

Welcome to the Sim-Diasca 2.4.3  
Documentation Overview Page  
Public Version

# Sim~Diasca

Simulation of Discrete Systems of All Scales



We recommend to have a look at the [Sim-Diasca General-Purpose Presentation](#) first, in order to discover the engine.

Afterwards, one may refer to the installation guide below.

Alternatively, much shorter versions thereof are:

- either to follow these quickstart instructions:

```
# Ensure that a recent Erlang version is installed, then:
$ git clone https://github.com/Olivier-Boudeville-EDF/Sim-Diasca.git
$ cd Sim-Diasca && make test
```

- or to have a look at the streamlined [executable procedure](#) on which we rely for continuous integration

The following public documentation for **Sim-Diasca** version 2.4.3 is available from here:

The <b>Sim-Diasca Technical Manual</b>	The richest detailed source of information regarding the engine (except the code itself of course).	<a href="#">[html]</a>	<a href="#">[pdf]</a>
The <b>Sim-Diasca Installation Guide</b>	A much detailed installation walkthrough, if needed.	<a href="#">[html]</a>	<a href="#">[pdf]</a>
The <b>Sim-Diasca Modeller Guide</b>	Hints to better design simulation models.	<a href="#">[html]</a>	<a href="#">[pdf]</a>
The <b>Sim-Diasca Developer Guide</b>	General recommendations when developing services offered by the engine (also applicable for model implementation).	<a href="#">[html]</a>	<a href="#">[pdf]</a>
The <b>Sim-Diasca Dataflow HOWTO</b>	For the very specific use cases where a dataflow paradigm would be more suitable than a multi-agent one.	<a href="#">[html]</a>	<a href="#">[pdf]</a>
The <b>Sim-Diasca Coupling HOWTO</b>	When having to couple third-party models, instead of writing them directly as Sim-Diasca ones.	<a href="#">[html]</a>	<a href="#">[pdf]</a>

Seasoned Sim-Diasca users may just bookmark the [Sim-Diasca Cheat Sheet](#).

As an example, this [IFL2021 article](#) discusses an application case (the ACME simulator) and, starting from its section 5, offers a walk-through of the full corresponding software stack. See also this [overview of the metaprogramming](#) taking place with Sim-Diasca.

One may also be interested in the interface of the [public main project](#) and in its [wiki](#).

A mirrored documentation of the lower layers used by Sim-Diasca is available

from here as well:

The <b>Traces</b> <b>Manual</b>	To benefit from convenient, distributed traces in order to collect all information of interest from the simulation.	<a href="#">[html]</a> <a href="#">[pdf]</a>	
The <b>WOOPER</b> <b>Manual</b>	So that the model implementation can rely on a suitable object-oriented paradigm.	<a href="#">[html]</a> <a href="#">[pdf]</a>	
The <b>Myriad</b> <b>Manual</b>	To take advantage of a general-purpose toolbox providing a range of built-in transverse services.	<a href="#">[html]</a> <a href="#">[pdf]</a>	

Seasoned WOOPER users may just bookmark the [WOOPER Cheat Sheet](#).

Finally, regarding the overall implementation language, [Erlang](#), one may refer to [its searchable online documentation](#), including its [module index](#).

*This public overview page for Sim-Diasca version 2.4.3 has been last updated on Tuesday, March 22, 2022.*