empirical's documentation!

We want our data instance variables to look like this:

```
emp::Ptr<emp::DataMonitor<type>> data_node_name;
```

For example, here is the new data monitor for tracking the efficiency trait of `EfficientSymbionts`:

```
/***
 *
 * Purpose: Data node tracking the average efficiency of efficient symbionts.
 *
 */
emp::Ptr<emp::DataMonitor<double>> data_node_efficiency;
```

Before we go further, we should make sure to clean up our memory, so let's define the world destructor to delete the data node:

```
~EfficientWorld() {
    if (data_node_efficiency) data_node_efficiency.Delete();
}
```

Next, we need to define what data is going to be stored in the data node. The data nodes use anonymous functions and event triggers so the world knows to add to them whenever specific events happen. The most common event that we want to add data during is `Update`, so we'll have a template that looks like this for defining what our data node does:

```
emp::DataMonitor<type>& GetNAMEDDataNode() {
    if (!data_node_name) { //if the data node doesn't already exist, make it!
        data_node_name.New();
        OnUpdate([this](size_t){ //this is the anonymous function that is called every
            update
            data_node_name->Reset(); //assuming you want to clear out data from the_
            //previous update
            for (size_t i = 0; i < pop.size(); i++) { //go through the population
                if (IsOccupied(i)) { //check that there is an organism
                    data_node_name->AddDatum(pop[i]->GetData()); //add whatever data to the_
                    //data node
                } //close if
            } //close for
        });
    }
}
```

(continues on next page)

(continued from previous page)

```

    }
    return *data_node_efficiency; //hand back the data node that's been created
}

```

As you can see from the inline comments, we are making a method that makes the data node if it doesn't already exist. When creating it, we add an unnamed function to the world's OnUpdate to-do list to go through the population and get the information that we want about each of our organisms, which we then add to the data node.

### DataFile

To get data out of the data node and into a file, we use Empirical's DataFile class. However, we need a method in the World class to actually setup the datafile and tell it what it will be doing. Here is the general structure of that method:

```

/*
 * 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 YOUR TRAIT
 */
emp::DataFile & SetupTRAITFile(const std::string & filename) {
    auto & file = SetupFile(filename); //A method from the Empirical World class
    auto & node = GetNAMEDataNode(); //The method you made previously
    file.AddVar(update, "update", "Update"); //You'll usually want the update_
    ↪information
    file.AddMean(node, "mean_TRAIT", "Average TRAIT", true);
    file.PrintHeaderKeys();

    return file;
}


```

Empirical's datafiles have many statistical methods already available including all the different flavors of averages, total, min/max, variance, skew, kurtosis, histogram bins, and ways for you to easily add new calculations. You can have multiple datanodes pulled from in the same file if you wish as well.

Finally, you should make a CreateDataFiles method that can be called in your .cc to make your new file in addition to the files from the superclass:

```

/*
 * Input: None.
 *
 * Output: None.
 *
 * Purpose: To create and set up the data files (excluding for phylogeny) that_
    ↪contain data for the YOUR_TRAIT condition experiment.
 */
void CreateDataFiles() {
    std::string file_ending = "_SEED"+std::to_string(my_config->SEED())+".data";
    SymWorld::CreateDateFiles();
    SetupTRAITFile(my_config->FILE_PATH()+"TRAIT"+my_config->FILE_NAME()+file_ending);
    ↪SetTimingRepeat(my_config->DATA_INT());
}


```

You should simply replace TRAIT with whatever your datafiles are called and add more setup calls if you have multiple datafiles.

### (Optional) World Setup

If you've made new organism(s) and a world, you'll need a new WorldSetup file. The world setup function is responsible for making organisms and placing them into the world. We are working on refactoring it to reduce duplicated code between modes, but for now, you will need to copy some code that is generally needed by every mode. The primary difference will be the organism types added to the world, which should now be your newly created host(s) and symbiont(s). Here is the structure with notes of what you should change:

```
#ifndef NAMEWORLD_SETUP_C //Change to your mode name throughout the includes
#define NAMEWORLD_SETUP_C

#include "YOURWorld.h"
#include "../ConfigSetup.h"
#include "YOURSymbiont.h"
#include "YOURHost.h"

//Change to your world type below
void worldSetup(emp::Ptr<YOURWorld> world, emp::Ptr<SymConfigBase> my_config) {
    emp::Random& random = world->GetRandom();

    double start_moi = my_config->START_MOI();
    long unsigned int POP_SIZE;
    if (my_config->POP_SIZE() == -1) {
        POP_SIZE = my_config->GRID_X() * my_config->GRID_Y();
    } else {
        POP_SIZE = my_config->POP_SIZE();
    }
    bool random_phen_host = false;
    bool random_phen_sym = false;
    if (my_config->HOST_INT() == -2) random_phen_host = true;
    if (my_config->SYM_INT() == -2) random_phen_sym = true;

    if (my_config->GRID() == 0) {world->SetPopStruct_Mixed(false);}
    else world->SetPopStruct_Grid(my_config->GRID_X(), my_config->GRID_Y(), false);

    //inject hosts
    for (size_t i = 0; i < POP_SIZE; i++) {
        emp::Ptr<YOURHost> new_org; //Change to your host type

        if (random_phen_host) {new_org.New(&random, world, my_config, random.GetDouble(-1,
            1));
        } else { new_org.New(&random, world, my_config, my_config->HOST_INT());
        }
        if (my_config->GRID()) {
            world->AddOrgAt(new_org, emp::WorldPosition(world->GetRandomCellID()));
        } else {
            world->AddOrgAt(new_org, world->size());
        }
    }

    //sets up the world size
    world->Resize(my_config->GRID_X(), my_config->GRID_Y());
}
```

(continues on next page)

(continued from previous page)

```
//This loop must be outside of the host generation loop since otherwise
//syms try to inject into mostly empty spots at first
int total_syms = POP_SIZE * start_moi;
for (int j = 0; j < total_syms; j++) {
    double sym_int = 0;
    if (random_phen_sym) {sym_int = random.GetDouble(-1,1);}
    else {sym_int = my_config->SYM_INT();}
}

//Change to your symbiont type below
emp::Ptr<YOURSymbiont> new_sym = emp::NewPtr<YOURSymbiont>(&random, world, my_
config, sym_int, 0, 1);
world->InjectSymbiont(new_sym);
}

#endif
```

If you have any new configuration settings that influence how organisms are first created, you will need to do that here.

## Native File

You'll next need to make a .cc file that sets everything up. This file can be fairly simple if you aren't making major changes from how Symbulation currently works. You should first add a file to source/native with the name symbulation\_<name>.cc where <name> is whatever you are calling your mode. This is the main source file that will allow the experiment to function. In it, the world is created, set up according to the setup file, and is permitted to run for the specified number of updates. Here is the template for this file:

```
#include "../YOUR_mode/YOURWorld.h"
#include "../YOUR_mode/YOURWorldSetup.cc"
#include "symbulation.h"

// This is the main function for the NATIVE version of this project.

int symbulation_main(int argc, char * argv[])
{
    SymConfigBase config;
    CheckConfigFile(config, argc, argv);

    config.Write(std::cout);
    emp::Random random(config.SEED());

    YOURWorld world(random, &config);

    YOURWorldSetup(&world, &config);
    world.CreateDateFiles();
    world.RunExperiment();

    return 0;
}

/*
This definition guard prevents main from being defined twice during testing.
In testing, Catch will define a main function which will initiate tests
(including testing the symbulation_main function above).
*/
```

(continues on next page)

(continued from previous page)

```
#ifndef CATCH_CONFIG_MAIN
int main(int argc, char * argv[]) {
    return symbulation_main(argc, argv);
}
#endif
```

Of course, if you want to make changes to how the experiment is setup or run, you can make this file more complicated or write your own version of provided functions.

## Experiment!

Once you have created your organisms and corresponding world, added a main source file, and added targets to the Makefile, you are ready to experiment! Run `make YOUR-mode` to build your code (and probably solve some bugs along the way).

- genindex
- search

## 2.3 ALIFE 2022 Symbulation Tutorial

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

### 2.3.1 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.

### 2.3.2 Advanced Prep

You don't need to do anything in advance of the workshop to gain from the time. However, if you want, there are some things that you could install ahead of time to speed things up:

- If you use a Windows system, we recommend that you install [Windows Subsystem for Linux](#) (Symbulation doesn't run on Windows unfortunately)
- Check if you have `gcc` and if you don't, install it. **This install can take a while**, hence why it's good to do ahead of the workshop. (`clang` might work, but we'll assume that you are using `gcc` and won't be able to help with issues with `clang`.)
- Install `cookiecutter` and pull down the Symbulation project template so that we can use it to set up the folder structure more easily. It will give an error at the end about not succeeding to push to GitHub; that's normal and nothing to worry about.
- Install [RStudio](#) so that you can make plots of the data that you generate with our provided R scripts (if you don't have an R program already)
- If you don't want to/can't install things, you can do most things using [this Replit](#), just not building the web version.

### 2.3.3 Schedule

#### Intro

(~20 Minutes)

- Get started downloading things (can be done in advance, see above for details)
  - If on Windows: [WSL](#)
  - `gcc`
  - `cookiecutter`
- High-level of what Symbulation is and how it works

#### Hands-on 1

(~15 minutes)

- Work through [making a cookie cutter project and running it](#)
- Work through [building the web version and running locally](#)
- [Change some settings and get some data](#); settings you could try changing:
  - VERTICAL\_TRANSMISSION
  - GRID
  - FREE\_LIVING\_SYMS
  - MUTATION\_SIZE

## Break

(~10 minutes)

## Explaining Code Structure

(~20 minutes)

- How all the classes work together
- Where the major functionality happens
- Where you might want to try out making changes

## Hands-on 2

(~15 minutes)

- Work through making a new configuration setting
- Work through making a new mode

## Wrap-up

(~10 minutes)

- What do you think?
- What would you want to make Symbulation something you would use in your own research or teaching?

## 2.4 ALIFE Talks Featuring Symbulation

Here are the talks at ALIFE 2022 that feature Symbulation! The links are in theory the Google Calendar invites, hopefully they work :).

- Monday 15:30-15:50 21. Claire Schregardus, Michael Wiser and Anya Vostinar Dirty Transmission Hypothesis: Increased Mutations During Horizontal Transmission Can Select for Increased Levels of Mutualism in Endosymbionts
- Wednesday 15:20-15:40 19. Kiara Johnson, Piper Welch, Emily Dolson and Anya Vostinar Endosymbiosis or Bust: Influence of Ectosymbiosis on Evolution of Obligate Endosymbiosis
- Wednesday 15:40-16:00 20. Alison Cameron, Seth Dorchen, Sarah Doore and Anya Vostinar Keep Your Frenemies Closer: Bacteriophage That Benefit Their Hosts Evolve to be More Temperate
- Thursday 15:00-15:20 4. Anya Vostinar, Katherine Skocelas, Alexander Lalejini and Luis Zaman Symbiosis in Digital Evolution: A Review and Future Directions
- Friday 15:00-15:20 126. Emily Dolson, Anya Vostinar, Shakeal Hodge and Zhen Ren Evolutionary stability of host-endosymbiont mutualism is reduced by multi-infection

## 2.5 Library API

### 2.5.1 Class Hierarchy

### 2.5.2 File Hierarchy

### 2.5.3 Full API

#### Namespaces

#### Namespace Catch

##### Contents

- *Namespaces*
- *Classes*
- *Enums*
- *Functions*
- *Typedefs*

#### Namespaces

- *Namespace Catch::detail*
- *Namespace Catch::detail*
- *Namespace Catch::Generators*
- *Namespace Catch::literals*
- *Namespace Catch::Matchers*

#### Classes

- *Template Struct always\_false*
- *Struct AssertionInfo*
- *Struct AssertionReaction*
- *Struct AutoReg*
- *Struct CaseSensitive*
- *Struct Counts*
- *Struct Decomposer*
- *Struct IConfig*
- *Struct IContext*
- *Struct IExceptionTranslator*

- *Struct IExceptionTranslatorRegistry*
- *Struct IGeneratorTracker*
- *Struct IMutableContext*
- *Struct IMutableEnumValuesRegistry*
- *Struct IMutableRegistryHub*
- *Struct IRegistryHub*
- *Struct IResultCapture*
- *Struct IRunner*
- *Template Struct is\_callable*
- *Template Struct is\_callable< Fun(Args... )>*
- *Struct is\_callable\_tester*
- *Template Struct is\_range*
- *Struct IStream*
- *Struct ITestCaseRegistry*
- *Struct ITestInvoker*
- *Struct ITransientExpression*
- *Struct MessageBuilder*
- *Struct MessageInfo*
- *Struct MessageStream*
- *Struct NameAndTags*
- *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 >*
- *Template Struct StringMaker< unsigned char[SZ]>*
- *Template Struct StringMaker< unsigned int >*
- *Template Struct StringMaker< unsigned long >*
- *Template Struct StringMaker< unsigned long long >*
- *Template Struct StringMaker< wchar\_t \* >*
- *Template Struct StringMaker< wchar\_t const \* >*
- *Struct TestCaseInfo*
- *Struct TestFailureException*
- *Struct Totals*
- *Template Struct true\_given*
- *Struct UseColour*
- *Struct WaitForKeypress*
- *Struct WarnAbout*
- *Class AssertionHandler*
- *Template Class BinaryExpr*
- *Class Capturer*
- *Class ExceptionTranslatorRegistrar*
- *Template Class ExceptionTranslatorRegistrar::ExceptionTranslator*
- *Template Class ExprLhs*

- *Class GeneratorException*
- *Class LazyExpression*
- *Template Class MatchExpr*
- *Class NonCopyable*
- *Template Class Option*
- *Class ReusableStringStream*
- *Class ScopedMessage*
- *Class Section*
- *Class SimplePcg32*
- *Class StringRef*
- *Class TestCase*
- *Template Class TestInvokerAsMethod*
- *Class Timer*
- *Template Class UnaryExpr*

## 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&, StringMatcher const&,StringRef const&)*
- *Function Catch::handleExceptionMatchExpr(AssertionHandler&, std::string const&,StringRef const&)*
- *Function Catch::handleExpression(ITransientExpression const&)*
- *Template Function Catch::handleExpression(ExprLhs<T> const&)*
- *Function Catch::isFalseTest*
- *Function Catch::isJustInfo*
- *Function Catch::isOk*
- *Function Catch::isThrowSafe*
- *Template Function Catch::makeMatchExpr*
- *Function Catch::makeStream*
- *Function Catch::makeTestCase*
- *Function Catch::makeTestInvoker(void(\*)())*
- *Template Function Catch::makeTestInvoker(void(C::\*))()*
- *Function Catch::matchTest*
- *Function Catch::operator""\_sr*
- *Template Function Catch::operator+*
- *Function Catch::operator+=*
- *Function Catch::operator<<(std::ostream&, SourceLineInfo const&)*
- *Function Catch::operator<<(std::ostream&,StringRef const&)*
- *Function Catch::operator|*
- *Template Function Catch::rangeToString(Range const&)*
- *Template Function Catch::rangeToString(std::vector<bool, Allocator> const&)*
- *Function Catch::replaceInPlace*
- *Function Catch::rng*

- *Function Catch::rngSeed*
- *Function Catch::shouldContinueOnFailure*
- *Function Catch::shouldSuppressFailure*
- *Function Catch::splitStringRef*
- *Function Catch::startsWith(std::string const&, std::string const&)*
- *Function Catch::startsWith(std::string const&, char)*
- *Function Catch::throw\_domain\_error*
- *Function Catch::throw\_exception*
- *Function Catch::throw\_logic\_error*
- *Function Catch::throw\_runtime\_error*
- *Function Catch::toLower*
- *Function Catch::toLowerInPlace*
- *Function Catch::translateActiveException*
- *Function Catch::trim(std::string const&)*
- *Function Catch::trim(StringRef)*

## Typedefs

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

## Namespace Catch::detail

### Contents

- *Classes*

## Classes

- *Template Struct is\_range\_impl*
- *Template Struct is\_range\_impl< T, typename void\_type< decltype(begin(std::declval< T >()))>::type >*
- *Template Struct void\_type*

## Namespace Catch::Detail

### Contents

- *Classes*
- *Functions*
- *Variables*

## 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*
- *Function Catch::Detail::rawMemoryToString(const void \*, std::size\_t)*
- *Template Function Catch::Detail::rawMemoryToString(const T&)*
- *Template Function Catch::Detail::stringify*

## Variables

- *Variable Catch::Detail::unprintableString*

## Namespace Catch::Generators

### Contents

- *Namespaces*
- *Classes*
- *Functions*
- *Typedefs*

### 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*

## Functions

- *Template Function Catch::Generators::pf::make\_unique*

## Namespace Catch::literals

### Contents

- *Functions*

## Functions

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

## Namespace Catch::Matchers

### Contents

- *NameSpaces*
- *Functions*

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

- *Function Catch::Matchers::Matches*
- *Function Catch::Matchers::Message*
- *Template Function Catch::Matchers::Predicate*
- *Function Catch::Matchers::StartsWith*
- *Template Function Catch::Matchers::UnorderedEquals*
- *Template Function Catch::Matchers::VectorContains*
- *Function Catch::Matchers::WithinAbs*
- *Function Catch::Matchers::WithinRel(double)*
- *Function Catch::Matchers::WithinRel(float, float)*
- *Function Catch::Matchers::WithinRel(float)*
- *Function Catch::Matchers::WithinRel(double, double)*
- *Function Catch::Matchers::WithinULP(double, uint64\_t)*
- *Function Catch::Matchers::WithinULP(float, uint64\_t)*

## Namespace Catch::Matchers::Exception

### Contents

- *Classes*

## Classes

- *Class ExceptionMessageMatcher*

## Namespace Catch::Matchers::Floating

### Contents

- *Classes*

## Classes

- *Struct WithinAbsMatcher*
- *Struct WithinRelMatcher*
- *Struct WithinUplsMatcher*

## Namespace Catch::Matchers::Generic

### Contents

- *Namespaces*
- *Classes*

### Namespaces

- *Namespace Catch::Matchers::Generic::Detail*

### Classes

- *Template Class PredicateMatcher*

## Namespace Catch::Matchers::Generic::Detail

### Contents

- *Functions*

### Functions

- *Function Catch::Matchers::Generic::Detail::finalizeDescription*

## Namespace Catch::Matchers::Impl

### Contents

- *Classes*

### Classes

- *Template Struct MatchAllOf*
- *Template Struct MatchAnyOf*
- *Template Struct MatcherBase*
- *Template Struct MatcherMethod*
- *Template Struct MatchNotOf*
- *Class MatcherUntypedBase*

## Namespace Catch::Matchers::StdString

### Contents

- *Classes*

### Classes

- *Struct CasedString*
- *Struct ContainsMatcher*
- *Struct EndsWithMatcher*
- *Struct EqualsMatcher*
- *Struct RegexMatcher*
- *Struct StartsWithMatcher*
- *Struct StringMatcherBase*

## Namespace Catch::Matchers::Vector

### Contents

- *Classes*

### Classes

- *Template Struct ApproxMatcher*
- *Template Struct ContainsElementMatcher*
- *Template Struct ContainsMatcher*
- *Template Struct EqualsMatcher*
- *Template Struct UnorderedEqualsMatcher*

## Namespace mpl\_

### Namespace std

#### Classes and Structs

##### Template Struct always\_false

- Defined in file\_sourceCatchCatch.hpp

## Inheritance Relationships

### Base Type

- public false\_type

### Struct Documentation

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

### Struct AssertionInfo

- Defined in file\_sourceCatchCatch.hpp

### Struct Documentation

```
struct Catch::AssertionInfo
```

#### Public Members

*StringRef* **macroName**  
*SourceLineInfo* **lineInfo**  
*StringRef* **capturedExpression**  
*ResultDisposition::Flags* **resultDisposition**

### Struct AssertionReaction

- Defined in file\_sourceCatchCatch.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 is\_range\_impl

- Defined in file\_source\_catch\_catch.hpp

## Inheritance Relationships

### Base Type

- public false\_type

## Struct Documentation

```
template<typename T, typename = void>
struct is_range_impl : public false_type
```

## Template Struct is\_range\_impl< T, typename void\_type< decltype(begin(std::declval< T >()))>::type >

- Defined in file\_source\_catch\_catch.hpp

## Inheritance Relationships

### Base Type

- public true\_type

## Struct Documentation

```
template<typename T>
struct is_range_impl<T, typename void_type<decltype(begin(std::declval<T>()))>::type> : public true_type
```

### Template Struct void\_type

- Defined in file\_sourceCatchCatch.hpp

## Struct Documentation

```
template<typename...>
struct Catch::detail::void_type
```

### Public Types

```
using type = void
```

### Template Struct as

- Defined in file\_sourceCatchCatch.hpp

## Struct Documentation

```
template<typename T>
struct as
```

### Template Struct IGenerator

- Defined in file\_sourceCatchCatch.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
double minDuration() 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 IEceptionTranslator

- Defined in file\_source\_catch\_catch.hpp

## Struct Documentation

**struct** Catch::IEceptionTranslator

### Public Functions

```
~IEceptionTranslator()  
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(StringRef generatorName, 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\_catchCatch.hpp

## Struct Documentation

```
template<typename T>  
struct is_callable
```

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

- Defined in file\_source\_catchCatch.hpp

## Inheritance Relationships

### Base Type

- public decltypeis\_callableTester::test< Fun, Args... >

## Struct Documentation

```
template<typename Fun, typename ...Args>  
struct is_callable<Fun(Args...)> : public decltypeis_callableTester::test<Fun, Args...>
```

### Struct `is_callableTester`

- Defined in file\_source\_catchCatch.hpp

## Struct Documentation

```
struct Catch::is_callableTester
```

## 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\_sourceCatchCatch.hpp

## Inheritance Relationships

### Base Type

- public Catch::detail::is\_range\_impl< T > (*Template Struct is\_rangeImpl*)

## Struct Documentation

```
template<typename T>  
struct is_range : public Catch::detail::is_range_impl<T>
```

## Struct `IStream`

- Defined in file\_sourceCatchCatch.hpp

## Struct Documentation

```
struct Catch::IStream
```

### Public Functions

```
~IStream()  
std::ostream &stream() const = 0
```

## Struct `ITestCaseRegistry`

- Defined in file\_sourceCatchCatch.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 `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\_sourceCatchCatch.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\_sourceCatchCatch.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, AllocComp, AllocMatch > (*Template Struct ApproxMatcher*)
- public Catch::Matchers::Vector::ContainsElementMatcher< T, Alloc > (*Template Struct ContainsElementMatcher*)
- public Catch::Matchers::Vector::ContainsMatcher< T, AllocComp, AllocMatch > (*Template Struct ContainsMatcher*)
- public Catch::Matchers::Vector::EqualsMatcher< T, AllocComp, AllocMatch > (*Template Struct EqualsMatcher*)
- public Catch::Matchers::Vector::UnorderedEqualsMatcher< T, AllocComp, AllocMatch > (*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, AllocComp, AllocMatch >, Catch::Matchers::Vector::ContainsElementMatcher<
T, Alloc >, Catch::Matchers::Vector::ContainsMatcher< T, AllocComp, AllocMatch >,
Catch::Matchers::Vector::EqualsMatcher< T, AllocComp, AllocMatch >,
Catch::Matchers::Vector::UnorderedEqualsMatcher< T, AllocComp, AllocMatch >
```

## 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::StringMatcherBase
```

### 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::string>
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, AllocMatch > >  
(*Template Struct MatcherBase*)

## Struct Documentation

```
template<typename T, typename AllocComp, typename AllocMatch>
struct Catch::Matchers::Vector::ApproxMatcher : public Catch::Matchers::Impl::MatcherBase<std::vector<T, AllocMatch>>
```

### Public Functions

```
ApproxMatcher(std::vector<T, AllocComp> const &comparator)
bool match(std::vector<T, AllocMatch> 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, AllocComp> 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, Alloc > > (*Template Struct MatcherBase*)

## Struct Documentation

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

## Public Functions

```
ContainsElementMatcher(T const &comparator)  
bool match(std::vector<T, Alloc> 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, AllocMatch > > (*Template Struct MatcherBase*)

## Struct Documentation

```
template<typename T, typename AllocComp, typename AllocMatch>
struct Catch::Matchers::Vector::ContainsMatcher : public Catch::Matchers::Impl::MatcherBase<std::vector<T,
```

### Public Functions

```
ContainsMatcher(std::vector<T, AllocComp> const &comparator)
bool match(std::vector<T, AllocMatch> const &v) const override
std::string describe() const override
```

### Public Members

```
std::vector<T, AllocComp> 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, AllocMatch > >  
(*Template Struct MatcherBase*)

## Struct Documentation

```
template<typename T, typename AllocComp, typename AllocMatch>
struct Catch::Matchers::Vector::EqualsMatcher : public Catch::Matchers::Impl::MatcherBase<std::vector<T, All
```

### Public Functions

```
EqualsMatcher(std::vector<T, AllocComp> const &comparator)
bool match(std::vector<T, AllocMatch> const &v) const override
std::string describe() const override
```

## Public Members

```
std::vector<T, AllocComp> const &m_comparator
```

## Template Struct UnorderedEqualsMatcher

- Defined in file\_sourceCatchCatch.hpp

## Inheritance Relationships

### Base Type

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

## Struct Documentation

```
template<typename T, typename AllocComp, typename AllocMatch>
struct Catch::Matchers::Vector::UnorderedEqualsMatcher : public Catch::Matchers::Impl::MatcherBase<std::vector< T, AllocMatch > >
```

## Public Functions

```
UnorderedEqualsMatcher(std::vector<T, AllocComp> const &target)
bool match(std::vector<T, AllocMatch> const &vec) const override
std::string describe() const override
```

## Struct MessageBuilder

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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 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 &line-  
Info)
```

## 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 Didn't ThrowException
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\_sourceCatchCatch.hpp

### Struct Documentation

```
struct Catch::StreamEndStop
```

### Public Functions

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

## Template Struct StringMaker

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.hpp

### Struct Documentation

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

#### Public Static Functions

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

## Template Struct StringMaker< char >

- Defined in file\_sourceCatchCatch.hpp

### Struct Documentation

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

#### Public Static Functions

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

## Template Struct StringMaker< char const \* >

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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>::value>::type>
```

### 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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatch.h

## 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\_sourceCatch.h

## 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\_sourceCatch.h

## 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\_sourceCatchCatch.hpp

## Struct Documentation

```
struct TestFailureException
```

## Struct Totals

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.hpp

## Inheritance Relationships

### Base Type

- public true\_type

## Struct Documentation

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

## Struct UseColour

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.hpp

## Struct Documentation

```
struct Catch_global_namespace_dummy
```

## Class Bacterium

- Defined in file\_sourceLysisModeBacterium.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 = {}, 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

```
std::string const GetName ()  
Input: None  
Output: Name of class as string, Bacterium  
Purpose: To know which subclass the object is  
  
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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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*) **const**

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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.hpp

## Nested Relationships

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

## Inheritance Relationships

### Base Type

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

## Class Documentation

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

### 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\_sourceCatchCatch.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 &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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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, Alloc > > (*Template Struct MatcherBase*)
- public Catch::Matchers::Impl::MatcherBase< std::vector< T, AllocMatch > > (*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, Alloc > >,
Catch::Matchers::Impl::MatcherBase< std::vector< T, AllocMatch > >
```

### 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 MatcherTclass 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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatch.h

## 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 = {}, 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

```
std::string const GetName ()
```

Input: None

Output: Name of class as string, *EfficientHost*

Purpose: To know which subclass the object is

```
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

```
std::string const GetName()
Input: None
Output: Name of class as string, EfficientSymbiont
Purpose: To know which subclass the object is

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 occurring. .

`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.

`void Setup()`

Definitions of setup functions, expanded in EfficientWorldSetup.cc

Input: None.

Output: None.

Purpose: Prepare the world for a simulation by applying the configuration settings and populating the world with efficient hosts and efficient symbionts.

```
void SetupHosts (long unsigned int *POP_SIZE)
Input: The number of efficient hosts.
Output: None.
Purpose: To populate the world with efficient hosts with appropriate phenotypes.

void SetupSymbionts (long unsigned int *total_syms)
Input: The number of efficient symbionts.
Output: None.
Purpose: To populate the world with efficient symbionts with appropriate phenotypes.

void CreateDataFiles ()
Input: None.
Output: None.
Purpose: To create and set up the data files (excluding for phylogeny) that contain data for the efficient condition experiment.

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 = {}, 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*)

`std::string const GetName()`

Input: None

Output: Name of class as string, *Host*

Purpose: To know which subclass the object is

`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.

`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 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.

int AddSymbiont (emp::Ptr<Organism> _in)
Input: The pointer to the organism that is to be added to the host's symbionts.
Output: The int describing the symbiont's position ID, or 0 if it did not successfully get added to the host's list of symbionts.
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 received 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 receive.

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 in-progress “reproductive” symbionts belonging to a host. These are symbionts that aren’t yet active. Symbionts can be added with AddReproSymb(). This can be cleared with [ClearSyms\(\)](#)

**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.

**void SetupHosts (long unsigned int \*POP\_SIZE)**

Definitions of setup functions, expanded in LysisWorldSetup.cc

Input: The number of bacteria.

Output: None.

Purpose: To populate the world with bacteria with appropriate phenotypes.

**void SetupSymbionts (long unsigned int \*total\_syms)**

Input: The number of phage.

Output: None.

Purpose: To populate the world with phage with appropriate phenotypes.

**void CreateDataFiles ()**

Input: None.

Output: None.

Purpose: To create and set up the data files (excluding for phylogeny) that contain data for the experiment.

**void SetupHostFileColumns (emp::DataFile &file)**

Input: The Empirical DataFile object tracking data nodes.

Output: None.

Purpose: To add bacterium data nodes to be tracked to the bacterium data file.

`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 symbiont.

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 bacterium and their phage

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

---

```
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
```

## 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
std::string const GetName()
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 ()
void SetResInProcess (double _in)
double GetResInProcess ()
double StealResources (double intval)
void SetSymbionts (emp::vector<emp::Ptr<Organism>> _in)
int 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 IncBurstTimerSetBurstTimer (double _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 (double _in)
void AddPool (double _in)
void DistribPool ()
double GetDonation ()
void SetDonation (double _in)
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* = {}, 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

std::string **const GetName** ()

Input: None

Output: Name of class as string, *PGGHost*

Purpose: To know which subclass the object is

**double GetPool** ()

Input: None

Output: The double representing the resource pool.

Purpose: To return the double representing the resource pool.

**void SetPool** (double *\_in*)

Input: A double to be set as the resource pool.

Output: None

Purpose: To set the number of resources in the resource pool.

**void AddPool** (double *\_in*)

Input: A double to be added to the resource pool.

Output: None

Purpose: To add resources into the host's resource pool.

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

Input: A double quantity of resources to be distributed.

Output: None

Purpose: To distribute resources to symbionts and collect resource donations from them.

**void DistribPool** ()

Input: None

Output: None

Purpose: To distribute the resource pool across the hosted symbionts, each donation multiplied by PGG synergy, and then set the resource pool to 0.

`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: Represents the collective resource pool held by the host.

`emp::Ptr<PGGWorld> my_world = NULL`

Purpose: Represents the world that the PGGHosts are living in.

## Class PGGSymbiont

- Defined in file\_source\_pgg\_mode\_PGGSymbiont.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: None

Output: None

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

`PGGSymbiont (PGGSymbiont&&) = default`

Input: None

Output: None

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

**PGGSymbiont ()** = default

Input: None

Output: None

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

**PGGSymbiont &operator= (const PGGsymbiont&)** = default

Input: None

Output: None

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

**PGGSymbiont &operator= (PGGSymbiont&&)** = default

Input: None

Output: None

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

**std::string const GetName ()**

Input: None

Output: Name of class as string, *PGGSymbiont*

Purpose: To know which subclass the object is

**double GetDonation ()**

Input: None

Output: The donation value for the symbiont.

Purpose: To get a symbiont's donation value.

**void SetDonation (double \_in)**

Input: The double to be set as the symbiont's donation value.

Output: None

Purpose: To set the symbiont's donation value.

**void Mutate ()**

Input: None

Output: None

Purpose: To mutate a *PGGSymbiont*'s donation value. 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.

**double ProcessPool ()**

Input: None

Output: The double representation of resources to be given to the host.

Purpose: Deteremines the resources that the symbiont is contributing to the host's resource pool, and decrements them from the symbiont's own own resource collection.

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

Input: None

Output: The pointer to the newly created organism

Purpose: To produce a new *PGGSymbiont*, identical to the original

---

```
std::string PrintSym(emp::Ptr<PGGSymbiont> org)
```

Input: The PGG symbiont to be printed

Output: A formatted string representation of the input symbiont, including its interaction value and donation value.

Purpose: To print the symbiont for testing purposes.

## Protected Attributes

```
double PGG_donate = 0
```

Purpose: the donation value for this symbiont.

```
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.

```
void SetupHosts (long unsigned int *POP_SIZE)
```

Definitions of setup functions, expanded in PGGWorldSetup.cc

Input: The number of PGG hosts.

Output: None.

Purpose: To populate the world with PGG hosts with appropriate phenotypes.

```
void SetupSymbionts (long unsigned int *total_syms)
```

Input: The number of PGG symbionts.

Output: None.

Purpose: To populate the world with PGG symbionts with appropriate phenotypes.

```
void CreateDataFiles ()
```

Input: None.

Output: None.

Purpose: To create and set up the data files (excluding for phylogeny) that contain data for the experiment.

```
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 mean information about the PGG symbionts in the world. This includes: (1) their total count, (2) the counts of the free and hosted symbionts, (3) the average donation values for all symbionts, and (4) the histogram data for the distribution of symbionts at various donation values.

```
emp::DataMonitor<double, emp::data::Histogram> &GetPGGDataNode ()
```

Input: None

Output: The DataMonitor<double, emp::data::Histogram>& that has information representing the average donation value for all symbionts in the system.

Purpose: To collect data on the average donation value to be saved to the data file that is tracking PGG data.

## 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 _inval = 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

std::string **const GetName ()**

Input: None

Output: Name of class as string, *Phage*

Purpose: To know which subclass the object is

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 chance 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: None  
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 host_donation, emp::Ptr<Organism> host = nullptr)
```

Input: The double representing the resources to be distributed to the phage and (optionally) the host from whom it comes; if no host is provided, the phage's host variable is used.

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 its 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 constructor 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.

```
std::string const GetName()
Input: None
Output: Name of class as string, Symbiont
Purpose: To know which subclass the object is

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 symbiont'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 symbiont'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 symbiont's host

Output: None

Purpose: To set a symbiont's host

```
void SetInfectionChance(double _in)
```

Input: The double that will be the symbiont's infection chance

Output: None

Purpose: To set a symbiont's infection host and check that the proposed value is valid.

```
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 *host\_donation*, emp::Ptr<*Organism*> *host* = nullptr)

Input: The double representing the resources to be distributed to the symbiont and (optionally) the host from whom it comes; if no host is provided, the symbiont's host variable is used.

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, emp::Ptr<SymConfigBase> \_config)  
Input: The world's random seed

Output: None

Purpose: To construct an instance of *SymWorld*

**~SymWorld()**

Input: None

Output: None

Purpose: To destruct the objects belonging to *SymWorld* to conserve memory.

**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 resources 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 id)**

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 InjectHost (emp::Ptr<Organism> new\_host)**

Input: The pointer to a host that will be added to the world. This function assumes that the pop vector has not been resized to fit the world yet.

Output: None

Purpose: To add a host to the world.

**void InjectSymbiont (emp::Ptr<Organism> new\_sym)**

Input: The pointer to an organism that will be injected into the world.

Output: None

Purpose: To add a symbiont to the world, either into a host or into a sym world cell.

**void CreateDataFiles ()**

Definitions of data node functions, expanded in DataNodes.h

Input: None.

Output: None.

Purpose: To create and set up the data files (excluding for phylogeny) that contain data for the experiment.

**void WritePhylogenyFile (const std::string &filename)**

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. Prints header keys to the file.

---

```
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::DataFile &SetUpTransmissionFile (const std::string &filename)
    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 counts of attempted transmissions.
```

```
void SetupHostFileColumns (emp::DataFile &file)
    Input: The Empirical DataFile object tracking data nodes.
    Output: None.

    Purpose: To define which data nodes should be tracked by this data file. Defines what columns should be called.
```

```
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> &GetHorizontalTransmissionAttemptCount()`

Input: None

Output: The DataMonitor<int>& that has the information representing how many attempts were made to horizontally transmit.

Purpose: To retrieve the data nodes that is tracking the number of attempted horizontal transmissions.

`emp::DataMonitor<int> &GetHorizontalTransmissionSuccessCount()`

Input: None

Output: The DataMonitor<int>& that has the information representing how many successful attempts were made to horizontally transmit.

Purpose: To retrieve the data nodes that is tracking the number of successful horizontal transmissions.

`emp::DataMonitor<int> &GetVerticalTransmissionAttemptCount()`

Input: None

Output: The DataMonitor<int>& that has the information representing how many attempts were made to vertically transmit.

Purpose: To retrieve the data nodes that is tracking the number of attempted vertical transmissions.

`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 Setup()`  
 Definitions of setup functions, expanded in WorldSetup.cc  
 Input: None.  
 Output: None.  
 Purpose: Prepare the world for an experiment by applying the configuration settings and populating the world with hosts and symbionts.

`void SetupHosts (long unsigned int *POP_SIZE)`  
 Input: The number of hosts.  
 Output: None.  
 Purpose: To populate the world with hosts with appropriate phenotypes.

`void SetupSymbionts (long unsigned int *total_syms)`  
 Input: The number of symbionts.  
 Output: None.  
 Purpose: To populate the world with symbionts with appropriate phenotypes.

`emp::WorldPosition 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: The WorldPosition object describing the symbiont's new location (it describes an invalid position if the symbiont is deleted during movement)  
 Purpose: To move a symbiont into a new world position.

`bool IsInboundsPos (emp::WorldPosition pos)`  
 Input: The WorldPosition object to be checked.  
 Output: Whether the input object is within world bounds.  
 Purpose: To determine whether the location of free-living organisms is within the bounds of the free-living worlds (the size of the pop and sym\_pop vectors).

`emp::WorldPosition 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: The WorldPosition object describing the position the symbiont was born into (index = position in a host, 0 for free living and offset by one for position in host sym vector. id = position of self or host in sym\_pop or pop vector). An invalid WorldPosition object is returned if the sym was killed.

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 extracted from the world.

Purpose: To extract a symbiont from the world without deleting it.

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 **SetMutationZero** ()

Input: None

Output: None

Purpose: To set all settings in the MUTATION group to 0 for the no-mutation updates.

void **RunExperiment** (bool *verbose* = true)

Input: Optional boolean “verbose” that specifies whether to print the update numbers to standard output or not, defaults to true.

Output: None

Purpose: Run the number of updates and non-mutation updates specified in the configuration settings.

void **Update** ()

Input: None

Output: None

Purpose: To simulate a timestep in the world, which includes calling the process functions for hosts and symbionts and updating the data nodes.

## Protected Types

using **fun\_calc\_info\_t** = std::function<int (*Organism*&)>

## Protected Attributes

`int total_res = -1`

Purpose: Represents the total resources in the world. This can be set with SetTotalRes()

`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<SymConfigBase> my_config = NULL`

Purpose: Represents the configuration settings for a particular run.

`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_uninf_hosts`

`emp::Ptr<emp::DataMonitor<int>> data_node_attempts_horiztrans`

`emp::Ptr<emp::DataMonitor<int>> data_node_successes_horiztrans`

`emp::Ptr<emp::DataMonitor<int>> data_node_attempts_verttrans`

## 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\_sourceCatchCatch.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<br/>  
→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\_sourceCatchCatch.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<br/>  
→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\_sourceCatchCatch.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 “Symbolation” from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareEqual(LhsT const &lhs, RhsT<br/>  
→ 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 “Symbolation” from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT<br/>  
→ &&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 “Symbolation” from directory: ./doxyoutput/xml. Potential matches:

```
- template<typename LhsT, typename RhsT> auto compareNotEqual(LhsT const &lhs, RhsT<br/>  
→ &&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\_sourceCatchCatch.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<br/>  
→&&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\_sourceCatchCatch.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<br/>  
→&&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\_sourceCatchCatch.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<br>→ &&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\_catchCatch.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::convertUnstreamable(T const&)
```

### Template Function `Catch::Detail::convertUnstreamable(T const&)`

- Defined in file\_source\_catchCatch.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::convertUnstreamable(T const&)
```

### Template Function `Catch::Detail::convertUnstreamable(T const&)`

- Defined in file\_source\_catchCatch.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::convertUnstreamable(T const&)
```

### Template Function `Catch::Detail::rangeToString`

- Defined in file\_source\_catchCatch.hpp

#### Function Documentation

```
template<typename InputIterator, typename Sentinel = InputIterator>
std::string Catch::Detail::rangeToString(InputIterator first, Sentinel last)
```

## Function Catch::Detail::rawMemoryToString(const void \*, std::size\_t)

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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: ./doxygenoutput/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: ./doxygenoutput/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, ICConfig 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\_sourceCatchCatch.hpp

## Function Documentation

```
auto Catch::Generators::acquireGeneratorTracker(StringRef generatorName, SourceLine-
Info const &lineInfo) -> IGenerator-
Tracker&
```

### Template Function Catch::Generators::chunk

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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 (StringRef generatorName, SourceLineInfo const
  &lineInfo, L const &generatorExpression) -> decltype(std::declval<decltype(generatorExpression())>().get())
```

## Template Function Catch::Generators::makeGenerators(GeneratorWrapper<T>&&, Gs&&...)

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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 `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 `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 `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\_sourceCatchCatch.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: ./doxygenoutput/xml. Potential matches:

```
- template<typename T> void handleExpression(ExprLhs<T> const &expr)
- void handleExpression(ITransientExpression const &expr)
```

## Function Catch::isFalseTest

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
bool Catch::isFalseTest (int flags)
```

## Function Catch::isJustInfo

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
bool Catch::isJustInfo (int flags)
```

## Function Catch::isOk

- Defined in file\_sourceCatchCatch.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, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>
Vector::ApproxMatcher<T, AllocComp, AllocMatch> Catch::Matchers::Approx(std::vector<T, AllocComp> 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, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>
→ Vector::ContainsMatcher<T, AllocComp, AllocMatch> Contains(std::vector<T, AllocComp> const &comparator)
```

## Template Function `Catch::Matchers::Contains(std::vector<T, AllocComp> const&)`

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

**Warning:** doxygenfunction: Unable to resolve multiple matches for function “`Catch::Matchers::Contains`” with arguments (`std::vector<T, AllocComp> 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, typename AllocComp = std::allocator<T>, typename AllocMatch_
  ↵= AllocComp> Vector::ContainsMatcher<T, AllocComp, AllocMatch>_
  ↵Contains(std::vector<T, AllocComp> const &comparator)
```

## Function `Catch::Matchers::EndsWith`

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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, typename AllocComp = std::allocator<T>, typename AllocMatch_
  ↵= AllocComp> Vector::EqualsMatcher<T, AllocComp, AllocMatch> Equals(std::vector<T,
  ↵ AllocComp> const &comparator)
```

## Template Function `Catch::Matchers::Equals(std::vector<T, AllocComp> const&)`

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

**Warning:** doxygenfunction: Unable to resolve multiple matches for function “`Catch::Matchers::Equals`” with arguments (`std::vector<T, AllocComp> 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, typename AllocComp = std::allocator<T>, typename AllocMatch_
  ↵= AllocComp> Vector::EqualsMatcher<T, AllocComp, AllocMatch> Equals(std::vector<T,
  ↵ AllocComp> const &comparator)
```

## Function `Catch::Matchers::Generic::Detail::finalizeDescription`

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
std::string Catch::Matchers::Generic::Detail::finalizeDescription(const std::string
  &desc)
```

## Function `Catch::Matchers::Matches`

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
StdString::RegexMatcher Catch::Matchers::Matches(std::string const &regex, CaseSensitive::Choice
  ↵caseSensitivity = CaseSensitive::Yes)
```

## Function `Catch::Matchers::Message`

- Defined in file\_sourceCatchCatch.hpp

## Function Documentation

Exception::*ExceptionMessageMatcher* Catch::Matchers::**Message** (std::string **const** &*message*)

### Template Function Catch::Matchers::Predicate

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.hpp

## Function Documentation

```
template<typename T, typename AllocComp = std::allocator<T>, typename AllocMatch = AllocComp>
Vector::UnorderedEqualsMatcher<T, AllocComp, AllocMatch> Catch::Matchers::UnorderedEquals (std::vector<T,
AllocComp> const &tar, get)
```

## Template Function Catch::Matchers::VectorContains

- Defined in file\_source\_catch\_catch.hpp

### Function Documentation

```
template<typename T, typename Alloc = std::allocator<T>>
Vector::ContainsElementMatcher<T, Alloc> Catch::Matchers::VectorContains(T const &com-
parator)
```

## 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\_sourceCatchCatch.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: ./doxygenoutput/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\_sourceCatchCatch.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: ./doxygenoutput/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\_sourceCatchCatch.hpp

#### Function Documentation

```
bool Catch::matchTest (TestCase const &testCase, TestSpec const &testSpec, IConfig const &config)
```

## Function Catch::operator""\_sr

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
constexpr auto Catch::operator""_sr(char const *rawChars, std::size_t size) noexcept ->
    StringRef
```

## Template Function Catch::operator+

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
template<typename T>
T const &Catch::operator+ (T const &value, StreamEndStop)
```

## Function Catch::operator+=

- Defined in file\_sourceCatchCatch.hpp

### Function Documentation

```
auto Catch::operator+= (std::string &lhs, StringRef const &sr) -> std::string&
```

## Function Catch::operator<<(std::ostream&, SourceLineInfo const&)

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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: ./doxygenoutput/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\_sourceCatchCatch.hpp

#### Function Documentation

*ResultDisposition::Flags* Catch::operator| (*ResultDisposition::Flags lhs*, *ResultDisposition::Flags rhs*)

### Template Function Catch::rangeToString(Range const&)

- Defined in file\_sourceCatchCatch.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: ./doxygenoutput/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\_sourceCatchCatch.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 CheckConfigFile

- Defined in file\_source\_native\_symbulation.h

### Function Documentation

void **CheckConfigFile** (SymConfigBase &*config*, int *argc*, char \**argv*[])

Input: The SymConfig object and the command line arguments.

Output: None

Purpose: To validate the passed config settings and throw appropriate error messages.

## 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 LysisCheckConfigFile

- Defined in file\_source\_native\_symbulation\_lysis.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “LysisCheckConfigFile” 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\_sourceCatchCatch.hpp

## Function Documentation

```
constexpr auto operator""_catch_sr(char const *rawChars, std::size_t size) noexcept ->
    Catch::StringRef
```

### Function operator<<

- Defined in file\_sourceCatchCatch.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 “symbolation\_main” in doxygen xml output for project “Symbolation” 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 “Symbolation” 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 “Symbolation” 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 “Symbolation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“GetHostIntValDataNode”, “”)

- 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(“GetSymInfectChanceDataNode”, “”)

- 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(“GetFreeSymInfectChanceDataNode”, “”)

- 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(“GetHostedSymInfectChanceDataNode”, “”)

- 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(“GetHorizontalTransmissionAttemptCount”, “”)

- 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“GetHorizontalTransmissionSuccessCount”, “”)

- 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“GetVerticalTransmissionAttemptCount”, “”)

- 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 “Symbolabation” 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 “Symbolabation” 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(“**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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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(“AddSymbiont”, “”)

- 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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(“Limited resources inflow”, “”)

- 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 without free living symbionts”, “”)

- 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 with free living symbionts”, “”)

- 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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(“Spatial structure”, “”)

- 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“SetMutationZero”, “”)

- 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“No mutation updates”, “ ”)

- 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(“IsInboundsPos”, “”)

- 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(“InjectHost”, “”)

- 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(“Setup”, “”)

- 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(“SetupSymbionts”, “”)

- 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“SetupHosts”, “”)

- 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“GetEfficiencyDataNode”, “”)

- Defined in file\_source\_test\_efficient\_mode\_test\_EfficientDataNodes.test.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “TEST\_CASE” in doxygen xml output for project “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“EfficientSymbiont VerticalTransmission”, “”)

- 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“Efficient SetupSymbionts”, “”)

- Defined in file\_source\_test\_efficient\_mode\_test\_EfficientWorld.test.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “TEST\_CASE” in doxygen xml output for project “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“Efficient SetupHosts”, “”)

- Defined in file\_source\_test\_efficient\_mode\_test\_EfficientWorld.test.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “TEST\_CASE” in doxygen xml output for project “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“Efficient Setup”, “”)

- Defined in file\_source\_test\_efficient\_mode\_test\_EfficientWorld.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(“Horizontal Mutation Rate Results”, “”)

- Defined in file\_source\_test\_integration\_test\_dirty\_transmission\_hz\_mut\_rate.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(“Resource Distribution Results”, “”)

- Defined in file\_source\_test\_integration\_test\_endosymbiosis\_res\_distribute.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(“Prophage Loss Rate Results”, “”)

- Defined in file\_source\_test\_integration\_test\_lysoxygen\_plr.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(“Multi-infection results”, “”)

- Defined in file\_source\_test\_integration\_test\_multi\_infection\_pgg.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 Results”, “”)

- Defined in file\_source\_test\_integration\_test\_spatial\_structure\_vt.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 overwrites Symbiont ProcessResources”, “”)

- 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(“GetCFUDataNode”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisDataNodes.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(“GetLysisChanceDataNode”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisDataNodes.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(“GetInductionChanceDataNode”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisDataNodes.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(“GetBurstSizeDataNode”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisDataNodes.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(“GetBurstCountDataNode”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisDataNodes.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(“GetIncorporationDifferenceDataNode”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisDataNodes.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(“Lysis mode Update()”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisWorld.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(“Lysis SetupSymbionts”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisWorld.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(“Lysis SetupHosts”, “”)

- Defined in file\_source\_test\_lysis\_mode\_test\_LysisWorld.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(“GetPGGDataNode”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGDataNodes.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 “Symbolabation” 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 “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“PGGSymbiont, GetHost”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGHostPGGSymbiontUnitTest.test.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “TEST\_CASE” in doxygen xml output for project “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“PGGHost DistribResources”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGHostPGGSymbiontUnitTest.test.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “TEST\_CASE” in doxygen xml output for project “Symbolabation” from directory: ./doxyoutput/xml

### Function TEST\_CASE(“PGGVertical Transmission of Symbiont”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGHostPGGSymbiontUnitTest.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\_PGGHostPGGSymbiontUnitTest.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\_PGGHostPGGSymbiontUnitTest.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\_PGGSymbiont.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\_PGGSymbiont.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\_PGGSymbiont.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\_PGGSymbiont.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\_PGGSymbiont.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\_PGGSymbiont.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(“PGG Interaction Patterns”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGWorld.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(“PGG SetupSymbionts”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGWorld.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(“PGG SetupHosts”, “”)

- Defined in file\_source\_test\_pgg\_mode\_test\_PGGWorld.test.cc

## Function Documentation

**Warning:** doxygenfunction: Cannot find function “TEST\_CASE” 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::Detail::unprintableString

- Defined in file\_source\_catch\_catch.hpp

## Variable Documentation

**const** std::string Catch::Detail::**unprintableString**

### 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\_catchCatch.hpp

### Define Documentation

**AND\_WHEN** (*desc*)

### Define ANON\_TEST\_CASE

- Defined in file\_source\_catchCatch.hpp

### Define Documentation

**ANON\_TEST\_CASE** ()

### Define CAPTURE

- Defined in file\_source\_catchCatch.hpp

### Define Documentation

**CAPTURE** (...)

### Define CATCH\_CATCH\_ALL

- Defined in file\_source\_catchCatch.hpp

### Define Documentation

**CATCH\_CATCH\_ALL**

### Define CATCH\_CATCH\_ANON

- Defined in file\_source\_catchCatch.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\_sourceCatch\_main.cc

### Define Documentation

**Warning:** doxygen define: 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\_sourceCatchCatch.hpp

### Define Documentation

#### CATCH\_CONFIG\_POSIX\_SIGNALS

### Define CATCH\_CONFIG\_WCHAR

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

#### CATCH\_CONFIG\_WCHAR

### Define CATCH\_DEFER

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

#### CATCH\_DEFER (*id*)

### Define CATCH\_EMPTY

- Defined in file\_sourceCatchCatch.hpp

## Define Documentation

`CATCH_EMPTY()`

## Define `CATCH_FORCE`

- Defined in `file_source_catch_catch.hpp`

## Define Documentation

`CATCH_FORCE` (*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\_catchCatch.hpp

**Define Documentation****CATCH\_INTERNAL\_STOP\_WARNINGS\_SUPPRESSION****Define CATCH\_INTERNAL\_STRINGIFY**

- Defined in file\_source\_catchCatch.hpp

**Define Documentation****CATCH\_INTERNAL\_STRINGIFY(...)****Define CATCH\_INTERNAL\_SUPPRESS\_GLOBALS\_WARNINGS**

- Defined in file\_source\_catchCatch.hpp

**Define Documentation****CATCH\_INTERNAL\_SUPPRESS\_GLOBALS\_WARNINGS****Define CATCH\_INTERNAL\_SUPPRESS\_PARENTHESSES\_WARNINGS**

- Defined in file\_source\_catchCatch.hpp

**Define Documentation****CATCH\_INTERNAL\_SUPPRESS\_PARENTHESSES\_WARNINGS**

## **Define CATCH\_INTERNAL\_SUPPRESS\_UNUSED\_TEMPLATE\_WARNINGS**

- Defined in file\_source\_catchCatch.hpp

### **Define Documentation**

**CATCH\_INTERNAL\_SUPPRESS\_UNUSED\_TEMPLATE\_WARNINGS**

## **Define CATCH\_INTERNAL\_SUPPRESS\_UNUSED\_WARNINGS**

- Defined in file\_source\_catchCatch.hpp

### **Define Documentation**

**CATCH\_INTERNAL\_SUPPRESS\_UNUSED\_WARNINGS**

## **Define CATCH\_INTERNAL\_SUPPRESS\_ZERO\_VARIADIC\_WARNINGS**

- Defined in file\_source\_catchCatch.hpp

### **Define Documentation**

**CATCH\_INTERNAL\_SUPPRESS\_ZERO\_VARIADIC\_WARNINGS**

## **Define CATCH\_MAKE\_MSG**

- Defined in file\_source\_catchCatch.hpp

### **Define Documentation**

**CATCH\_MAKE\_MSG(...)**

## **Define CATCH\_REC\_END**

- Defined in file\_source\_catchCatch.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\_catchCatch.hpp

#### Define Documentation

**CATCH\_REC\_LIST0** (*f, x, peek, ...*)

### Define CATCH\_REC\_LIST0\_UD

- Defined in file\_source\_catchCatch.hpp

#### Define Documentation

**CATCH\_REC\_LIST0\_UD** (*f, userdata, x, peek, ...*)

### Define CATCH\_REC\_LIST1

- Defined in file\_source\_catchCatch.hpp

#### Define Documentation

**CATCH\_REC\_LIST1** (*f, x, peek, ...*)

### Define CATCH\_REC\_LIST1\_UD

- Defined in file\_source\_catchCatch.hpp

#### Define Documentation

**CATCH\_REC\_LIST1\_UD** (*f, userdata, x, peek, ...*)

### Define CATCH\_REC\_LIST2

- Defined in file\_source\_catchCatch.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\_RECUSION\_LEVEL0

- Defined in file\_source\_catch\_catch.hpp

### Define Documentation

**CATCH\_RECUSION\_LEVEL0** (...)

### Define CATCH\_RECUSION\_LEVEL1

- Defined in file\_source\_catch\_catch.hpp

## Define Documentation

`CATCH_RECUSION_LEVEL1(...)`

## Define `CATCH_RECUSION_LEVEL2`

- Defined in `file_source_catch_catch.hpp`

## Define Documentation

`CATCH_RECUSION_LEVEL2(...)`

## Define `CATCH_RECUSION_LEVEL3`

- Defined in `file_source_catch_catch.hpp`

## Define Documentation

`CATCH_RECUSION_LEVEL3(...)`

## Define `CATCH_RECUSION_LEVEL4`

- Defined in `file_source_catch_catch.hpp`

## Define Documentation

`CATCH_RECUSION_LEVEL4(...)`

## Define `CATCH_RECUSION_LEVEL5`

- Defined in `file_source_catch_catch.hpp`

## Define Documentation

`CATCH_RECUSION_LEVEL5(...)`

### Define CATCH\_REGISTER\_ENUM

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

**CATCH\_REGISTER\_ENUM** (*enumName*, ...)

### Define CATCH\_REGISTER\_TAG\_ALIAS

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

**CATCH\_REGISTER\_TAG\_ALIAS** (*alias*, *spec*)

### Define CATCH\_RUNTIME\_ERROR

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

**CATCH\_RUNTIME\_ERROR** (...)

### Define CATCH\_TRANSLATE\_EXCEPTION

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

**CATCH\_TRANSLATE\_EXCEPTION** (*signature*)

### Define CATCH\_TRY

- Defined in file\_sourceCatchCatch.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:** doxygen: 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\_catchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X` (*TestName*, *signature*, ...)

### Define INTERNAL\_CATCH\_DEFINE\_SIG\_TEST\_X

- Defined in file\_source\_catchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_DEFINE_SIG_TEST_X` (*TestName*, *signature*, ...)

### Define INTERNAL\_CATCH\_DYNAMIC\_SECTION

- Defined in file\_source\_catchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_DYNAMIC_SECTION` (...)

### Define INTERNAL\_CATCH\_ELSE

- Defined in file\_source\_catchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_ELSE` (*macroName*, *resultDisposition*, ...)

### Define INTERNAL\_CATCH\_EXPAND1

- Defined in file\_source\_catchCatch.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\_sourceCatchCatch.hpp

### Define Documentation

`INTERNAL_CATCH_STRINGIZE_WITHOUT_PARENS (param)`

## Define INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

`INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE (Name, Tags, TmplList)`

## Define INTERNAL\_CATCH\_TEMPLATE\_LIST\_TEST\_CASE\_2

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.hpp

## Define Documentation

**INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE** (*Name*, *Tags*, ...)

## Define INTERNAL\_CATCH\_TEMPLATE\_PRODUCT\_TEST\_CASE2

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.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\_sourceCatchCatch.hpp

### Define Documentation

`INTERNAL_CATCH_TEMPLATE_PRODUCT_TEST_CASE_SIG (Name, Tags, Signature, ...)`

## Define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

`INTERNAL_CATCH_TEMPLATE_TEST_CASE (Name, Tags, ...)`

## Define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_2

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

`INTERNAL_CATCH_TEMPLATE_TEST_CASE_2 (TestName, TestFunc, Name, Tags, Signature, ...)`

## Define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD

- Defined in file\_sourceCatchCatch.hpp

**Define Documentation**

**INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD** (*ClassName*, *Name*, *Tags*, ...)

**Define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_2**

- Defined in file\_sourceCatchCatch.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\_sourceCatchCatch.hpp

**Define Documentation**

**INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_METHOD\_SIG** (*ClassName*, *Name*, *Tags*, *Signature*, ...)

**Define INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_SIG**

- Defined in file\_sourceCatchCatch.hpp

**Define Documentation**

**INTERNAL\_CATCH\_TEMPLATE\_TEST\_CASE\_SIG** (*Name*, *Tags*, *Signature*, ...)

**Define INTERNAL\_CATCH\_TEST**

- Defined in file\_sourceCatchCatch.hpp

**Define Documentation**

**INTERNAL\_CATCH\_TEST** (*macroName*, *resultDisposition*, ...)

### Define INTERNAL\_CATCH\_TEST\_CASE\_METHOD

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_TEST_CASE_METHOD (ClassName, ...)`

### Define INTERNAL\_CATCH\_TEST\_CASE\_METHOD2

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_TEST_CASE_METHOD2 (TestName, ClassName, ...)`

### Define INTERNAL\_CATCH\_TESTCASE

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_TESTCASE (...)`

### Define INTERNAL\_CATCH\_TESTCASE2

- Defined in file\_sourceCatchCatch.hpp

#### Define Documentation

`INTERNAL_CATCH_TESTCASE2 (TestName, ...)`

### Define INTERNAL\_CATCH\_THROWS

- Defined in file\_sourceCatchCatch.hpp

## Define Documentation

**INTERNAL\_CATCH\_THROWS** (*macroName*, *resultDisposition*, ...)

## Define INTERNAL\_CATCH\_THROWS\_AS

- Defined in file\_source\_catchCatch.hpp

## Define Documentation

**INTERNAL\_CATCH\_THROWS\_AS** (*macroName*, *exceptionType*, *resultDisposition*, *expr*)

## Define INTERNAL\_CATCH\_THROWS\_MATCHES

- Defined in file\_source\_catchCatch.hpp

## Define Documentation

**INTERNAL\_CATCH\_THROWS\_MATCHES** (*macroName*, *exceptionType*, *resultDisposition*, *matcher*, ...)

## Define INTERNAL\_CATCH\_THROWS\_STR\_MATCHES

- Defined in file\_source\_catchCatch.hpp

## Define Documentation

**INTERNAL\_CATCH\_THROWS\_STR\_MATCHES** (*macroName*, *resultDisposition*, *matcher*, ...)

## Define INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION

- Defined in file\_source\_catchCatch.hpp

## Define Documentation

**INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION** (*signature*)

## Define INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION2

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

**INTERNAL\_CATCH\_TRANSLATE\_EXCEPTION2** (*translatorName, signature*)

## Define INTERNAL\_CATCH\_TRY

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

**INTERNAL\_CATCH\_TRY**

## Define INTERNAL\_CATCH\_TYPE\_GEN

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

**INTERNAL\_CATCH\_TYPE\_GEN**

## Define INTERNAL\_CATCH\_UNIQUE\_NAME

- Defined in file\_sourceCatchCatch.hpp

### Define Documentation

**INTERNAL\_CATCH\_UNIQUE\_NAME** (*name*)

## Define INTERNAL\_CATCH\_UNIQUE\_NAME\_LINE

- Defined in file\_sourceCatchCatch.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:** doxygen define: 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:** doxygen define: 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:** doxygen define: 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

## Typeface Documentation

```
using Catch::exceptionTranslateFunction = std::string (*)()
```

## Typeface Catch::ExceptionTranslators

- Defined in file\_source\_catch\_catch.hpp

## Typeface Documentation

```
using Catch::ExceptionTranslators = std::vector<std::unique_ptr<IExceptionTranslator const>>
```

## Typeface Catch::FunctionReturnType

- Defined in file\_source\_catch\_catch.hpp

## Typeface Documentation

```
using Catch::FunctionReturnType = typename std::remove_reference<typename std::remove_cv<typename std::remove
```

## Typeface Catch::Generators::GeneratorBasePtr

- Defined in file\_source\_catch\_catch.hpp

## Typeface Documentation

```
using Catch::Generators::GeneratorBasePtr = std::unique_ptr<GeneratorUntypedBase>
```

## Typeface Catch::IConfigPtr

- Defined in file\_source\_catch\_catch.hpp

## Typeface 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>
```

## 2.6 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](#).

### 2.6.1 One-time Preparation

- Get a [GitHub](#) account.

(We use GitHub to manage Symbulation contributions.)

- Fork [github.com/anyaevostiar/SymbulationEmp](https://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).

- 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!

### 2.6.2 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.)

### 2.6.3 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.

### 2.6.4 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](#).

## 2.6.5 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
```

## 2.7 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

### 2.7.1 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 leveled. 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.

## 2.7.2 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.

## 2.7.3 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 automatically 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.

## 2.8 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.

### 2.8.1 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
```

## 2.8.2 Building Documentation Locally

You are then ready to make your local documentation and run it:

```
make html
cd _build/html/
python3 -m http.server
```

## 2.8.3 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.

---

## 2.8.4 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](#).

## 2.8.5 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
```

```

## 2.9 Guide to Testing in Symbulation

This document details how testing works in Symbulation, both for writing and understanding tests.

### 2.9.1 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
./symbulation.test
=====
All tests passed (592 assertions in 70 test cases)
```

### 2.9.2 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

## A

AND\_GIVEN (*C macro*), 227  
AND\_THEN (*C macro*), 227  
AND\_WHEN (*C macro*), 228  
ANON\_TEST\_CASE (*C macro*), 228

## B

Bacterium (*C++ class*), 85  
Bacterium::Bacterium (*C++ function*), 85  
Bacterium::GetIncVal (*C++ function*), 86  
Bacterium::GetName (*C++ function*), 85  
Bacterium::host\_incorporation\_val (*C++ member*), 86  
Bacterium::MakeNew (*C++ function*), 86  
Bacterium::Mutate (*C++ function*), 86  
Bacterium::my\_world (*C++ member*), 86  
Bacterium::ProcessLysogenResources (*C++ function*), 86  
Bacterium::SetIncVal (*C++ function*), 86

## C

CAPTURE (*C macro*), 228  
Catch::always\_false (*C++ struct*), 37  
Catch::AssertionHandler (*C++ class*), 87  
Catch::AssertionHandler::~AssertionHandler (*C++ function*), 87  
Catch::AssertionHandler::allowThrows (*C++ function*), 87  
Catch::AssertionHandler::AssertionHandler (*C++ function*), 87  
Catch::AssertionHandler::complete (*C++ function*), 87  
Catch::AssertionHandler::handleExceptionNotThrownAsExpected (*C++ function*), 87  
Catch::AssertionHandler::handleExceptionThrownAsExpected (*C++ function*), 87  
Catch::AssertionHandler::handleExpr (*C++ function*), 87  
Catch::AssertionHandler::handleMessage (*C++ function*), 87  
Catch::AssertionHandler::handleThrowingCallsSkipped (*C++ function*), 87  
Catch::AssertionHandler::handleUnexpectedException (*C++ function*), 87  
Catch::AssertionHandler::handleUnexpectedInflightException (*C++ function*), 87  
Catch::AssertionHandler::setCompleted (*C++ function*), 87  
Catch::AssertionInfo (*C++ struct*), 37  
Catch::AssertionInfo::capturedExpression (*C++ member*), 37  
Catch::AssertionInfo::lineInfo (*C++ member*), 37  
Catch::AssertionInfo::macroName (*C++ member*), 37  
Catch::AssertionInfo::resultDisposition (*C++ member*), 37  
Catch::AssertionReaction (*C++ struct*), 37  
Catch::AssertionReaction::shouldDebugBreak (*C++ member*), 37  
Catch::AssertionReaction::shouldThrow (*C++ member*), 37  
Catch::AutoReg (*C++ struct*), 38  
Catch::AutoReg::~AutoReg (*C++ function*), 38  
Catch::AutoReg::AutoReg (*C++ function*), 38  
Catch::BinaryExpr (*C++ class*), 87  
Catch::BinaryExpr::BinaryExpr (*C++ function*), 88  
Catch::BinaryExpr::operator!= (*C++ function*), 88  
Catch::BinaryExpr::operator== (*C++ function*), 88  
Catch::BinaryExpr::operator&& (*C++ function*), 88  
Catch::BinaryExpr::operator|| (*C++ function*), 88  
Catch::BinaryExpr::operator> (*C++ function*), 88  
Catch::BinaryExpr::operator>= (*C++ function*), 88  
Catch::BinaryExpr::operator< (*C++ function*), 88  
Catch::BinaryExpr::operator<= (*C++ function*), 88

```

Catch::Capturer (C++ class), 88
Catch::Capturer::~Capturer (C++ function), 88
Catch::Capturer::Capturer (C++ function), 88
Catch::Capturer::captureValue (C++ function), 88
Catch::Capturer::captureValues (C++ function), 88
Catch::CaseSensitive (C++ struct), 38
Catch::CaseSensitive::Choice (C++ enum), 38
Catch::CaseSensitive::Choice::No (C++ enumerator), 38
Catch::CaseSensitive::Choice::Yes (C++ enumerator), 38
Catch::cerr (C++ function), 148
Catch::cleanUp (C++ function), 148
Catch::cleanUpContext (C++ function), 148
Catch::clog (C++ function), 149
Catch::contains (C++ function), 153
Catch::Counts (C++ struct), 39
Catch::Counts::allok (C++ function), 39
Catch::Counts::allPassed (C++ function), 39
Catch::Counts::failed (C++ member), 39
Catch::Counts::failedButOk (C++ member), 39
Catch::Counts::operator+= (C++ function), 39
Catch::Counts::operator- (C++ function), 39
Catch::Counts::passed (C++ member), 39
Catch::Counts::total (C++ function), 39
Catch::cout (C++ function), 153
Catch::Decomposer (C++ struct), 39
Catch::Decomposer::operator<= (C++ function), 39
Catch::Detail::Approx (C++ class), 89
Catch::Detail::Approx::Approx (C++ function), 89
Catch::Detail::Approx::custom (C++ function), 89
Catch::Detail::Approx::epsilon (C++ function), 89
Catch::Detail::Approx::margin (C++ function), 89
Catch::Detail::Approx::operator!= (C++ function), 89
Catch::Detail::Approx::operator() (C++ function), 89
Catch::Detail::Approx::operator== (C++ function), 89
Catch::Detail::Approx::operator- (C++ function), 89
Catch::Detail::Approx::operator>= (C++ function), 89
Catch::Detail::Approx::operator<= (C++ function), 89
Catch::Detail::Approx::scale (C++ function), 89
Catch::Detail::Approx::toString (C++ function), 89
Catch::Detail::convertUnknownEnumToString (C++ function), 153
Catch::Detail::convertUnstreamable (C++ function), 154
Catch::Detail::EnumInfo (C++ struct), 40
Catch::Detail::EnumInfo::~EnumInfo (C++ function), 40
Catch::Detail::EnumInfo::lookup (C++ function), 40
Catch::Detail::EnumInfo::m_name (C++ member), 40
Catch::Detail::EnumInfo::m_values (C++ member), 40
Catch::detail::is_range_impl (C++ struct), 40
Catch::detail::is_range_impl<T, typename void_type<decltype(begin(std::declval<T>))> (C++ struct), 41
Catch::Detail::IsStreamInsertable (C++ class), 90
Catch::Detail::IsStreamInsertable::value (C++ member), 90
Catch::Detail::rangeToString (C++ function), 154
Catch::Detail::stringify (C++ function), 155
Catch::Detail::unprintableString (C++ member), 227
Catch::detail::void_type (C++ struct), 41
Catch::detail::void_type::type (C++ type), 41
Catch::exceptionTranslateFunction (C++ type), 276
Catch::ExceptionTranslatorRegistrar (C++ class), 90
Catch::ExceptionTranslatorRegistrar::ExceptionTranslator (C++ class), 91
Catch::ExceptionTranslatorRegistrar::ExceptionTranslator (C++ function), 91
Catch::ExceptionTranslatorRegistrar::ExceptionTranslator (C++ member), 91
Catch::ExceptionTranslatorRegistrar::ExceptionTranslator (C++ function), 91
Catch::ExceptionTranslatorRegistrar::ExceptionTranslator (C++ function), 91
Catch::ExceptionTranslatorRegistrar::ExceptionTranslator (C++ function), 90
Catch::ExceptionTranslators (C++ type), 276
Catch::ExprLhs (C++ class), 91
Catch::ExprLhs::ExprLhs (C++ function), 91
Catch::ExprLhs::makeUnaryExpr (C++ function), 92

```

Catch::ExprLhs::operator!= (*C++ function*), 91  
 Catch::ExprLhs::operator== (*C++ function*), 91  
 Catch::ExprLhs::operator& (*C++ function*), 92  
 Catch::ExprLhs::operator&& (*C++ function*), 92  
 Catch::ExprLhs::operator^ (*C++ function*), 92  
 Catch::ExprLhs::operator| (*C++ function*), 92  
 Catch::ExprLhs::operator|| (*C++ function*), 92  
 Catch::ExprLhs::operator> (*C++ function*), 91  
 Catch::ExprLhs::operator>= (*C++ function*), 92  
 Catch::ExprLhs::operator< (*C++ function*), 92  
 Catch::ExprLhs::operator<= (*C++ function*), 92  
 Catch::filterTests (*C++ function*), 156  
 Catch::formatReconstructedExpression (*C++ function*), 157  
 Catch::FunctionReturnType (*C++ type*), 276  
 Catch::GeneratorException (*C++ class*), 92  
 Catch::GeneratorException::GeneratorException::what (*C++ function*), 92  
 Catch::GeneratorException::what (*C++ function*), 92  
 Catch::Generators::acquireGeneratorTrack::what (*C++ function*), 157  
 Catch::Generators::as (*C++ struct*), 41  
 Catch::Generators::chunk (*C++ function*), 157  
 Catch::Generators::ChunkGenerator (*C++ class*), 93  
 Catch::Generators::ChunkGenerator::ChunkGenerator (*C++ function*), 93  
 Catch::Generators::FilterGenerator (*C++ class*), 93  
 Catch::Generators::FilterGenerator::FilterGenerator (*C++ function*), 94  
 Catch::Generators::FilterGenerator::get (*C++ function*), 94  
 Catch::Generators::FilterGenerator::nextCatch::Generators::map (*C++ function*), 161  
 Catch::Generators::FixedValuesGenerator (*C++ class*), 94  
 Catch::Generators::FixedValuesGenerator::FixedValuesGenerator (*C++ function*), 94  
 Catch::Generators::FixedValuesGenerator::get (*C++ function*), 94  
 Catch::Generators::FixedValuesGenerator::next  
 Catch::Generators::FixedValuesGenerator::next  
 Catch::Generators::GeneratorBasePtr (*C++ type*), 276  
 Catch::Generators::Generators (*C++ class*), 95  
 Catch::Generators::Generators::Generators (*C++ function*), 95  
 Catch::Generators::Generators::get (*C++ function*), 95  
 Catch::Generators::Generators::next (*C++ function*), 95  
 Catch::Generators::GeneratorUntypedBase (*C++ class*), 95  
 Catch::Generators::GeneratorUntypedBase::~GeneratorUntypedBase (*C++ function*), 95  
 Catch::Generators::GeneratorUntypedBase::GeneratorUntypedBase (*C++ function*), 95  
 Catch::Generators::GeneratorUntypedBase::next (*C++ function*), 95  
 Catch::Generators::GeneratorWrapper::GeneratorWrapper (*C++ class*), 96  
 Catch::Generators::GeneratorWrapper::GeneratorWrapper (*C++ function*), 96  
 Catch::Generators::GeneratorWrapper::GeneratorWrapper (*C++ function*), 96  
 Catch::Generators::IGenerator (*C++ struct*), 42  
 Catch::Generators::IGenerator::~IGenerator (*C++ function*), 42  
 Catch::Generators::IGenerator::get (*C++ function*), 42  
 Catch::Generators::IteratorGenerator (*C++ class*), 96  
 Catch::Generators::IteratorGenerator::IteratorGenerator (*C++ function*), 96  
 Catch::Generators::MapGenerator (*C++ class*), 97  
 Catch::Generators::MapGenerator::get  
 Catch::Generators::MapGenerator::MapGenerator  
 Catch::Generators::MapGenerator::MapGenerator (*C++ function*), 97  
 Catch::Generators::MapGenerator::next

(*C++ function*), 97  
Catch::Generators::pf::make\_unique (*C++ function*), 161  
Catch::Generators::RandomFloatingGenerator (*C++ class*), 97  
Catch::Generators::RandomFloatingGenerator::getton (*C++ function*), 98  
Catch::Generators::RandomFloatingGenerator::getton, 164  
Catch::Generators::RandomIntegerGenerator (*C++ class*), 98  
Catch::Generators::RandomIntegerGenerator::getton, 164  
Catch::Generators::RandomIntegerGenerator::getton, 164  
Catch::Generators::RandomIntegerGenerator::getton, 164  
Catch::Generators::RandomIntegerGenerator::getton, 164  
Catch::Generators::RandomIntegerGenerator::getton, 164  
Catch::Generators::RandomIntegerGenerator::getton, 164  
Catch::Generators::RangeGenerator (*C++ class*), 99  
Catch::Generators::RangeGenerator::get (*C++ function*), 99  
Catch::Generators::RangeGenerator::next (*C++ function*), 99  
Catch::Generators::RangeGenerator::next, 43  
Catch::Generators::RangeGenerator::next, 43  
Catch::Generators::repeat (*C++ function*), 163  
Catch::Generators::RepeatGenerator (*C++ class*), 99  
Catch::Generators::RepeatGenerator::get (*C++ function*), 99  
Catch::Generators::RepeatGenerator::next, 43  
Catch::Generators::RepeatGenerator::next, 43  
Catch::Generators::RepeatGenerator::Repeategerator::getTestsOrTags (*C++ function*), 43  
Catch::Generators::SingleValueGenerator (*C++ class*), 100  
Catch::Generators::SingleValueGenerator::get, 43  
Catch::Generators::SingleValueGenerator::get, 43  
Catch::Generators::SingleValueGenerator::get, 43  
Catch::Generators::SingleValueGenerator::get, 43  
Catch::Generators::table (*C++ function*), 163  
Catch::Generators::take (*C++ function*), 163  
Catch::Generators::TakeGenerator (*C++ class*), 100  
Catch::Generators::TakeGenerator::get (*C++ function*), 100  
Catch::Generators::TakeGenerator::next (*C++ function*), 100  
Catch::Generators::TakeGenerator::next, 43  
Catch::Generators::TakeGenerator::TakeGenerator::GetahorIConfig::testSpec (*C++ function*), 43  
Catch::Generators::value (*C++ function*), 163  
Catch::Generators::values (*C++ function*), 164  
Catch::getAllTestCasesSorted (*C++ func-*  
Catch::getCurrentContext (*C++ function*), 164  
Catch::getMutableContext (*C++ function*), 164  
Catch::RandomEloationNSeaseatndsSinceEpoch (*C++ function*), 165  
Catch::getEstimatedClockResolution (*C++ function*), 165  
Catch::getMutableRegistryHub (*C++ function*), 165  
Catch::getRegistryHub (*C++ function*), 165  
Catch::getResultCapture (*C++ function*), 165  
Catch::RandomehantdConfig::43  
Catch::IConfig::~IConfig (*C++ function*), 43  
Catch::IConfig::abortAfter (*C++ function*), 43  
Catch::IConfig::allowThrows (*C++ function*), 43  
Catch::IConfig::benchmarkConfidenceInterval (*C++ function*), 43  
Catch::IConfig::benchmarkNoAnalysis (*C++ function*), 43  
Catch::IConfig::benchmarkResamples (*C++ function*), 43  
Catch::IConfig::benchmarkSamples (*C++ function*), 43  
Catch::IConfig::benchmarkWarmupTime (*C++ function*), 43  
Catch::IConfig::getSectionsToRun (*C++ function*), 43  
Catch::IConfig::getTestsOrTags (*C++ function*), 43  
Catch::IConfig::hasTestFilters (*C++ function*), 43  
Catch::IConfig::includeSuccessfulResults (*C++ function*), 43  
Catch::IConfig::minDuration (*C++ function*), 43  
Catch::IConfig::rngSeed (*C++ function*), 43  
Catch::IConfig::runOrder (*C++ function*), 43  
Catch::IConfig::shouldDebugBreak (*C++ function*), 43  
Catch::IConfig::showDurations (*C++ function*), 43  
Catch::IConfig::showInvisibles (*C++ function*), 43  
Catch::IConfig::stream (*C++ function*), 43  
Catch::IConfig::testSpec (*C++ function*), 43

```
Catch::IConfig::useColour (C++ function), 43           (C++ function), 47
Catch::IConfig::verbosity (C++ function), 43          Catch::IMutableRegistryHub::getMutableEnumValuesRe...
Catch::IConfig::warnAboutMissingAssertions (C++ function), 43
Catch::IConfig::warnAboutNoTests (C++ function), 43
Catch::IConfigPtr (C++ type), 276
Catch::IContext (C++ struct), 44
Catch::IContext::~IContext (C++ function), 44
Catch::IContext::getConfig (C++ function), 44
Catch::IContext::getResultCapture (C++ function), 44
Catch::IContext::getRunner (C++ function), 44
Catch::IExceptionTranslator (C++ struct), 44
Catch::IExceptionTranslator::~IExceptionTranslator RegistryHub::getExceptionTranslatorRegistry (C++ function), 44
Catch::IExceptionTranslator::translate (C++ function), 44
Catch::IExceptionTranslatorRegistry (C++ struct), 45
Catch::IExceptionTranslatorRegistry::~IExceptionTranslator Registry (C++ function), 45
Catch::IExceptionTranslatorRegistry::translateAtTheRegistryHub (C++ function), 45
Catch::IGeneratorTracker (C++ struct), 45
Catch::IGeneratorTracker::~IGeneratorTracker (C++ function), 45
Catch::IGeneratorTracker::getGenerator (C++ function), 45
Catch::IGeneratorTracker::hasGenerator (C++ function), 45
Catch::IGeneratorTracker::setGenerator (C++ function), 45
Catch::IMutableContext (C++ struct), 46
Catch::IMutableContext::~IMutableContext (C++ function), 46
Catch::IMutableContext::setConfig (C++ function), 46
Catch::IMutableContext:: setResultCapture (C++ function), 46
Catch::IMutableContext::setRunner (C++ function), 46
Catch::IMutableEnumValuesRegistry (C++ struct), 46
Catch::IMutableEnumValuesRegistry::~IMutableEnumValues Registry (C++ function), 46
Catch::IMutableEnumValuesRegistry::registerEnumValue (C++ function), 46
Catch::IMutableRegistryHub (C++ struct), 46
Catch::IMutableRegistryHub::~IMutableRegistryHub (C++ function), 46
Catch::IMutableRegistryHub::registerListener (C++ function), 47
Catch::IMutableRegistryHub::registerReporter (C++ function), 47
Catch::IMutableRegistryHub::registerStartupException (C++ function), 47
Catch::IMutableRegistryHub::registerTagAlias (C++ function), 47
Catch::IMutableRegistryHub::registerTest (C++ function), 47
Catch::IMutableRegistryHub::registerTranslator (C++ function), 47
Catch::IRegistryHub (C++ struct), 47
Catch::IRegistryHub::~IRegistryHub (C++ function), 47
Catch::IRegistryHub::getReporterRegistry (C++ function), 47
Catch::IRegistryHub::getStartupExceptionRegistry (C++ function), 47
Catch::IRegistryHub::getTestCaseRegistry (C++ function), 47
Catch::IReporterFactoryPtr (C++ type), 277
Catch::IResultCapture (C++ struct), 47
Catch::IResultCapture::~IResultCapture (C++ function), 48
Catch::IResultCapture::acquireGeneratorTracker (C++ function), 48
Catch::IResultCapture::assertionPassed (C++ function), 48
Catch::IResultCapture::emplaceUnscopedMessage (C++ function), 48
Catch::IResultCapture::exceptionEarlyReported (C++ function), 48
Catch::IResultCapture::getCurrentTestName (C++ function), 48
Catch::IResultCapture::getLastResult (C++ function), 48
Catch::IResultCapture::handleExpr (C++ function), 48
Catch::IResultCapture::handleFatalErrorCondition (C++ function), 48
Catch::IResultCapture::handleIncomplete (C++ function), 48
Catch::IResultCapture::handleMessage (C++ function), 48
Catch::IResultCapture::handleNonExpr (C++ function), 48
```

```
Catch::IResultCapture::handleUnexpectedException (Not function), 52
(C++ function), 48
Catch::IResultCapture::handleUnexpectedInflight (Incomplete member), 52
(C++ function), 48
Catch::IResultCapture::lastAssertionPassed (C++ member), 52
(C++ function), 48
Catch::IResultCapture::popScopedMessage (C++ function), 52
(C++ function), 48
Catch::IResultCapture::pushScopedMessage (C++ function), 52
(C++ function), 48
Catch::IResultCapture::sectionEnded (C++ function), 48
Catch::IResultCapture::sectionEndedEarly (C++ function), 48
(C++ function), 48
Catch::IResultCapture::sectionStarted (C++ function), 48
Catch::IRunner (C++ struct), 48
Catch::IRunner::~IRunner (C++ function), 49
Catch::IRunner::aborting (C++ function), 49
Catch::is_callable (C++ struct), 49
Catch::is_callable_tester (C++ struct), 49
Catch::is_callable_tester::test (C++ function), 50
Catch::is_callable<Fun(Args...)> (C++ struct), 49
Catch::is_range (C++ struct), 50
Catch::isFalseTest (C++ function), 167
Catch::isJustInfo (C++ function), 167
Catch::isOk (C++ function), 167
Catch::isThrowSafe (C++ function), 168
Catch::IStream (C++ struct), 50
Catch::IStream::~IStream (C++ function), 50
Catch::IStream::stream (C++ function), 50
Catch::ITestCaseRegistry (C++ struct), 51
Catch::ITestCaseRegistry::~ITestCaseRegistry (C++ function), 51
Catch::ITestCaseRegistry::getAllTests (C++ function), 51
Catch::ITestCaseRegistry::getAllTestsSorted (C++ function), 51
Catch::ITestInvoker (C++ struct), 51
Catch::ITestInvoker::~ITestInvoker (C++ function), 51
Catch::ITestInvoker::invoke (C++ function), 51
Catch::ITransientExpression (C++ struct), 52
Catch::ITransientExpression::~ITransientExpression (C++ function), 52
Catch::ITransientExpression::getResults (C++ function), 52
Catch::ITransientExpression::isBinaryExpression (C++ function), 52
Catch::ITransientExpression::ITransientExpression (C++ function), 52
```

```

Catch::Matchers::Generic::Detail::finalizer@Matchers::Impl::MatcherUntypedBase::toString
    (C++ function), 172
                                                (C++ function), 103
Catch::Matchers::Generic::PredicateMatch@Matchers::Impl::MatchNotOf (C++
    (C++ class), 102
                                                struct), 57
Catch::Matchers::Generic::PredicateMatch@Matchers::Impl::MatchNotOf::describe
    (C++ function), 102
                                                (C++ function), 57
Catch::Matchers::Generic::PredicateMatch@Matchers::Impl::MatchNotOf::m_underlyingMatch
    (C++ function), 102
                                                (C++ member), 57
Catch::Matchers::Generic::PredicateMatch@Matchers::Impl::MatchNotOf::match
    (C++ function), 102
                                                (C++ function), 57
Catch::Matchers::Impl::MatchAllOf (C++ Catch::Matchers::Impl::MatchNotOf::MatchNotOf
    (C++ struct), 54
                                                struct), 54
Catch::Matchers::Impl::MatchAllOf::describe@Matchers::Matches (C++ function), 172
    (C++ function), 54
                                                Catch::Matchers::Message (C++ function), 173
Catch::Matchers::Impl::MatchAllOf::m_match@Matchers::Predicate (C++ function),
    (C++ member), 54
                                                173
Catch::Matchers::Impl::MatchAllOf::matchCatch@Matchers::StartsWith (C++ function),
    (C++ function), 54
                                                173
Catch::Matchers::Impl::MatchAllOf::operator@Matchers::StdString::CasedString
    (C++ function), 54
                                                (C++ struct), 57
Catch::Matchers::Impl::MatchAnyOf (C++ Catch::Matchers::StdString::CasedString::adjustStr
    (C++ struct), 54
                                                (C++ function), 57
Catch::Matchers::Impl::MatchAnyOf::describe@Matchers::StdString::CasedString::CasedString
    (C++ function), 55
                                                (C++ function), 57
Catch::Matchers::Impl::MatchAnyOf::m_match@Matchers::StdString::CasedString::caseSensit
    (C++ member), 55
                                                (C++ function), 57
Catch::Matchers::Impl::MatchAnyOf::matchCatch@Matchers::StdString::CasedString::m_caseSens
    (C++ function), 55
                                                (C++ member), 57
Catch::Matchers::Impl::MatchAnyOf::operator@Matchers::StdString::CasedString::m_str
    (C++ function), 55
                                                (C++ member), 57
Catch::Matchers::Impl::MatcherBase (C++ Catch::Matchers::StdString::ContainsMatcher
    (C++ struct), 56
                                                (C++ struct), 58
Catch::Matchers::Impl::MatcherBase::operator@Matchers::StdString::ContainsMatcher::Contain
    (C++ function), 56
                                                (C++ function), 58
Catch::Matchers::Impl::MatcherBase::operator@Matchers::StdString::ContainsMatcher::match
    (C++ function), 56
                                                (C++ function), 58
Catch::Matchers::Impl::MatcherBase::operator@Matchers::StdString::EndsWithMatcher
    (C++ function), 56
                                                (C++ struct), 58
Catch::Matchers::Impl::MatcherMethod      Catch::Matchers::StdString::EndsWithMatcher::EndsW
    (C++ struct), 56
                                                (C++ function), 58
Catch::Matchers::Impl::MatcherMethod::match@Matchers::StdString::EndsWithMatcher::match
    (C++ function), 56
                                                (C++ function), 58
Catch::Matchers::Impl::MatcherUntypedBase@Matchers::Matchers::StdString::EqualsMatcher
    (C++ class), 103
                                                (C++ struct), 59
Catch::Matchers::Impl::MatcherUntypedBase@Matchers::Matchers::StdString::EqualsMatcher::Equal
    (C++ function), 103
                                                (C++ function), 59
Catch::Matchers::Impl::MatcherUntypedBase@Matchers::Matchers::StdString::EqualsMatcher::match
    (C++ function), 103
                                                (C++ function), 59
Catch::Matchers::Impl::MatcherUntypedBase@Matchers::Matchers::RegexMatcher
    (C++ member), 103
                                                (C++ struct), 59
Catch::Matchers::Impl::MatcherUntypedBase@Matchers::Matchers::RegexMatcher::describe
    (C++ function), 103
                                                (C++ function), 59
Catch::Matchers::Impl::MatcherUntypedBase@Matchers::Matchers::RegexMatcher::match
    (C++ function), 103
                                                (C++ function), 59

```



Catch::MessageInfo::type (*C++ member*), 65  
 Catch::MessageStream (*C++ struct*), 66  
 Catch::MessageStream::m\_stream (*C++ member*), 66  
 Catch::MessageStream::operator<< (*C++ function*), 66  
 Catch::NameAndTags (*C++ struct*), 66  
 Catch::NameAndTags::name (*C++ member*), 66  
 Catch::NameAndTags::NameAndTags (*C++ function*), 66  
 Catch::NameAndTags::tags (*C++ member*), 66  
 Catch::NonCopyable (*C++ class*), 104  
 Catch::NonCopyable::~NonCopyable (*C++ function*), 104  
 Catch::NonCopyable::NonCopyable (*C++ function*), 104  
 Catch::operator""\_sr (*C++ function*), 177  
 Catch::operator+ (*C++ function*), 177  
 Catch::operator+= (*C++ function*), 177  
 Catch::operator| (*C++ function*), 178  
 Catch::Option (*C++ class*), 104  
 Catch::Option::~Option (*C++ function*), 105  
 Catch::Option::none (*C++ function*), 105  
 Catch::Option::operator bool (*C++ function*), 105  
 Catch::Option::operator! (*C++ function*), 105  
 Catch::Option::operator\* (*C++ function*), 105  
 Catch::Option::operator= (*C++ function*), 105  
 Catch::Option::operator-> (*C++ function*), 105  
 Catch::Option::Option (*C++ function*), 105  
 Catch::Option::reset (*C++ function*), 105  
 Catch::Option::some (*C++ function*), 105  
 Catch::Option::valueOr (*C++ function*), 105  
 Catch::pluralise (*C++ struct*), 67  
 Catch::pluralise::m\_count (*C++ member*), 67  
 Catch::pluralise::m\_label (*C++ member*), 67  
 Catch::pluralise::operator<< (*C++ function*), 67  
 Catch::pluralise::pluralise (*C++ function*), 67  
 Catch::RegistrarForTagAliases (*C++ struct*), 67  
 Catch::RegistrarForTagAliases::RegistrarForTagAliases::InWhatOrder (*C++ enum*), 69  
 Catch::replaceInPlace (*C++ function*), 179  
 Catch::ResultDisposition (*C++ struct*), 68  
 Catch::ResultDisposition::Flags (*C++ enum*), 68  
 Catch::ResultDisposition::Flags::ContinuationFailureTests::InWhatOrder::InDeclarationOrder (*C++ enumerator*), 69  
 Catch::ResultDisposition::Flags::ContinuationFailureTests::InWhatOrder::InLexicographicalOrder (*C++ enumerator*), 69  
 Catch::ResultDisposition::Flags::FalseTests::InWhatOrder::InRandomOrder (*C++ enumerator*), 69  
 Catch::ResultDisposition::Flags::FalseTests::ScopedMessage (*C++ class*), 106  
 Catch::ResultDisposition::Flags::FalseTests::ScopedMessage::~ScopedMessage (*C++ function*), 106

```

Catch::ScopedMessage::m_info (C++ member), 106
Catch::ScopedMessage::m_moved (C++ member), 106
Catch::ScopedMessage::ScopedMessage (C++ function), 106
Catch::Section (C++ class), 107
Catch::Section::~Section (C++ function), 107
Catch::Section::operator bool (C++ function), 107
Catch::Section::Section (C++ function), 107
Catch::SectionEndInfo (C++ struct), 69
Catch::SectionEndInfo::durationInSeconds (C++ member), 69
Catch::SectionEndInfo::prevAssertions (C++ member), 69
Catch::SectionEndInfo::sectionInfo (C++ member), 69
Catch::SectionInfo (C++ struct), 69
Catch::SectionInfo::description (C++ member), 70
Catch::SectionInfo::lineInfo (C++ member), 70
Catch::SectionInfo::name (C++ member), 70
Catch::SectionInfo::SectionInfo (C++ function), 70
Catch::shouldContinueOnFailure (C++ function), 180
Catch::shouldSuppressFailure (C++ function), 180
Catch::ShowDurations (C++ struct), 70
Catch::ShowDurations::OrNot (C++ enum), 70
Catch::ShowDurations::OrNot::Always (C++ enumerator), 70
Catch::ShowDurations::OrNot::DefaultForRepeaterStringMaker<char[SZ]> (C++ struct), 73
Catch::ShowDurations::OrNot::Never (C++ enumerator), 70
Catch::SimplePcg32 (C++ class), 107
Catch::SimplePcg32::discard (C++ function), 107
Catch::SimplePcg32::operator() (C++ function), 107
Catch::SimplePcg32::result_type (C++ type), 107
Catch::SimplePcg32::seed (C++ function), 107
Catch::SimplePcg32::SimplePcg32 (C++ function), 107
Catch::SourceLineInfo (C++ struct), 70
Catch::SourceLineInfo::empty (C++ function), 71
Catch::SourceLineInfo::file (C++ member), 71
Catch::SourceLineInfo::line (C++ member), 71
Catch::SourceLineInfo::operator= (C++ function), 70
Catch::SourceLineInfo::operator== (C++ function), 71
Catch::SourceLineInfo::operator< (C++ function), 71
Catch::SourceLineInfo::SourceLineInfo (C++ function), 70
Catch::splitStringRef (C++ function), 180
Catch::StreamEndStop (C++ struct), 71
Catch::StreamEndStop::operator+ (C++ function), 71
Catch::StringMaker (C++ struct), 71
Catch::StringMaker::convert (C++ function), 71
Catch::StringMaker<bool> (C++ struct), 72
Catch::StringMaker<bool>::convert (C++ function), 72
Catch::StringMaker<Catch::Detail::Approx> (C++ struct), 72
Catch::StringMaker<Catch::Detail::Approx>::convert (C++ function), 72
Catch::StringMaker<char const*> (C++ struct), 73
Catch::StringMaker<char const*>::convert (C++ function), 73
Catch::StringMaker<char*> (C++ struct), 72
Catch::StringMaker<char*>::convert (C++ function), 72
Catch::StringMaker<char> (C++ struct), 73
Catch::StringMaker<char>::convert (C++ function), 73
Catch::StringMaker<double> (C++ struct), 74
Catch::StringMaker<double>::convert (C++ function), 74
Catch::StringMaker<double>::precision (C++ member), 74
Catch::StringMaker<float> (C++ struct), 74
Catch::StringMaker<float>::convert (C++ function), 74
Catch::StringMaker<float>::precision (C++ member), 74
Catch::StringMaker<int> (C++ struct), 75
Catch::StringMaker<int>::convert (C++ function), 75
Catch::StringMaker<long long> (C++ struct), 75
Catch::StringMaker<long long>::convert

```

(C++ function), 75  
 Catch::StringMaker<long> (C++ struct), 75  
 Catch::StringMaker<long>::convert (C++ function), 75  
 Catch::StringMaker<R C::\*> (C++ struct), 76  
 Catch::StringMaker<R C::\*>::convert (C++ function), 76  
 Catch::StringMaker<R, typename std::enable\_if<is\_range<R>>::value && !::Catch::Detail::IsStreamInsertable<R>::value (C++ struct), 76  
 Catch::StringMaker<R, typename std::enable\_if<is\_range<R>>::value && !::Catch::Detail::IsStreamInsertable<R>::value (C++ function), 76  
 Catch::StringMaker<signed char> (C++ struct), 76  
 Catch::StringMaker<signed char>::convert (C++ function), 76  
 Catch::StringMaker<signed char[SZ]> (C++ struct), 77  
 Catch::StringMaker<signed char[SZ]>::convert (C++ function), 77  
 Catch::StringMaker<std::nullptr\_t> (C++ struct), 77  
 Catch::StringMaker<std::nullptr\_t>::convert (C++ function), 77  
 Catch::StringMaker<std::string> (C++ struct), 77  
 Catch::StringMaker<std::string>::convert (C++ function), 77  
 Catch::StringMaker<std::wstring> (C++ struct), 78  
 Catch::StringMaker<std::wstring>::convert (C++ function), 78  
 Catch::StringMaker<T\*> (C++ struct), 78  
 Catch::StringMaker<T\*>::convert (C++ function), 78  
 Catch::StringMaker<T[SZ]> (C++ struct), 78  
 Catch::StringMaker<T[SZ]>::convert (C++ function), 78  
 Catch::StringMaker<unsigned char> (C++ struct), 79  
 Catch::StringMaker<unsigned char>::convert (C++ function), 79  
 Catch::StringMaker<unsigned char[SZ]> (C++ struct), 79  
 Catch::StringMaker<unsigned char[SZ]>::convert (C++ function), 79  
 Catch::StringMaker<unsigned int> (C++ struct), 79  
 Catch::StringMaker<unsigned int>::convert (C++ function), 79  
 Catch::StringMaker<unsigned long long> (C++ struct), 80  
 Catch::StringMaker<unsigned long long>::convert (C++ function), 80  
 Catch::StringMaker<wchar\_t const\*> (C++ struct), 81  
 Catch::StringMaker<wchar\_t const\*>::convert (C++ function), 81  
 Catch::StringRef (C++ class), 108  
 Catch::StringRef::begin (C++ function), 108  
 Catch::StringRef::c\_str (C++ function), 108  
 Catch::StringRef::const\_iterator (C++ type), 108  
 Catch::StringRef::data (C++ function), 108  
 Catch::StringRef::empty (C++ function), 108  
 Catch::StringRef::end (C++ function), 108  
 Catch::StringRef::isNullTerminated (C++ function), 108  
 Catch::StringRef::operator std::string (C++ function), 108  
 Catch::StringRef::operator != (C++ function), 108  
 Catch::StringRef::operator == (C++ function), 108  
 Catch::StringRef::operator [] (C++ function), 108  
 Catch::StringRef::size (C++ function), 108  
 Catch::StringRef::size\_type (C++ type), 108  
 Catch::StringRef::StringRef (C++ function), 108  
 Catch::StringRef::substr (C++ function), 108  
 Catch::TestCase (C++ class), 109  
 Catch::TestCase::getTestCaseInfo (C++ function), 109  
 Catch::TestCase::invoke (C++ function), 109  
 Catch::TestCase::operator == (C++ function), 109  
 Catch::TestCase::operator < (C++ function), 109  
 Catch::TestCase::TestCase (C++ function), 109  
 Catch::TestCase::withName (C++ function), 109  
 Catch::TestCaseInfo (C++ struct), 81  
 Catch::TestCaseInfo::className (C++ mem-

```

ber), 82
Catch::TestCaseInfo::description (C++ member), 82
Catch::TestCaseInfo::expectedToFail (C++ function), 82
Catch::TestCaseInfo::isHidden (C++ function), 82
Catch::TestCaseInfo::lcaseTags (C++ member), 82
Catch::TestCaseInfo::lineInfo (C++ member), 82
Catch::TestCaseInfo::name (C++ member), 82
Catch::TestCaseInfo::okToFail (C++ function), 82
Catch::TestCaseInfo::properties (C++ member), 82
Catch::TestCaseInfo::setTags (C++ function), 82
Catch::TestCaseInfo::SpecialProperties (C++ enum), 81
Catch::TestCaseInfo::SpecialProperties (C++ enumerator), 81
Catch::TestCaseInfo::SpecialProperties::Benchmark (C++ function), 182
Catch::TestCaseInfo::SpecialProperties::CatchBinaryExpr (C++ class), 110
Catch::TestCaseInfo::SpecialProperties::MayFail (C++ enumerator), 81
Catch::TestCaseInfo::SpecialProperties::NonPortable (C++ enumerator), 81
Catch::TestCaseInfo::SpecialProperties::Should (C++ enumerator), 81
Catch::TestCaseInfo::SpecialProperties::Throws (C++ enumerator), 81
Catch::TestCaseInfo::tags (C++ member), 82
Catch::TestCaseInfo::tagsAsString (C++ function), 82
Catch::TestCaseInfo::TestCaseInfo (C++ function), 82
Catch::TestCaseInfo::throws (C++ function), 82
Catch::TestFailureException (C++ struct), 82
Catch::TestInvokerAsMethod (C++ class), 109
Catch::TestInvokerAsMethod::invoke (C++ function), 109
Catch::TestInvokerAsMethod::TestInvokerAsMethod (C++ function), 109
Catch::throw_domain_error (C++ function), 181
Catch::throw_exception (C++ function), 181
Catch::throw_logic_error (C++ function), 181
Catch::throw_runtime_error (C++ function), 182
Catch::Timer (C++ class), 110
Catch::Timer::getElapsedMicroseconds (C++ function), 110
Catch::Timer::getElapsedMilliseconds (C++ function), 110
Catch::Timer::getElapsedNanoseconds (C++ function), 110
Catch::Timer::getElapsedSeconds (C++ function), 110
Catch::Timer::start (C++ function), 110
Catch::toLowerCase (C++ function), 182
Catch::toLowerCaseInPlace (C++ function), 182
Catch::Totals (C++ struct), 83
Catch::Totals::assertions (C++ member), 83
Catch::Totals::delta (C++ function), 83
Catch::Totals::error (C++ member), 83
Catch::Totals::operator+= (C++ function), 83
Catch::Totals::operator- (C++ function), 83
Catch::Totals::testCases (C++ member), 83
Catch::translateActiveException (C++ function), 182
Catch::true_given (C++ struct), 83
Catch::UnaryExpr (C++ class), 110
Catch::UnaryExpr::UnaryExpr (C++ function), 110
Catch::UseColour (C++ struct), 83
Catch::UseColour::YesOrNo (C++ enum), 84
Catch::UseColour::YesOrNo::Auto (C++ enumerator), 84
Catch::UseColour::YesOrNo::No (C++ enum), 84
Catch::UseColour::YesOrNo::Yes (C++ enum), 84
Catch::Verbosity (C++ enum), 148
Catch::Verbosity::High (C++ enumerator), 148
Catch::Verbosity::Normal (C++ enumerator), 148
Catch::Verbosity::Quiet (C++ enumerator), 148
Catch::WaitForKeypress (C++ struct), 84
Catch::WaitForKeypress::When (C++ enum), 84
Catch::WaitForKeypress::When::BeforeExit (C++ enumerator), 84
Catch::WaitForKeypress::When::BeforeStart (C++ enumerator), 84
Catch::WaitForKeypress::When::BeforeStartAndExit (C++ enumerator), 84
Catch::WaitForKeypress::When::Never (C++ enumerator), 84
Catch::WarnAbout (C++ struct), 84
Catch::WarnAbout::What (C++ enum), 84
Catch::WarnAbout::What::NoAssertions (C++ enumerator), 84

```

|   |       |   |
|---|-------|---|
| Catch::WarnAbout::What::NoTests<br>enumerator), 84                          | (C++) | CATCH_REC_LIST1_UD ( <i>C macro</i> ), 236<br>CATCH_REC_LIST2 ( <i>C macro</i> ), 237   |
| Catch::WarnAbout::What::Nothing<br>enumerator), 84                          | (C++) | CATCH_REC_LIST2_UD ( <i>C macro</i> ), 237<br>CATCH_REC_LIST_UD ( <i>C macro</i> ), 237 |
| CATCH_CATCH_ALL ( <i>C macro</i> ), 228                                     |       | CATCH_REC_NEXT ( <i>C macro</i> ), 237  |
| CATCH_CATCH_ANON ( <i>C macro</i> ), 229                                    |       | CATCH_REC_NEXTO ( <i>C macro</i> ), 237   |
| CATCH_CONFIG_COUNTER ( <i>C macro</i> ), 229                                |       | CATCH_REC_NEXT1 ( <i>C macro</i> ), 238   |
| CATCH_CONFIG_CPP11_TO_STRING ( <i>C macro</i> ), 229                        |       | CATCH_REC_OUT ( <i>C macro</i> ), 238   |
| CATCH_CONFIG_DISABLE_EXCEPTIONS<br>macro), 229                              | (C)   | CATCH_RECURSE ( <i>C macro</i> ), 238   |
| CATCH_CONFIG_GLOBAL_NEXTAFTER ( <i>C macro</i> ,<br>229                     |       | CATCH_RECURSION_LEVEL0 ( <i>C macro</i> ), 238  |
| CATCH_CONFIG_POSIX_SIGNALS ( <i>C macro</i> ), 230                          |       | CATCH_RECURSION_LEVEL1 ( <i>C macro</i> ), 239  |
| CATCH_CONFIG_WCHAR ( <i>C macro</i> ), 230                                  |       | CATCH_RECURSION_LEVEL2 ( <i>C macro</i> ), 239  |
| CATCH_DEFER ( <i>C macro</i> ), 230   |       | CATCH_RECURSION_LEVEL3 ( <i>C macro</i> ), 239  |
| CATCH_EMPTY ( <i>C macro</i> ), 231   |       | CATCH_RECURSION_LEVEL4 ( <i>C macro</i> ), 239  |
| CATCH_ENFORCE ( <i>C macro</i> ), 231                                       |       | CATCH_RECURSION_LEVEL5 ( <i>C macro</i> ), 239  |
| CATCH_ERROR ( <i>C macro</i> ), 231   |       | CATCH_REGISTER_ENUM ( <i>C macro</i> ), 240   |
| Catch_global_namespace_dummy ( <i>C++ struct</i> ,<br>85                    |       | CATCH_REGISTER_TAG_ALIAS ( <i>C macro</i> ), 240  |
| CATCH_INTERNAL_CONFIG_COUNTER ( <i>C macro</i> ,<br>231                     |       | CATCH_RUNTIME_ERROR ( <i>C macro</i> ), 240   |
| CATCH_INTERNAL_CONFIG_GLOBAL_NEXTAFTER<br>( <i>C macro</i> ), 231           |       | CATCH_TRANSLATE_EXCEPTION ( <i>C macro</i> ), 240                                       |
| CATCH_INTERNAL_CONFIG_POSIX_SIGNALS ( <i>C<br/>macro</i> ), 232             |       | CATCH_TRY ( <i>C macro</i> ), 241   |
| CATCH_INTERNAL_ERROR ( <i>C macro</i> ), 232                                |       | CATCH_VERSION_MAJOR ( <i>C macro</i> ), 241   |
| CATCH_INTERNAL_IGNORE_BUT_WARN ( <i>C macro</i> ,<br>232                    |       | CATCH_VERSION_MINOR ( <i>C macro</i> ), 241   |
| CATCH_INTERNAL_LINEINFO ( <i>C macro</i> ), 232                             |       | CATCH_VERSION_PATCH ( <i>C macro</i> ), 241   |
| CATCH_INTERNAL_START_WARNINGS_SUPPRESSION<br>( <i>C macro</i> ), 233        |       | CHECK ( <i>C macro</i> ), 241   |
| CATCH_INTERNAL_STOP_WARNINGS_SUPPRESSION<br>( <i>C macro</i> ), 233         |       | CHECK_FALSE ( <i>C macro</i> ), 242   |
| CATCH_INTERNAL_STRINGIFY ( <i>C macro</i> ), 233                            |       | CHECK_NOFAIL ( <i>C macro</i> ), 242  |
| CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGSconfig<br>( <i>C macro</i> ), 233   |       | CHECK_NOTHROW ( <i>C macro</i> ), 242   |
| CATCH_INTERNAL_SUPPRESS_PARENTHESES_WARNINGS<br>( <i>C macro</i> ), 233     |       | CHECK_THAT ( <i>C macro</i> ), 242  |
| CATCH_INTERNAL_SUPPRESS_UNUSED_TEMPLATE_WARNINGS<br>( <i>C macro</i> ), 234 |       | CHECK_THROWS ( <i>C macro</i> ), 243  |
| CATCH_INTERNAL_SUPPRESS_UNUSED_WARNINGS<br>( <i>C macro</i> ), 234          |       | CHECK_THROWS_AS ( <i>C macro</i> ), 243   |
| CATCH_INTERNAL_SUPPRESS_ZERO_VARIADIC_WARNINGS<br>( <i>C macro</i> ), 234   |       | CHECK_THROWS_MATCHES ( <i>C macro</i> ), 243  |
| CATCH_MAKE_MSG ( <i>C macro</i> ), 234                                      |       | CHECK_THROWS_WITH ( <i>C macro</i> ), 243   |
| CATCH_REC_END ( <i>C macro</i> ), 235                                       |       | CheckConfigFile ( <i>C++ function</i> ), 183  |
| CATCH_REC_GET_END ( <i>C macro</i> ), 235                                   |       | CHECKED_ELSE ( <i>C macro</i> ), 243  |
| CATCH_REC_GET_END1 ( <i>C macro</i> ), 235                                  |       | CHECKED_IF ( <i>C macro</i> ), 244  |
| CATCH_REC_GET_END2 ( <i>C macro</i> ), 235                                  |       | CATCH_INTERNAL_SUPPRESS_GLOBALS_WARNINGSconfig<br>( <i>C macro</i> ), 227               |
| CATCH_REC_LIST ( <i>C macro</i> ), 235                                      |       | DYNAMIC_SECTION ( <i>C macro</i> ), 244   |
| CATCH_REC_LIST0 ( <i>C macro</i> ), 236                                     |       | E   |
| CATCH_REC_LIST0_UD ( <i>C macro</i> ), 236                                  |       | EfficientHost ( <i>C++ class</i> ), 111   |
| CATCH_REC_LIST1 ( <i>C macro</i> ), 236                                     |       | EfficientHost::efficiency ( <i>C++ member</i> ),<br>112                                 |
|   |       | EfficientHost::EfficientHost ( <i>C++ func-<br/>tion</i> ), 111                         |
|   |       | EfficientHost::GetEfficiency ( <i>C++ func-<br/>tion</i> ), 111                         |
|   |       | EfficientHost::GetName ( <i>C++ function</i> ), 111                                     |
|   |       | EfficientHost::MakeNew ( <i>C++ function</i> ), 111                                     |
|   |       | EfficientHost::my_world ( <i>C++ member</i> ), 112                                      |
|   |       | EfficientHost::SetEfficiency ( <i>C++ func-<br/>tion</i> ), 111                         |
|   |       | EfficientSymbiont ( <i>C++ class</i> ), 112   |

|  |  |
|--|--|
| EfficientSymbiont::AddPoints ( <i>C++ function</i> ), 113              | Host::AddPoints ( <i>C++ function</i> ), 118           |
| EfficientSymbiont::eff_mut_rate ( <i>C++ member</i> ), 114             | Host::AddReproSym ( <i>C++ function</i> ), 119         |
| EfficientSymbiont::efficiency ( <i>C++ member</i> ), 114               | Host::AddSymbiont ( <i>C++ function</i> ), 119         |
| EfficientSymbiont::EfficientSymbiont ( <i>C++ function</i> ), 112      | Host::age ( <i>C++ member</i> ), 120                   |
| EfficientSymbiont::GetEfficiency ( <i>C++ function</i> ), 113          | Host::ClearReproSyms ( <i>C++ function</i> ), 118      |
| EfficientSymbiont::GetName ( <i>C++ function</i> ), 113                | Host::ClearSyms ( <i>C++ function</i> ), 118           |
| EfficientSymbiont::HorizontalTransmission ( <i>C++ function</i> ), 113 | Host::dead ( <i>C++ member</i> ), 121                  |
| EfficientSymbiont::ht_mut_rate ( <i>C++ member</i> ), 114              | Host::DistribResources ( <i>C++ function</i> ), 119    |
| EfficientSymbiont::ht_mut_size ( <i>C++ member</i> ), 114              | Host::DistribResToSym ( <i>C++ function</i> ), 120     |
| EfficientSymbiont::MakeNew ( <i>C++ function</i> ), 113                | Host::GetAge ( <i>C++ function</i> ), 118              |
| EfficientSymbiont::Mutate ( <i>C++ function</i> ), 113                 | Host::GetDead ( <i>C++ function</i> ), 118             |
| EfficientSymbiont::my_world ( <i>C++ member</i> ), 114                 | Host::GetDoEctosymbiosis ( <i>C++ function</i> ), 120  |
| EfficientSymbiont::Reproduce ( <i>C++ function</i> ), 113              | Host::GetIntVal ( <i>C++ function</i> ), 117           |
| EfficientSymbiont::SetEfficiency ( <i>C++ function</i> ), 113          | Host::GetName ( <i>C++ function</i> ), 117             |
| EfficientSymbiont::VerticalTransmission ( <i>C++ function</i> ), 113   | Host::GetPoints ( <i>C++ function</i> ), 117           |
| EfficientWorld ( <i>C++ class</i> ), 114                               | Host::GetReproSymbionts ( <i>C++ function</i> ), 117   |
| EfficientWorld::~EfficientWorld ( <i>C++ function</i> ), 114           | Host::GetResInProcess ( <i>C++ function</i> ), 117     |
| EfficientWorld::CreateDataFiles ( <i>C++ function</i> ), 115           | Host::GetSymbionts ( <i>C++ function</i> ), 117        |
| EfficientWorld::GetEfficiencyDataNode ( <i>C++ function</i> ), 115     | Host::GrowOlder ( <i>C++ function</i> ), 118           |
| EfficientWorld::Setup ( <i>C++ function</i> ), 114                     | Host::HandleEctosymbiosis ( <i>C++ function</i> ), 120 |
| EfficientWorld::SetupEfficiencyFile ( <i>C++ function</i> ), 115       | Host::HasSym ( <i>C++ function</i> ), 119              |

## F

FAIL (*C macro*), 244  
FAIL\_CHECK (*C macro*), 245

## G

GENERATE (*C macro*), 245  
GENERATE\_COPY (*C macro*), 245  
GENERATE\_REF (*C macro*), 245  
GIVEN (*C macro*), 245

## H

Host (*C++ class*), 116  
Host::~Host (*C++ function*), 116

|  |  |
|--|--|
| Host::AddPoints ( <i>C++ function</i> ), 118           | Host::AddReproSym ( <i>C++ function</i> ), 119         |
| Host::AddSymbiont ( <i>C++ function</i> ), 119         | Host::age ( <i>C++ member</i> ), 120                   |
| Host::ClearReproSyms ( <i>C++ function</i> ), 118      | Host::ClearSyms ( <i>C++ function</i> ), 118           |
| Host::dead ( <i>C++ member</i> ), 121                  | Host::DistribResources ( <i>C++ function</i> ), 119    |
| Host::DistribResToSym ( <i>C++ function</i> ), 120     | Host::GetAge ( <i>C++ function</i> ), 118              |
| Host::GetDead ( <i>C++ function</i> ), 118             | Host::GetDoEctosymbiosis ( <i>C++ function</i> ), 120  |
| Host::GetDoEctosymbiosis ( <i>C++ function</i> ), 120  | Host::GetIntVal ( <i>C++ function</i> ), 117           |
| Host::GetName ( <i>C++ function</i> ), 117             | Host::GetPoints ( <i>C++ function</i> ), 117           |
| Host::GetReproSymbionts ( <i>C++ function</i> ), 117   | Host::GetResInProcess ( <i>C++ function</i> ), 117     |
| Host::GetSymbionts ( <i>C++ function</i> ), 117        | Host::GetSymbionts ( <i>C++ function</i> ), 117        |
| Host::GrowOlder ( <i>C++ function</i> ), 118           | Host::HandleEctosymbiosis ( <i>C++ function</i> ), 120 |
| Host::HandleEctosymbiosis ( <i>C++ function</i> ), 120 | Host::HasSym ( <i>C++ function</i> ), 119              |
| Host::HasSym ( <i>C++ function</i> ), 119              | Host::Host ( <i>C++ function</i> ), 116                |
| Host::Host ( <i>C++ function</i> ), 116                | Host::interaction_val ( <i>C++ member</i> ), 120       |
| Host::interaction_val ( <i>C++ member</i> ), 120       | Host::IsHost ( <i>C++ function</i> ), 117              |
| Host::IsHost ( <i>C++ function</i> ), 117              | Host::MakeNew ( <i>C++ function</i> ), 119             |
| Host::MakeNew ( <i>C++ function</i> ), 119             | Host::Mutate ( <i>C++ function</i> ), 119              |
| Host::Mutate ( <i>C++ function</i> ), 119              | Host::my_config ( <i>C++ member</i> ), 120             |
| Host::my_config ( <i>C++ member</i> ), 120             | Host::my_world ( <i>C++ member</i> ), 120              |
| Host::my_world ( <i>C++ member</i> ), 120              | Host::operator!= ( <i>C++ function</i> ), 116          |
| Host::operator!= ( <i>C++ function</i> ), 116          | Host::operator= ( <i>C++ function</i> ), 116           |
| Host::operator= ( <i>C++ function</i> ), 116           | Host::operator== ( <i>C++ function</i> ), 116          |
| Host::operator== ( <i>C++ function</i> ), 116          | Host::points ( <i>C++ member</i> ), 120                |
| Host::points ( <i>C++ member</i> ), 120                | Host::Process ( <i>C++ function</i> ), 120             |
| Host::Process ( <i>C++ function</i> ), 120             | Host::random ( <i>C++ member</i> ), 120                |
| Host::random ( <i>C++ member</i> ), 120                | Host::repro_syms ( <i>C++ member</i> ), 120            |
| Host::repro_syms ( <i>C++ member</i> ), 120            | Host::Reproduce ( <i>C++ function</i> ), 119           |
| Host::Reproduce ( <i>C++ function</i> ), 119           | Host::res_in_process ( <i>C++ member</i> ), 120        |
| Host::res_in_process ( <i>C++ member</i> ), 120        | Host::SetAge ( <i>C++ function</i> ), 118              |
| Host::SetAge ( <i>C++ function</i> ), 118              | Host::SetDead ( <i>C++ function</i> ), 118             |
| Host::SetDead ( <i>C++ function</i> ), 118             | Host::SetIntVal ( <i>C++ function</i> ), 117           |
| Host::SetIntVal ( <i>C++ function</i> ), 117           | Host::SetPoints ( <i>C++ function</i> ), 117           |
| Host::SetPoints ( <i>C++ function</i> ), 117           | Host::SetResInProcess ( <i>C++ function</i> ), 118     |
| Host::SetResInProcess ( <i>C++ function</i> ), 118     | Host::SetSymbionts ( <i>C++ function</i> ), 117        |
| Host::SetSymbionts ( <i>C++ function</i> ), 117        | Host::StealResources ( <i>C++ function</i> ), 118      |
| Host::StealResources ( <i>C++ function</i> ), 118      | Host::SymAllowedIn ( <i>C++ function</i> ), 119        |
| Host::SymAllowedIn ( <i>C++ function</i> ), 119        | Host::syms ( <i>C++ member</i> ), 120                  |

## I

|  |   |
|--|---|
| INFO ( <i>C macro</i> ), 246                 | INTERNAL_CATCH_CAPTURE ( <i>C macro</i> ), 246          |
| INTERNAL_CATCH_CATCH ( <i>C macro</i> ), 246 | INTERNAL_CATCH_DECLARE_SIG_TEST ( <i>C macro</i> ), 246 |

|   |    |   |
|---|----|---|
| INTERNAL_CATCH_DECLARE_SIG_TEST0<br><i>(C macro)</i> , 247          | (C | INTERNAL_CATCH_NTTP_0 ( <i>C macro</i> ), 253<br>INTERNAL_CATCH_NTTP_1 ( <i>C macro</i> ), 254                |
| INTERNAL_CATCH_DECLARE_SIG_TEST1<br><i>(C macro)</i> , 247          | (C | INTERNAL_CATCH_NTTP_GEN ( <i>C macro</i> ), 254<br>INTERNAL_CATCH_NTTP_REG_GEN ( <i>C macro</i> ), 254        |
| INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD<br><i>(C macro)</i> , 247    |    | INTERNAL_CATCH_NTTP_REG_METHOD_GEN ( <i>C macro</i> ), 254  |
| INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD0<br><i>(C macro)</i> , 247   |    | INTERNAL_CATCH_NTTP_REGISTER ( <i>C macro</i> ), 255<br>INTERNAL_CATCH_NTTP_REGISTER0 ( <i>C macro</i> ), 255 |
| INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD1<br><i>(C macro)</i> , 247   |    | INTERNAL_CATCH_NTTP_REGISTER_METHOD ( <i>C macro</i> ), 255   |
| INTERNAL_CATCH_DECLARE_SIG_TEST_METHOD_X<br><i>(C macro)</i> , 248  |    | INTERNAL_CATCH_NTTP_REGISTER_METHOD0 ( <i>C macro</i> ), 255  |
| INTERNAL_CATCH_DECLARE_SIG_TEST_X<br><i>(C macro)</i> , 248         | (C | INTERNAL_CATCH.REACT ( <i>C macro</i> ), 255  |
| INTERNAL_CATCH_DEF ( <i>C macro</i> ), 248                          |    | INTERNAL_CATCH_REGISTER_ENUM ( <i>C macro</i> ), 256  |
| INTERNAL_CATCH_DEFINE_SIG_TEST ( <i>C macro</i> ), 248              |    | INTERNAL_CATCH_REGISTER_TESTCASE ( <i>C macro</i> ), 256  |
| INTERNAL_CATCH_DEFINE_SIG_TEST0<br><i>(C macro)</i> , 249           | (C | INTERNAL_CATCH_REMOVE_PARENS ( <i>C macro</i> ), 256  |
| INTERNAL_CATCH_DEFINE_SIG_TEST1<br><i>(C macro)</i> , 249           | (C | INTERNAL_CATCH_REMOVE_PARENS_10_ARG ( <i>C macro</i> ), 256   |
| INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD<br><i>(C macro)</i> , 249     |    | INTERNAL_CATCH_REMOVE_PARENS_11_ARG ( <i>C macro</i> ), 257   |
| INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD0<br><i>(C macro)</i> , 249    |    | INTERNAL_CATCH_REMOVE_PARENS_1_ARG ( <i>C macro</i> ), 257  |
| INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD1<br><i>(C macro)</i> , 249    |    | INTERNAL_CATCH_REMOVE_PARENS_2_ARG ( <i>C macro</i> ), 257  |
| INTERNAL_CATCH_DEFINE_SIG_TEST_METHOD_X<br><i>(C macro)</i> , 250   |    | INTERNAL_CATCH_REMOVE_PARENS_3_ARG ( <i>C macro</i> ), 257  |
| INTERNAL_CATCH_DEFINE_SIG_TEST_X<br><i>(C macro)</i> , 250          | (C | INTERNAL_CATCH_REMOVE_PARENS_4_ARG ( <i>C macro</i> ), 257  |
| INTERNAL_CATCH_DYNAMIC_SECTION ( <i>C macro</i> ), 250              |    | INTERNAL_CATCH_REMOVE_PARENS_5_ARG ( <i>C macro</i> ), 258  |
| INTERNAL_CATCH_ELSE ( <i>C macro</i> ), 250                         |    | INTERNAL_CATCH_REMOVE_PARENS_6_ARG ( <i>C macro</i> ), 258  |
| INTERNAL_CATCH_EXPAND1 ( <i>C macro</i> ), 251                      |    | INTERNAL_CATCH_REMOVE_PARENS_7_ARG ( <i>C macro</i> ), 258  |
| INTERNAL_CATCH_EXPAND2 ( <i>C macro</i> ), 251                      |    | INTERNAL_CATCH_REMOVE_PARENS_8_ARG ( <i>C macro</i> ), 258  |
| INTERNAL_CATCH_IF ( <i>C macro</i> ), 251                           |    | INTERNAL_CATCH_REMOVE_PARENS_9_ARG ( <i>C macro</i> ), 259  |
| INTERNAL_CATCH_INFO ( <i>C macro</i> ), 251                         |    | INTERNAL_CATCH_REMOVE_PARENS_GEN ( <i>C macro</i> ), 259  |
| INTERNAL_CATCH_MAKE_NAMESPACE ( <i>C macro</i> ), 251               |    | INTERNAL_CATCH_SECTION ( <i>C macro</i> ), 259  |
| INTERNAL_CATCH_MAKE_NAMESPACE2 ( <i>C macro</i> ), 252              |    | INTERNAL_CATCH_STRINGIZE ( <i>C macro</i> ), 259  |
| INTERNAL_CATCH_MAKE_TYPE_LIST ( <i>C macro</i> ), 252               |    | INTERNAL_CATCH_STRINGIZE2 ( <i>C macro</i> ), 259   |
| INTERNAL_CATCH_MAKE_TYPE_LIST2 ( <i>C macro</i> ), 252              |    | INTERNAL_CATCH_STRINGIZE_WITHOUT_PARENS ( <i>C macro</i> ), 260   |
| INTERNAL_CATCH_MAKE_TYPE_LISTS_FROM_TYPES<br><i>(C macro)</i> , 252 |    | INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE ( <i>C macro</i> ), 260  |
| INTERNAL_CATCH_METHOD_AS_TEST_CASE<br><i>(C macro)</i> , 253        | (C | INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_2 ( <i>C macro</i> ), 260  |
| INTERNAL_CATCH_MSG ( <i>C macro</i> ), 253                          |    | INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD ( <i>C macro</i> ), 260   |
| INTERNAL_CATCH_NO_THROW ( <i>C macro</i> ), 253                     |    | INTERNAL_CATCH_TEMPLATE_LIST_TEST_CASE_METHOD_2   |
| INTERNAL_CATCH_NOINTERNAL_CATCH_DEF<br><i>(C macro)</i> , 253       |    |   |



Organism::GetIntVal (*C++ function*), 123  
 Organism::GetLysisChance (*C++ function*), 125  
 Organism::GetLysogeny (*C++ function*), 125  
 Organism::GetName (*C++ function*), 123  
 Organism::GetPoints (*C++ function*), 123  
 Organism::GetReproSymbionts (*C++ function*),  
     124  
 Organism::GetResInProcess (*C++ function*),  
     124  
 Organism::GetSymbionts (*C++ function*), 124  
 Organism::GetTaxon (*C++ function*), 124  
 Organism::HasSym (*C++ function*), 124  
 Organism::HorizontalTransmission (*C++  
     function*), 124  
 Organism::IncBurstTimer (*C++ function*), 125  
 Organism::InfectionFails (*C++ function*), 124  
 Organism::IsHost (*C++ function*), 124  
 Organism::IsPhage (*C++ function*), 124  
 Organism::LysisBurst (*C++ function*), 125  
 Organism::LysisStep (*C++ function*), 125  
 Organism::MakeNew (*C++ function*), 124  
 Organism::Mutate (*C++ function*), 124  
 Organism::operator!= (*C++ function*), 123  
 Organism::operator= (*C++ function*), 123  
 Organism::operator== (*C++ function*), 123  
 Organism::Organism (*C++ function*), 123  
 Organism::Process (*C++ function*), 124  
 Organism::ProcessLysogenResources (*C++  
     function*), 125  
 Organism::ProcessPool (*C++ function*), 125  
 Organism::ProcessResources (*C++ function*),  
     124  
 Organism::Reproduce (*C++ function*), 124  
 Organism::SetAge (*C++ function*), 124  
 Organism::SetBurstTimer (*C++ function*), 125  
 Organism::SetDead (*C++ function*), 123  
 Organism::SetDonation (*C++ function*), 125  
 Organism::SetEfficiency (*C++ function*), 124  
 Organism::SetHost (*C++ function*), 123  
 Organism::SetIncVal (*C++ function*), 124  
 Organism::SetInductionChance (*C++ func-  
     tion*), 125  
 Organism::SetInfectionChance (*C++ func-  
     tion*), 124  
 Organism::SetIntVal (*C++ function*), 123  
 Organism::SetLysisChance (*C++ function*), 125  
 Organism::SetPoints (*C++ function*), 123  
 Organism::SetPool (*C++ function*), 125  
 Organism::SetResInProcess (*C++ function*),  
     124  
 Organism::SetSymbionts (*C++ function*), 124  
 Organism::SetTaxon (*C++ function*), 124  
 Organism::StealResources (*C++ function*), 124  
 Organism::UponInjection (*C++ function*), 125  
 Organism::VerticalTransmission (*C++ func-  
     tion*), 124  
 Organism::WantsToInfect (*C++ function*), 124

**P**

PGGHost (*C++ class*), 125  
 PGGHost::AddPool (*C++ function*), 126  
 PGGHost::DistribPool (*C++ function*), 126  
 PGGHost::DistribResources (*C++ function*),  
     126  
 PGGHost::GetName (*C++ function*), 126  
 PGGHost::GetPool (*C++ function*), 126  
 PGGHost::MakeNew (*C++ function*), 127  
 PGGHost::my\_world (*C++ member*), 127  
 PGGHost::PGGHost (*C++ function*), 126  
 PGGHost::SetPool (*C++ function*), 126  
 PGGHost::sourcepool (*C++ member*), 127  
 PGGSymbiont (*C++ class*), 127  
 PGGSymbiont::GetDonation (*C++ function*), 128  
 PGGSymbiont::GetName (*C++ function*), 128  
 PGGSymbiont::MakeNew (*C++ function*), 128  
 PGGSymbiont::Mutate (*C++ function*), 128  
 PGGSymbiont::my\_world (*C++ member*), 129  
 PGGSymbiont::operator= (*C++ function*), 128  
 PGGSymbiont::PGG\_donate (*C++ member*), 129  
 PGGSymbiont::PGGSymbiont (*C++ function*), 127  
 PGGSymbiont::PrintSym (*C++ function*), 128  
 PGGSymbiont::ProcessPool (*C++ function*), 128  
 PGGSymbiont::SetDonation (*C++ function*), 128  
 PGGWorld (*C++ class*), 129  
 PGGWorld::~PGGWorld (*C++ function*), 129  
 PGGWorld::CreateDataFiles (*C++ function*),  
     129  
 PGGWorld::GetPGGDataNode (*C++ function*), 130  
 PGGWorld::SetupPGGSymIntValFile (*C++  
     function*), 130  
 Phage (*C++ class*), 130  
 Phage::burst\_timer (*C++ member*), 133  
 Phage::chance\_of\_lysis (*C++ member*), 133  
 Phage::GetBurstTimer (*C++ function*), 131  
 Phage::GetIncVal (*C++ function*), 131  
 Phage::GetInductionChance (*C++ function*),  
     131  
 Phage::GetLysisChance (*C++ function*), 131  
 Phage::GetLysogeny (*C++ function*), 132  
 Phage::GetName (*C++ function*), 131  
 Phage::IncBurstTimer (*C++ function*), 131  
 Phage::incorporation\_val (*C++ member*), 133  
 Phage::induction\_chance (*C++ member*), 133  
 Phage::IsPhage (*C++ function*), 132  
 Phage::LysisBurst (*C++ function*), 132  
 Phage::LysisStep (*C++ function*), 132  
 Phage::lysogeny (*C++ member*), 133  
 Phage::MakeNew (*C++ function*), 132

Phage::Mutate (*C++ function*), 132  
Phage::my\_world (*C++ member*), 133  
Phage::Phage (*C++ function*), 130, 131  
Phage::Process (*C++ function*), 133  
Phage::ProcessResources (*C++ function*), 133  
Phage::SetBurstTimer (*C++ function*), 131  
Phage::SetIncVal (*C++ function*), 131  
Phage::SetInductionChance (*C++ function*), 132  
Phage::SetLysisChance (*C++ function*), 131  
Phage::UponInjection (*C++ function*), 132  
Phage::VerticalTransmission (*C++ function*), 132

## R

REGISTER\_TEST\_CASE (*C macro*), 268  
REQUIRE (*C macro*), 269  
REQUIRE\_FALSE (*C macro*), 269  
REQUIRE\_NO\_THROW (*C macro*), 269  
REQUIRE\_THAT (*C macro*), 269  
REQUIRE\_THROWS (*C macro*), 270  
REQUIRE\_THROWS\_AS (*C macro*), 270  
REQUIRE\_THROWS\_MATCHES (*C macro*), 270  
REQUIRE\_THROWS\_WITH (*C macro*), 270

## S

SCENARIO (*C macro*), 270  
SCENARIO\_METHOD (*C macro*), 271  
SECTION (*C macro*), 271  
STATIC\_REQUIRE (*C macro*), 271  
STATIC\_REQUIRE\_FALSE (*C macro*), 271  
SUCCEED (*C macro*), 272  
SymAnimate (*C++ class*), 134  
SymAnimate::DoFrame (*C++ function*), 134  
SymAnimate::drawPetriDish (*C++ function*), 134  
SymAnimate::initializeWorld (*C++ function*), 134  
SymAnimate::matchColor (*C++ function*), 134  
SymAnimate::setButtonStyle (*C++ function*), 134  
SymAnimate::SymAnimate (*C++ function*), 134  
Symbiont (*C++ class*), 135  
Symbiont::~Symbiont (*C++ function*), 135  
Symbiont::AddPoints (*C++ function*), 137  
Symbiont::age (*C++ member*), 139  
Symbiont::dead (*C++ member*), 139  
Symbiont::GetAge (*C++ function*), 137  
Symbiont::GetDead (*C++ function*), 137  
Symbiont::GetHost (*C++ function*), 136  
Symbiont::GetInfectionChance (*C++ function*), 136  
Symbiont::GetIntVal (*C++ function*), 136  
Symbiont::GetName (*C++ function*), 135

Symbiont::GetPoints (*C++ function*), 136  
Symbiont::GetTaxon (*C++ function*), 136  
Symbiont::GrowOlder (*C++ function*), 137  
Symbiont::HorizontalTransmission (*C++ function*), 139  
Symbiont::infection\_chance (*C++ member*), 139  
Symbiont::InfectionFails (*C++ function*), 138  
Symbiont::interaction\_val (*C++ member*), 139  
Symbiont::IsHost (*C++ function*), 136  
Symbiont::IsPhage (*C++ function*), 136  
Symbiont::LoseResources (*C++ function*), 138  
Symbiont::MakeNew (*C++ function*), 138  
Symbiont::Mutate (*C++ function*), 138  
Symbiont::my\_config (*C++ member*), 139  
Symbiont::my\_host (*C++ member*), 139  
Symbiont::my\_taxon (*C++ member*), 139  
Symbiont::my\_world (*C++ member*), 139  
Symbiont::operator= (*C++ function*), 135  
Symbiont::points (*C++ member*), 139  
Symbiont::Process (*C++ function*), 138  
Symbiont::ProcessResources (*C++ function*), 138  
Symbiont::random (*C++ member*), 139  
Symbiont::Reproduce (*C++ function*), 138  
Symbiont::SetAge (*C++ function*), 137  
Symbiont::SetDead (*C++ function*), 136  
Symbiont::SetHost (*C++ function*), 137  
Symbiont::SetInfectionChance (*C++ function*), 137  
Symbiont::SetIntVal (*C++ function*), 137  
Symbiont::SetPoints (*C++ function*), 137  
Symbiont::SetTaxon (*C++ function*), 136  
Symbiont::Symbiont (*C++ function*), 135  
Symbiont::UponInjection (*C++ function*), 137  
Symbiont::VerticalTransmission (*C++ function*), 138  
Symbiont::WantsToInfect (*C++ function*), 138  
SymWorld (*C++ class*), 140  
SymWorld::~SymWorld (*C++ function*), 140  
SymWorld::AddOrgAt (*C++ function*), 141  
SymWorld::AddSymToSystematic (*C++ function*), 141  
SymWorld::calc\_info\_fun (*C++ member*), 147  
SymWorld::CreateDataFiles (*C++ function*), 142  
SymWorld::data\_node\_attempts\_horiztrans (*C++ member*), 147  
SymWorld::data\_node\_attempts\_vertrans (*C++ member*), 147  
SymWorld::data\_node\_freesymcount (*C++ member*), 147

SymWorld::data\_node\_freesyminfectchance  
     (*C++ member*), 147  
 SymWorld::data\_node\_freesymintval (*C++ member*), 147  
 SymWorld::data\_node\_hostcount (*C++ member*), 147  
 SymWorld::data\_node\_hostedsymcount (*C++ member*), 147  
 SymWorld::data\_node\_hostedsyminfectchance  
     (*C++ member*), 147  
 SymWorld::data\_node\_hostedsymintval  
     (*C++ member*), 147  
 SymWorld::data\_node\_hostintval (*C++ member*), 147  
 SymWorld::data\_node\_successes\_horiztrans  
     (*C++ member*), 147  
 SymWorld::data\_node\_symcount (*C++ member*), 147  
 SymWorld::data\_node\_syminfectchance  
     (*C++ member*), 147  
 SymWorld::data\_node\_symintval (*C++ member*), 147  
 SymWorld::data\_node\_uninf\_hosts  
     (*C++ member*), 147  
 SymWorld::DoBirth (*C++ function*), 141  
 SymWorld::DoSymDeath (*C++ function*), 146  
 SymWorld::ExtractSym (*C++ function*), 146  
 SymWorld::fun\_calc\_info\_t (*C++ type*), 146  
 SymWorld::GetCalcInfoFun (*C++ function*), 141  
 SymWorld::GetCountFreeSymsDataNode (*C++ function*), 143  
 SymWorld::GetCountHostedSymsDataNode  
     (*C++ function*), 143  
 SymWorld::GetDominantFreeHostedSymTaxon  
     (*C++ function*), 142  
 SymWorld::GetDominantHostTaxon (*C++ function*), 142  
 SymWorld::GetDominantSymTaxon (*C++ function*), 142  
 SymWorld::GetFreeSymInfectChanceDataNode  
     (*C++ function*), 144  
 SymWorld::GetFreeSymIntValDataNode (*C++ function*), 144  
 SymWorld::GetHorizontalTransmissionAttempt  
     (*C++ function*), 143  
 SymWorld::GetHorizontalTransmissionSuccess  
     (*C++ function*), 144  
 SymWorld::GetHostCountDataNode (*C++ function*), 143  
 SymWorld::GetHostedSymInfectChanceDataNode  
     (*C++ function*), 145  
 SymWorld::GetHostedSymIntValDataNode  
     (*C++ function*), 144  
 SymWorld::GetHostIntValDataNode  
     (*C++ function*), 144  
 SymWorld::GetHostSys  
     (*C++ function*), 144  
 SymWorld::GetHostSys  
     (*C++ function*), 140  
 SymWorld::GetNeighborHost  
     (*C++ function*), 142  
 SymWorld::GetPop  
     (*C++ function*), 140  
 SymWorld::GetSymAt  
     (*C++ function*), 146  
 SymWorld::GetSymCountDataNode  
     (*C++ function*), 143  
 SymWorld::GetSymInfectChanceDataNode  
     (*C++ function*), 144  
 SymWorld::GetSymIntValDataNode  
     (*C++ function*), 144  
 SymWorld::GetSymPop  
     (*C++ function*), 140  
 SymWorld::GetSymSys  
     (*C++ function*), 141  
 SymWorld::GetUninfectedHostsDataNode  
     (*C++ function*), 143  
 SymWorld::GetVerticalTransmissionAttemptCount  
     (*C++ function*), 144  
 SymWorld::host\_sys  
     (*C++ member*), 147  
 SymWorld::InjectHost  
     (*C++ function*), 142  
 SymWorld::InjectSymbiont  
     (*C++ function*), 142  
 SymWorld::IsInboundsPos  
     (*C++ function*), 145  
 SymWorld::MoveFreeSym  
     (*C++ function*), 146  
 SymWorld::MoveIntoNewFreeWorldPos  
     (*C++ function*), 145  
 SymWorld::my\_config  
     (*C++ member*), 147  
 SymWorld::PullResources  
     (*C++ function*), 141  
 SymWorld::Resize  
     (*C++ function*), 141  
 SymWorld::RunExperiment  
     (*C++ function*), 146  
 SymWorld::SetMutationZero  
     (*C++ function*), 146  
 SymWorld::Setup  
     (*C++ function*), 145  
 SymWorld::SetUpFreeLivingSymFile  
     (*C++ function*), 142  
 SymWorld::SetupHostFileColumns  
     (*C++ function*), 143  
 SymWorld::SetupHostIntValFile  
     (*C++ function*), 142  
 SymWorld::SetupSymIntValFile  
     (*C++ function*), 142  
 SymWorld::SetUpTransmissionFile  
     (*C++ function*), 143  
 SymWorld::sym\_pop  
     (*C++ member*), 147  
 SymWorld::sym\_sys  
     (*C++ member*), 147  
 SymWorld::SymDoBirth  
     (*C++ function*), 145  
 SymWorld::SymWorld  
     (*C++ function*), 140  
 SymWorld::total\_res  
     (*C++ member*), 147  
 SymWorld::Update  
     (*C++ function*), 146  
 SymWorld::WillTransmit  
     (*C++ function*), 140  
 SymWorld::WriteDominantPhylogenyFiles  
     (*C++ function*), 142  
 SymWorld::WritePhylogenyFile  
     (*C++ function*), 142

## T

TEMPLATE\_LIST\_TEST\_CASE (*C macro*), 272  
TEMPLATE\_LIST\_TEST\_CASE\_METHOD (*C macro*),  
272  
TEMPLATE\_PRODUCT\_TEST\_CASE (*C macro*), 272  
TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD (*C  
macro*), 272  
TEMPLATE\_PRODUCT\_TEST\_CASE\_METHOD\_SIG  
(*C macro*), 273  
TEMPLATE\_PRODUCT\_TEST\_CASE\_SIG (*C macro*),  
273  
TEMPLATE\_TEST\_CASE (*C macro*), 273  
TEMPLATE\_TEST\_CASE\_METHOD (*C macro*), 273  
TEMPLATE\_TEST\_CASE\_METHOD\_SIG (*C macro*),  
274  
TEMPLATE\_TEST\_CASE\_SIG (*C macro*), 274  
TEST\_CASE (*C macro*), 274  
TEST\_CASE\_METHOD (*C macro*), 274  
THEN (*C macro*), 274

## U

UNSCOPED\_INFO (*C macro*), 275

## W

WARN (*C macro*), 275  
WHEN (*C macro*), 275