

---

# NEST Desktop

Aug 20, 2021



---

## Contents

---

<b>1</b>	<b>NEST Desktop</b>	<b>1</b>
<b>2</b>	<b>Abstract</b>	<b>3</b>
<b>3</b>	<b>Past events</b>	<b>5</b>
<b>4</b>	<b>How to use NEST Desktop</b>	<b>7</b>
<b>5</b>	<b>Usage Guide</b>	<b>9</b>
5.1	Explore neuron models and devices . . . . .	11
5.2	Manage projects . . . . .	11
5.3	Construct neuronal networks . . . . .	12
5.4	Simulate neuronal networks . . . . .	14
5.5	Explore network activity . . . . .	16
<b>6</b>	<b>Setup Guide</b>	<b>19</b>
6.1	Via Python Package . . . . .	19
6.2	Via Docker . . . . .	20
6.3	Via Singularity . . . . .	21
<b>7</b>	<b>Command API</b>	<b>23</b>
7.1	nest-desktop . . . . .	23
7.2	nest-server . . . . .	24
<b>8</b>	<b>Troubleshootings</b>	<b>25</b>
<b>9</b>	<b>Use NEST Desktop on EBRAINS</b>	<b>31</b>
<b>10</b>	<b>How to deploy NEST Desktop</b>	<b>33</b>
<b>11</b>	<b>Deploy NEST Desktop in Docker</b>	<b>35</b>
<b>12</b>	<b>Deploy NEST Desktop on HBP</b>	<b>37</b>
<b>13</b>	<b>Deploy NEST Desktop on bwCloud</b>	<b>39</b>
<b>14</b>	<b>How to develop NEST Desktop</b>	<b>41</b>
<b>15</b>	<b>Concept of the interface</b>	<b>43</b>

<b>16</b>	<b>Preparation</b>	<b>45</b>
<b>17</b>	<b>Source code</b>	<b>47</b>
<b>18</b>	<b>Icon description</b>	<b>49</b>
<b>19</b>	<b>The semantic versioning</b>	<b>51</b>
<b>20</b>	<b>Python Package Index (PyPI)</b>	<b>53</b>
<b>21</b>	<b>User documentation workflow</b>	<b>55</b>
<b>22</b>	<b>Sources</b>	<b>57</b>
	<b>Bibliography</b>	<b>59</b>

# CHAPTER 1

---

## NEST Desktop

---



Hello there!

NEST Desktop is a web-based GUI application for NEST Simulator, an advanced simulation tool for the computational neuroscience.

The app enables the rapid construction, parametrization, and instrumentation of neuronal network models.

It's so great that you want to use NEST Desktop.

**Let's get started.**

---

### NEST Desktop: A web-based GUI for NEST Simulator

Sebastian Spreizer<sup>1,2</sup>, Stefan Rotter<sup>1</sup>, Benjamin Weyers<sup>3</sup>, Hans E Plesser<sup>4</sup>, Markus Diesmann<sup>2</sup>

1. Bernstein Center Freiburg, Faculty of Biology, University of Freiburg, Freiburg, Germany
2. Institute of Neuroscience and Medicine (INM-6) Jülich Research Centre, Jülich, Germany
3. Department IV - Computer Science, Human-Computer Interaction, University of Trier, Trier, Germany
4. Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

Email: [spreizer@uni-trier.de](mailto:spreizer@uni-trier.de)

In the past few years, we have developed a web-based graphical user interface (GUI) for the NEST simulation code: NEST Desktop<sup>1</sup>. This GUI enables the rapid construction, parametrization, and instrumentation of neuronal network models typically used in computational neuroscience. The primary objective was to create a tool of classroom strength that allows users to rapidly explore neuroscience concepts without the need to learn a simulator control language at the same time.

Currently, NEST Desktop requires a full NEST installation on the user's machine, limiting uptake by a non-expert audience, and limiting the networks studied to such that can be simulated on a laptop or desktop machine. To ease the use of the app and the range of simulations possible with NEST Desktop, we want to separate the GUI from the simulation kernel, rendering the GUI in the web browser of the user, while the simulation kernel is running on a centrally maintained server.

NEST Desktop has a high potential to be successful as a widely used GUI for the NEST Simulator. To achieve this goal, in a first step all tools have to agree on a communication scheme and data format (using JSON) used to interact with a server-side running NEST instance in a session management, e.g. Docker or Singularity containers. Next, previously developed tools, namely the NEST Instrumentation App<sup>2</sup> and VIOLA<sup>3</sup>, will be integrated as a plugin into the app to extend visual modeling and analysis functionalities. In the course of this work, the use of an in situ pipeline<sup>4</sup>

<sup>1</sup> NEST Desktop <https://github.com/babsey/nest-desktop>

<sup>2</sup> NEST InstrumentationApp <https://github.com/compneuronmbu/NESTInstrumentationApp>

<sup>3</sup> Senk J, et al. (2018) VIOLA—A Multi-Purpose and Web-Based Visualization Tool for Neuronal-Network Simulation Output. *Front. Neuroinform.* doi: 10.3389/fninf.2018.00075

<sup>4</sup> Oehrl et al. (2018) Streaming Live Neuronal Simulation Data into Visualization and Analysis. Conference Paper doi: 10.1007/978-3-030-02465-9\_18

developed for neuronal network simulators will be considered to also enable the app to receive larger data sets from NEST during a running simulation, enhancing the interactivity of the app also for large simulations on HPC facilities.

We plan to develop and maintain NEST desktop sustainably. Therefore, we intend to integrate NEST desktop into the HBP infrastructure. Additionally, the open-source code of NEST Desktop will be published as a standalone distribution for teaching and training.

### Acknowledgements

This project has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under Specific Grant Agreement No. 785907 (Human Brain Project SGA2)

### References



## CHAPTER 3

---

### Past events

---

#### 2020

- 30 September - 1 October** Hand-on Session and Poster at [Bernstein Conference 2020](#), online (Berlin, Germany).
- 18 - 22 July** Tutorial with NESTML, presented by Dr. Linssen, at [CNS 2020](#), online (Melbourne, Australia).
- 29 - 30 June** Talk “NEST Desktop” at [NEST Conference](#) at [\[NMBU\]](#), online (As, Norway).
- 2 - 17 June** MSc course “Biophysics of Neurons and Networks” at [\[BCF\]](#), online (Freiburg, Germany).
- 16 April** Presentation and demo at [NeuroMat](#), online (Sao Paulo, Brazil).
- 3 - 6 February** Talk and Demo/Hand-on session at [HBP Summit and Open Days](#), Athene, Greece.

#### 2019

- 28 November** 2nd HPAC Platform Training, Heidelberg, Germany.
- 20 October** Live demo, presented by Prof. Plesser, at HBP Booth at SfN, Chicago, USA.
- 18 - 20 September** Poster/Live presentation at [Bernstein Conference](#), Berlin, Germany.
- 22 July** Talk and Tutorial/Hand-on session at [\[INM-6\]](#), Jülich, Germany.
- 18 July** NESTML/NEST-desktop integration workshop, [\[BCF\]](#), Freiburg, Germany.
- 24 - 25 June** Talk and Tutorial/Hand-on session “[NEST Desktop](#)” at NEST Conference at [\[NMBU\]](#), As, Norway.
- 16 April** Kick-Off workshop at [\[HCI\]](#), Trier, Germany.
- 25 - 31 March** Tutorial workshop for [IICSSS](#) at [\[BCF\]](#), Freiburg, Germany.

**11 - 22 February** BSc course “Simple Neuron Models” at [BCF], Freiburg, Germany.

### 2018

**26 - 27 September** Poster/Live presentation [NEST Desktop](#) at Bernstein Conference, Berlin, Germany.

**27 - 28 August** Technical meeting at [BCF], Freiburg, Germany.

**25 - 26 June** Talk “[NEST Web API](#)” at NEST Conference at [NMBU], As, Norway.

**23 - 27 April** MSc course “Biophysics of Neurons and Networks” at [BCF], Freiburg, Germany.

**12 - 23 February** BSc course “Simple Neuron Models” at [BCF], Freiburg, Germany.

### 2017

**19 - 20 December** Talk “[NEST Desktop](#)” at NEST Conference, Jülich, Germany.

**20 - 22 November** Live presentation with Ad Aertsen at [Neural networks mini school](#), Strasbourg, France.

**02 - 05 May** MSc course “Biophysics of Neurons and Networks” at [BCF], Freiburg, Germany.

**24 January** Talk (Informal Seminar) “[NEST Desktop](#)” at [BCF], Freiburg, Germany.

### 2016

#### December

- [NEST Desktop](#) is born in Freiburg.

# CHAPTER 4

---

## How to **use** NEST Desktop

---

### Quick start

The video shows few steps to construct network and explore activity. For more information, please read *detailed guides of usage*.

---

### Getting started in Terminal

```
docker run -i -p 5000:5000 -p 8000:8000 -t babsey/nest-desktop
```

If a image is not existed, it will be pulled from docker hub. Then, NEST Desktop is serving at `http://localhost:8000`.

For more information, please read *detailed guides of setup*.

---

### List of detailed guides

- *How to use NEST Desktop*
- *How to setup NEST Desktop*
- *Command API*
- *Troubleshooting*

### Guides for the infrastructure

- *Use NEST Desktop on EBRAINS*



## CHAPTER 5

---

### Usage Guide

---

This guides provides detailed documentation for the Graphical User Interface (GUI) of NEST Desktop.

---

**Note:** If you want to see quick start in NEST Desktop we prepared a [video](#) showing steps to construct networks and explore activity.

---

#### Getting Started

Once you start NEST Desktop, you see the start page a image of a laptop with NEST logo. At the bottom it shows a short description of NEST Desktop (left) and some useful links and the current version (right).








# NEST Desktop

An educational GUI for neuroscience



NEST Desktop is a web-based GUI application for NEST Simulator, an advanced simulation tool for computational neuroscience.

The application enables the rapid construction, parametrization, and instrumentation of neuronal network models. The primary objective is to provide an accessible classroom tool that allows users to rapidly explore neuroscience concepts without the need to learn a simulator control language at the same time.






**Documentation** <https://nest-desktop.readthedocs.io>

**Source Code** <https://github.com/babsey/nest-desktop>

**License** MIT License

**Current Version** 2.5.0-dev

**Contact** Sebastian Spreizer

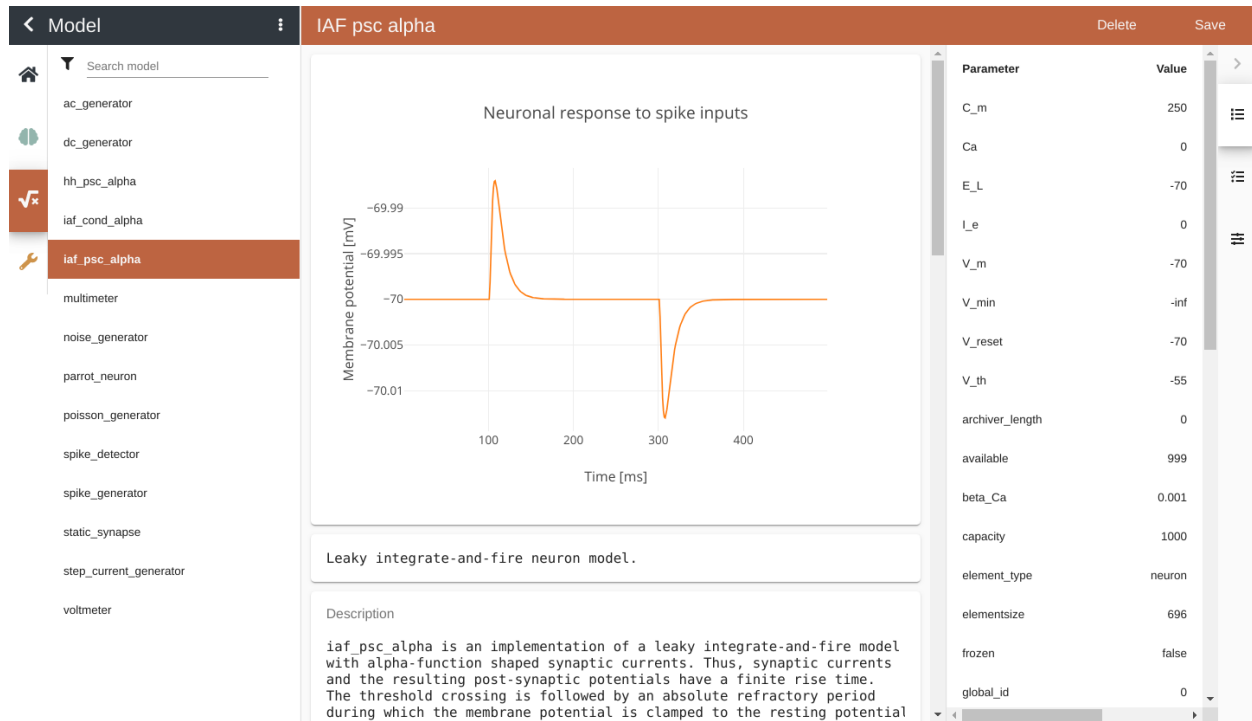
It checks settings and databases of the application. When the server responds, the server icon is colored to green.

---

**Note:** You can reload the page if NEST Desktop is somehow crashed.

---

## 5.1 Explore neuron models and devices



The model page provides you detailed documentation of models. When a selected model is a neuron, it also shows activity graph of neuronal response to excitatory and inhibitory spike inputs at 100 ms and 300 ms, respectively.

## 5.2 Manage projects

NEST Desktop has project management helping you to organize your networks and network activity. An important remark is that it stores only neuronal networks in database and activity will be lost after page reload. If you want to explore activity of the project, you have to start the simulation (See [Simulate neuronal networks](#)).

Clicking on a *New project* creates a new project where you can construct network from the scratch (See [Construct neuronal networks](#)). It is useful to give project a proper name that you can recognize your projects (The video shows a bad manner). An asterisk after project name (instead of time since created or updated) indicates that changes of the project are not stored in database.

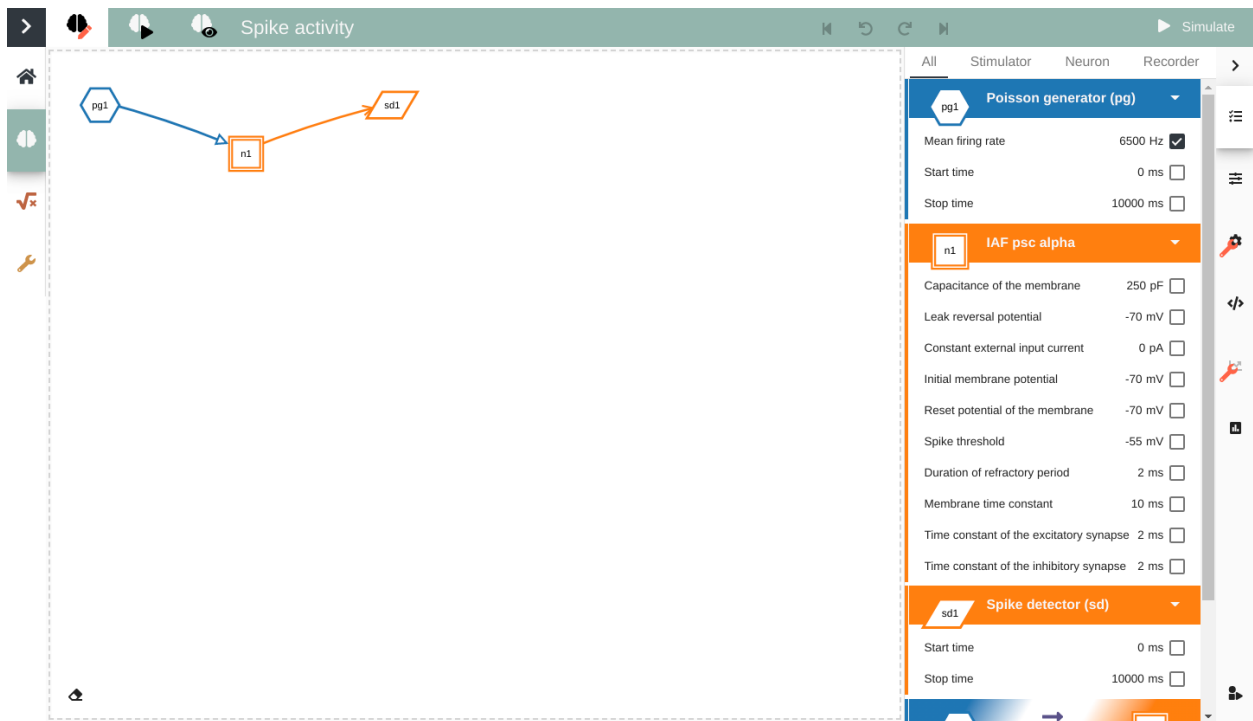
Below the search field it shows a list of projects where you can load existed networks. Clicking right mouse button on a project item in the list shows options to reload, duplicate, download or delete it. In the project menu, you find other methods to upload, download or delete all projects or to manage them.

The greatest advantage of NEST Simulator is the reproducibility of the activity of the same network, even on other machine. At result, NEST Desktop shows the same simulation events on different machines.

### Project revisions

You are able to checkout older version of the project data. An item of the context menu shows an option to call a list of project revisions. Since the database contains version control management, it is able to store and load older versions of the project.

## 5.3 Construct neuronal networks



If you want to construct network, you have to go to the network editor with the button showing red pen on brain icon. The network graph is composed of nodes (shapes) and connections (lines). Here, we explain steps to create and connect nodes.

### Create nodes



In order to create a new node you click the right mouse button in the content area, a *pie* panel with three letters appears to select a element type. A node is divided in three element types: stimulus (*S*), recording (*R*) devices and neuron (*N*). Then you select a element type and then it creates a specific node in the center of the *pie* panel.

### Node shapes

The specific shape defines an element type of a node:

- **Hexagon:** A stimulus device alias stimulator is an instrument which only produces signals towards target nodes.
- **Parallelogram:** A recording device alias recorder is also an instrument which observe states of a recordable node.
- **Square/Triangle/Circle:** A neuron node is the core engine of neuronal network model which received inputs from other nodes and produces specific output using intrinsic equation.

### Neuron shapes



The shape of neurons is differently represented by the set of synaptic weight of their connections.

- **Square:** Neurons without connections or mixed (positive and negative) synaptic weights to neurons.
- **Triangle:** Neurons with excitatory connections to neurons. All synapse weights are positive.
- **Circle:** Neurons with inhibitory connections to neurons. All synapse weights are negative.

### Connect nodes

Forming a network of nodes is defined by making connections between and within nodes. In order to connect nodes, you select a node, then move the mouse towards other node and finally click a target node. It creates a connection between selected source node and clicked target node.

### Node labels

Each node graph is labeled to identify the model of the node. By default, it creates direct current generator (*dc*) for stimulus, voltmeter (*vm*) for recording devices. Neuron are just labeled by *n*. You can see the neuron model in the controller view.

### Node colors

Nodes and connections contains parameter configurations which are displayed in the controller panel in the side navigation. The color of nodes helps you to associate the network graph with the controller as well as the visualization of the network activity. The color of lines is defined by the source node.

### Network controller

You are able to select node or synapse model in the network controller providing two view modes. One is the selection mode which is designed to select parameter items for its visibility. The other is the editing mode showing only visible parameters allows you to edit it.

### Network history

With network changes it makes a snapshot of the network afterwards and push to the network history list. With this network history you can undo or redo the network changes. A load from a network from the history is called *checkout network*.

## 5.4 Simulate neuronal networks

You can click *Simulate* button to start the simulation of your network. In a controller content you can have an look on script code.

## Script code

```

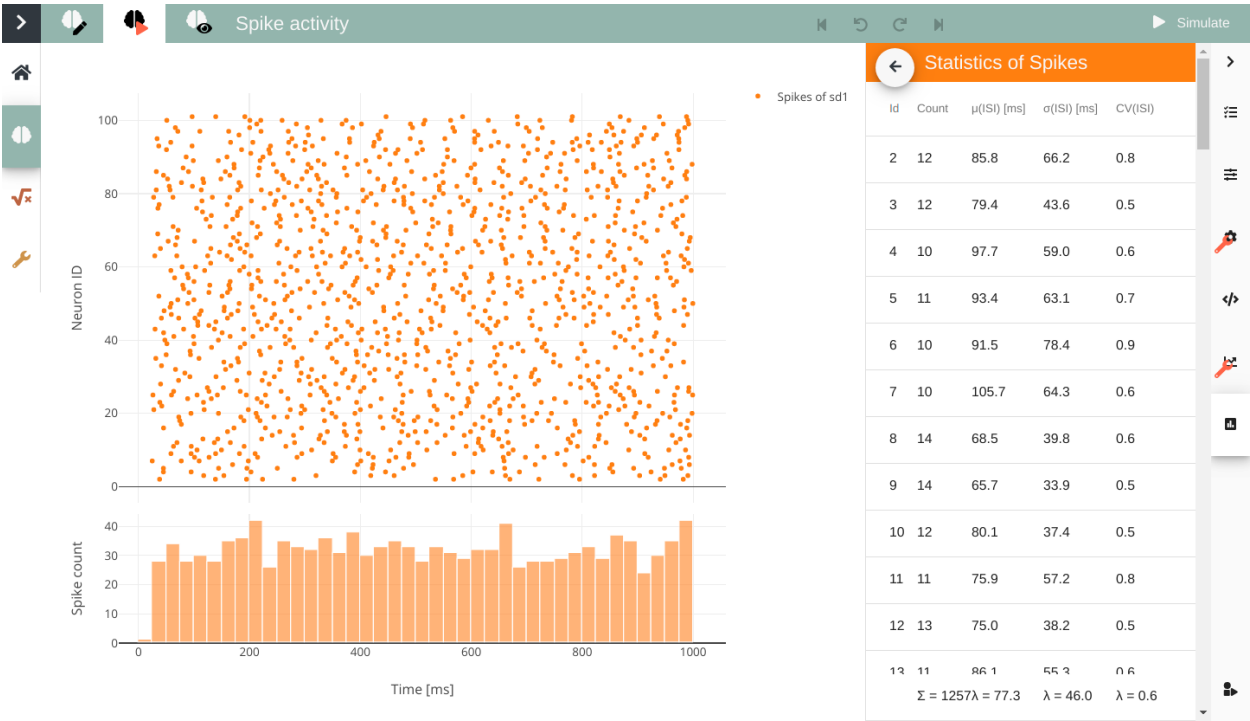
1 import nest
2 import numpy as np
3
4 nest.ResetKernel()
5
6
7 # Simulation kernel
8 np.random.seed(700)
9 nest.SetKernelStatus({
10     "local_num_threads": 1,
11     "resolution": 0.1,
12     "rng_seeds": np.random.randint(0, 1000, 1).tolist()
13 })
14
15
16 # Create nodes
17 pg1 = nest.Create("poisson_generator", 1, params={
18     "rate": 6500.0
19 })
20 n1 = nest.Create("iaf_psc_alpha", 100)
21 sd1 = nest.Create("spike_detector", 1)
22
23
24 # Connect nodes
25 nest.Connect(pg1, n1, syn_spec={
26     "weight": 10.0
27 })
28 nest.Connect(n1, sd1)
29
30
31 # Run simulation
32 nest.Simulate(1000.0)
33
34
35 # Collect activities
36 response = {
37     "kernel": {"time": nest.GetKernelStatus("time")},
38     "activities": [{
39         "events": nest.GetStatus(sd1, "events")[0],
40         "nodeIds": nest.GetStatus(nest.GetConnections(None,
41         sd1), "source"),
42     }]
43 }

```

NEST Desktop generates textual code from the constructed network. The generated code can be applied in any Python interface and you will obtain the same simulation results. Here, we show that the code semantics of the NEST Simulator is understandable and easily to learn.

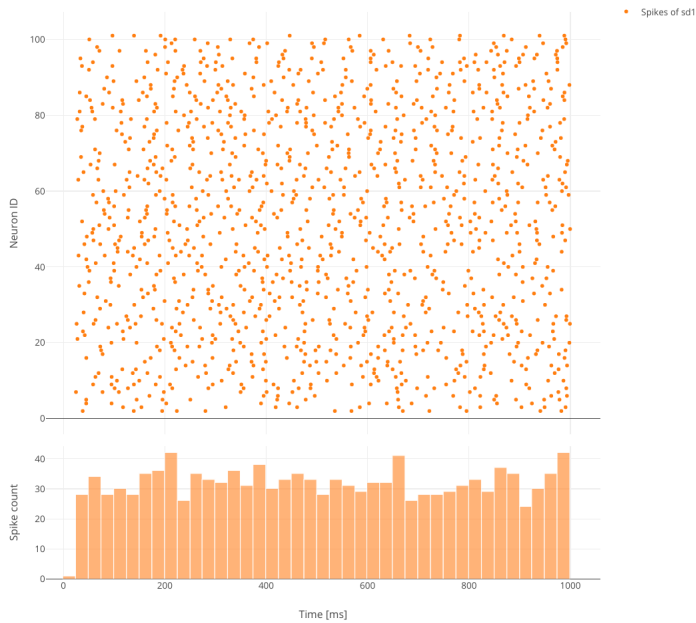
The graphical representatives of the nodes deliver arguments to the block of `nest.Create(*)` function. Next, connections supply specification for the block of `nest.Connect(*)` function. The function `nest.Simulate(*)` triggers simulation of your constructed network. All recording nodes fills a block to collect activities using the code in PyNEST `nest.GetStatus(rec, 'events')`. These activities contain neuronal properties, e.g. ids and positions, and activity events.

## 5.5 Explore network activity



Network activity is composed of neuronal properties (positions and ids of neurons) and recorded events from recording devices. Events can be subdivided in two groups: spike events or analog signals. Spike events contains times and ids of the senders emitting events to the recording devices which can be considered as collectors (`spike_detector`). Analog signals contains continuous quantities from the recording devices aka samplers (`voltmeter` or `multimeter`) interrogate their targets at given time intervals. Network activity can be explored in a graph or table.

### Activity graph



It displays either a chart graph or animated 3D graph for the spatial network forming layers in topology which neurons has geographical positions. The chart graph contains graphical panels organized in vertical stacks. Chart panels is introduced specifically to explore network activity with mouse interaction. It visualizes scatter graph for spike activity, line graph for analog signals. When you add one or more subpanels for spike activity it shows a histogram graph of spike times or of inter-spike intervals.

### Activity table

Statistics of Spikes				
Id	Count	$\mu$ (ISI) [ms]	$\sigma$ (ISI) [ms]	CV(ISI)
2	12	85.8	66.2	0.8
3	12	79.4	43.6	0.5
4	10	97.7	59.0	0.6
5	11	93.4	63.1	0.7
6	10	91.5	78.4	0.9
7	10	105.7	64.3	0.6
8	14	68.5	39.8	0.6

You can go to the table by clicking on *Activity statistics* tab button in the right side navigation. A table shows simple statistics e.g.  $\mu$  (mean) and  $\sigma$  (standard deviation) of activities for a recording device. Moreover, Spike

events show statistics of  $ISI$  (inter-spike interval) and  $CV_{ISI}$  (Coefficient of variation in inter-spike interval). Each row represents a recorded neuron (id) from a node connected to the selected recording device.

This guide provides a detailed documentation on how to install NEST Desktop.  
You can read the installation instructions by clicking one of these logos below:

### 6.1 Via Python Package

1. Both NEST Desktop and NEST Server require NEST Simulator supporting only in Linux systems.

**Note:** For more information read the full installing docs of NEST Simulator [here](#).

2. NEST Desktop is distributed on PyPI and can be installed with pip:

```
pip3 install nest-desktop [--user] [--upgrade]
```

It installs NEST Server (`nest-server`) automatically. For more information read the full installing docs [here](#).

3. Before you start the frontend NEST Desktop, you have to run NEST Server as a backend.

Start NEST Server:

```
nest-server start
```

NEST Server is now serving at `http://localhost:5000`.

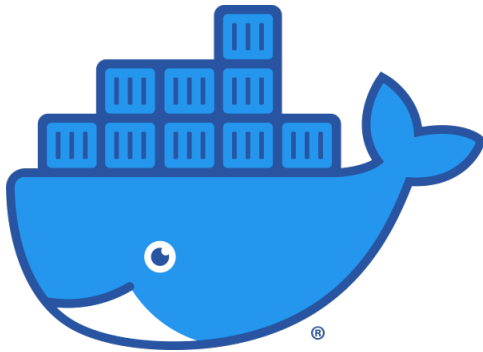
4. Start NEST Desktop (in other terminal session):

```
nest-desktop start
```

NEST Desktop is now serving at `http://localhost:8000`.

For more information read the full documentation of command API [here](#).

## 6.2 Via Docker



Docker is a virtualization software packaging applications and its dependencies in a virtual container that can run on any Linux server. It is available for a variety of operating systems, e.g. Linux, Mac and Windows. For more information [here](#).

1. Pull the image from docker hub:

```
docker pull babsey/nest-desktop
```

2. Start the docker container:

```
docker run -i -p 5000:5000 -p 8000:8000 -t babsey/nest-desktop
```

---

**Note:** In Docker container NEST Desktop is serving at port 8000 and NEST Server at port 5000. So, we need to bind ports (5000 and 8000) of host and container.

---

NEST Desktop and NEST Server are now serving at `http://localhost:8000` and `http://localhost:5000`, respectively.

### Arguments

You can find help text of docker arguments by `docker run --help`.

-p	Publish a container's port(s) to the host
-i	Keep STDIN open even if not attached
-t	Allocate a pseudo-TTY



## 6.3 Via Singularity

Singularity is an application container for Linux systems. For more information read the full documentation [here](#).

1. Clone working copy from repository:

```
git clone https://github.com/babsey/nest-desktop
```

2. Go to Singularity folder of NEST Desktop:

```
cd nest-desktop/singularity
```

3. Build singularity container (with sudo):

```
singularity build nest-desktop.sif nest-desktop.def
```

4. Start singularity container

```
singularity run nest-desktop.sif
```

NEST Desktop and NEST Server are now serving at `http://localhost:8000` and `http://localhost:5000`, respectively.



This documentation guide provides detailed information about the commands `nest-desktop` and `nest-server`.

### 7.1 nest-desktop

Show usage for the command `nest-desktop`

```
nest-desktop
```

Options for `nest-desktop`

```
nest-desktop <command> -h <host> -p <port>
```

#### Commands

**status** Status of NEST Desktop

**start** Start NEST Desktop

**stop** Stop NEST Desktop

**restart** Stop then start

#### Arguments

**-h** Host (Default: 127.0.0.1)

**-p** Port (Default: 8000)

## 7.2 nest-server

Show usage for for the command `nest-server`

```
nest-server
```

Options for `nest-server`

```
nest-server <command> [-d] [-h <HOST>] [-p <PORT>] [-u <UID>]
```

### Commands

**log** Monitor NEST Server  
**status** Status of NEST Server  
**start** Start NEST Server  
**stop** Stop NEST Server  
**restart** Stop then start

### Arguments

**-d** Demonize the process  
**-h** Host (Default: 127.0.0.1)  
**-p** Port (Default: 5000)  
**-u** User ID

---

### Troubleshootings

---

Having trouble getting something working? Got a question that the rest of our docs can't answer? Maybe we can help with some answers to commonly asked questions and troublesome spots.

#### Error messages

##### Server not found

**This error has two possible sources:**

- NEST Desktop has wrong URL of the server. See FAQ for NEST Server below.
- The server is down. Enter the url of the NEST Server, e.g. `localhost:5000` in web browser. For advanced users: Check in terminal whether NEST Server is running (`curl localhost:5000`).

##### Internal server error

NEST Simulator produces value error, e.g. `The value cannot be negative..`

#### Frequently asked questions (FAQ)

##### NEST Server

**How can I check NEST Server?** In the loading page you can click on menu icon of the server section to open menu. One of the menu item is to check NEST Server.

**How can I change the URL of the NEST Server?** In the settings of the NEST Server you find the form fields for the URL of the NEST Server.

### Model

**What is terminology of this model?** This model includes neuron, synapse and device models.

**How can I read the documentation of a model?** In the model section you find a list of models on the left side navigation. Click on one model, it request NEST Server for the documentation text of the selected model.

**How can I get default values of a model?** In the model section you find a list of models on the left side navigation. Click on one model, it request NEST Server for the default values of the selected model. The default values appear as a list on the right side bar.

### Project

**How can I make a new project?** In the navigation sidebar you find a list item `New Project` to create a new project.

**How can I duplicate a project?** In the project toolbox you find a button to clone project.

**What does the asterisk icon after project name mean?** It means that change in network was registered and the project is not saved in the database.

**How can I rename project?** In the navigation sidebar you find the input field of the project name. There you are able to change the name of the project.

**How can I save project?** In the left navigation of the page you find a floppy disc icon to save project.

**How can I delete a project?** In the context menu of the projects (by clicking mouse right button on a project) you find this method to delete corresponding project.

**How can I download a project?** In the context menu of the projects (by clicking on mouse right button on a project) you find this method to delete corresponding project.

**How can I delete several projects?** In the menu of the project in the top navigation you find an method to select projects. Then select several projects and then at the end you find the action to delete them.

**How can I download several projects?** In the menu of the project in the top navigation you find an method to select projects. Then select several projects and then at the end you find the action to download them.

**How can I download all projects?** In the menu of the project in the top navigation you find an method to download all projects.

**How can I upload projects?** In the menu of the project in the top navigation you find an method to upload projects from a file.

### Network

**Where can I find network controller?** You find network controller by clicking on top icon ('project diagram') right to controller. Node and link controller are parts of the network controller.

**Where can I find network settings?** You find a 'bezier-curve' button towards network settings in the network

editor. Alternatively, you find a network settings in the settings section.

**How can I empty network?** In network sketch you find bottom left a trash button that empties the network.

**How can I create nodes?** In network sketch you can click the right mouse button, then it appears a selector panel to select an element type of the node.

**How can I connect nodes?** In network sketch you can select a node as a source node, move the mouse towards target node and then click the target node.

**How can I keep selected node to connect other nodes?** Hold pressing CTRL key before connecting nodes.

**How can I (un)select node / link?** When a node or link is selected you can press 'ESC' to unselect it or in network sketch you can click on node or link to select it. Click on background area of the network sketch or on the selected in the network controller unselect node or link. An other method to (un)select is to click on node shape or link toolbar in the network controller on the right side.

**How can I colorize nodes?** You find the method to color in the context menu of the node by clicking the right mouse button on node shape in network sketch or node toolbar in the controller.

**How can I change the color cycle of nodes?** In the network settings you find the way to change the color cycle.

**How can I delete node/link?** You find this method in the context menu of the node or link by clicking the right mouse button on node / link shape in network sketch or on node / link toolbar in the network controller.

**How can I modify parameters?** You find a list of parameters in network controller. If they are not visible, switch to network selection to check the visibility of the parameters.

**How can I reset all parameter value?** In the context menu of node or link you find the method to reset all parameters of the corresponding node or link.

**How can I reset a parameter value?** In the context menu of a parameter (by clicking right button on a parameter) you find the method to reset parameter. It also shows the default value of the parameter.

**How can I set inhibitory connection?** You can set weights to negative value in the link controller.

**How can I get distribution for parameters?** In the current version you are able to activate the distribution of the parameters in kernel, weights and delay of the spatial projections.

**How can I set node in spatial mode?** In the context menu of the node, you can (un)set the spatial mode of the node.

**How can I add mask for projection?** When source and target nodes are set to spatial, then you find mask item of the spatial projections in the link controller. Click on the mask item to modify it.

**How can I generate grid/free positions?** When the node is set to spatial, position item will replace the population item. Click on position item to get a position panel. Modify a value will generate positions, at the end of the panel you find a button to generate positions.

**How can I add positions?** In the positions panel you find a text area of the positions, you can add custom positions.

**How can I modify the slider?** In the context menu of the parameter, you find a method to alter slider settings.

**How can I generate array?** In the context menu of the array parameters (e.g. spike times of spike generator) you find the method to generate array. It opens a dialog for array generation.

## Simulation

**How can I start simulation?** Click on the *Simulate* button in top right of the page to start the simulation.

**How can I stop simulation?** Unfortunately, the option to stop simulation is not implemented.

**Why is the *Simulate* button shaking/bouncing?** It means that changes of the network was registered and the activity graph might not match with the network. It reminds you to start the simulation.

**How can I activate simulation after change?** In the context menu of the *Simulate* button (by clicking mouse right button) you find an option to activate simulation after change.

**How can I activate simulation after load?** When you click on a project In the context menu of the *Simulate* button (by clicking mouse right button) you find an option to activate simulation after load.

**How can I activate simulation after checkout?** When you go to other network version of the history, it automatically start the simulation. In the context menu of the *Simulate* button (by clicking mouse right button) you find an option to activate simulation after load.

**Where can I find simulation controller?** The simulation controller appears by clicking on the cog icon right to controller side bar. The kernel controller is a part of the simulation controller.

**Where can I set simulation time?** In the simulation controller you find a simulation section. There you are able to change simulation time.

**Where can I modify kernel of NEST Simulator?** You find a kernel section for the NEST Simulator in the simulation controller.

**Where can I change time resolution of the kernel?** You find time resolution for the NEST Simulator in the simulation controller.

**Warning:** Please verify that the resolution of the recorders are larger than the resolution in the kernel.

**Where can I change seed?** In the simulation controller you find a simulation section. There you are able to change seed value.

**How can I activate seed randomization?** You find an option to activate seed randomization in the simulation controller.

**How can I find Python script code of the simulation?** In tabs of the right side navigation you find a symbol <> for code editor.

**How can I find data for the project in JSON format?** When the development mode is on, in one tab of the simulation details you find JSON data of the current project.

## Activity

**How can I download activity data?** In the context menu of the simulation (on play icon) you find a menu item to download data and records of the current simulation.

## Chart

**Where can I find chart controller?** You find animation controller by clicking on the 'chart' icon in the stacked menu left to controller.

**How can I drag/zoom the chart?** Select the mode in the mode bar (top). Then click on the chart for dragging or zooming.

**How can I reset to default view?** Click on the house icon in the mode bar (top) to reset to default view.

**How can I download plot of the chart?** Click on the photo icon (top) to download plot of the chart. It saves into SVG format.



**How can I migrate chart to Plotly Chart Studio?** Click on the text “Edit chart” (bottom) to see the chart in Plotly Chart Studio.

**How can I modify bin size of the PSTH?** In the chart controller you find tick slider to modify bin size.

**How can I change the mode of the PSTH?** In the chart controller you find options to change the mode of the PSTH.

**How can I change the label of axes or the title?** Click on the label of the axes or the title to change it.

**How can I hide/show dots/lines?** Click on the legend to alter the visibility of the dots/lines.

## Animation

**Where can I find animation controller?** You find animation controller by clicking on the ‘braille’ icon.

**How can I stop animation?** Go to animation controller. You find a pause icon to stop animation.

**How can I increase/decrease animation speed?** In the animation controller you find forward or backward to alter animation speed.

**How can I change colorscale of dots?** In the animation controller you find a colormap of the current colorscale. Next below of it you find an options to select colorscale.

**How can I change size of dots?** In the animation controller you find slider of the dot size.

**How can I add trailing for dots?** It only works in the animation of the spikes.

**How can I rotate camera?** Hold the mouse button on the animation area and then move it to rotate the camera.

**How can I activate camera motion?** In the animation controller you can increase the speed of the camera motion.

## Settings

**Where can I find settings?** You find settings by clicking on the cog icon right to navigation side bar. The settings are stored as ‘local storage’ of the browser.

**How can I change settings?** You can change settings in the settings section by clicking on cog icon next to navigation side bar.

**How can I reset settings?** In the loading page you can click on menu icon of the settings section to open menu. One of the menu item is to reset settings.

**How can I reset databases?** In the loading page you can click on menu icon of the database section to open menu. One of the menu item is to reset databases.

**Where can I find the database?** The databases are stored as ‘Indexed DB’ of the browser.

**How can I activate automatic start?** In the settings of the application you find the option to activate automatic start.

**How can I switch to development view?** In the settings of the application you find an option to switch to development view.



---

### Use NEST Desktop on **EBRAINS**

---



EBRAINS is a platform infrastructure for neuroscience. NEST Desktop is available as a prototype online. Go to the [description page on EBRAINS](#).

#### Access to NEST Desktop

---

**Note:** An *HBP* account is required to access NEST Desktop.

---

If you have an *HBP* account, follow the link below:

- <https://nest-desktop.apps.hbp.eu>.

You will see the login interface of *HBP*. Enter your account name and password, then it forwards you to NEST Desktop.

#### Request account of *HBP*

If you do not have *HBP* account, you can ask a member of *HBP* to invite you or request an account here:

- <https://services.humanbrainproject.eu/oidc/login>

### Trouble shootings

**Why cannot I find NEST Server?** Sometimes the issue is resolved when you check NEST Server.

If not, the problem lies in cookies and site data for the user authentication of the *HBP*. That cookie expires after a session. This happens when you re-access to `https://nest-desktop.apps.hbp.eu` after the browser is closed.

The simple solution is to reload the page that you can re-login.

A persistent solution is to set the browser configuration that it deletes cookies and site data when the browser is closed (It works in Firefox).

---

### How to **deploy** NEST Desktop

---

This guide provides detailed documentation on how to deploy NEST Desktop.  
You can read the deployment instructions by clicking one of these images below:

---

#### Table of content

- *Deploy NEST Desktop in Docker engine*
- *Deploy NEST Desktop on Human Brain Project (HBP) resources*
- *Deploy NEST Desktop on bwCloud resources*



---

## Deploy NEST Desktop in **Docker**

---



Docker is a virtualization software packaging applications and its dependencies in a virtual container that can run on any Linux server. It is available for a variety of the operating systems, e.g. Linux, Mac and Windows. For more information follow the link [here](#).

### Requirements

- Docker engine

### Installation

The image from docker hub contains all required software (NEST Desktop, NEST Server and NEST Simulator). For more information, follow the link [here](#).

```
docker pull babsey/nest-desktop
```

### Getting started

## NEST Desktop

---

Start docker container with arguments.

```
docker run -i -p 5000:5000 -p 8000:8000 -t babsey/nest-desktop
```

NEST Desktop and NEST Server are now serving at `http://localhost:8000` and `http://localhost:5000`, respectively.

---

**Note:** In Docker container NEST Desktop is serving at port 8000 and NEST Server at port 5000. So, we need to bind ports (5000 and 8000) of host and container.

---

## Arguments

You can find help text of docker arguments by `docker run --help`.

-p	Publish a container's port(s) to the host
-i	Keep STDIN open even if not attached
-t	Allocate a pseudo-TTY

## Acknowledgments

Thanks for the help:

- Steffen Graber (Docker hub for NEST Simulator)



---

### Deploy NEST Desktop on **HBP**

---



## Human Brain Project

The documentation shows how to deploy NEST Desktop from docker hub on *HBP* resources.

HBP provides two OpenShift infrastructures

- <https://okd-dev.hbp.eu> for the development.
- <https://okd.hbp.eu> for the production.

---

**Note:** I strongly recommend to use the development page before you push NEST Desktop on `https://openshift.hbp.eu`.

---

#### Requirements

- OC Client Tools (<https://www.okd.io/download.html#oc-platforms>)

### Step to deploy (in development)

First configure OpenID Connect client for NEST Desktop [here](#).

You can copy command line from the web console of `https://okd-dev.hbp.eu`.

Login to openshift-dev.hbp.eu:

```
oc login https://okd-dev.hbp.eu:443 --token=<token>
```

Get status of current project:

```
oc status
```

You find a configuration and a bash files for setting up NEST Desktop on *HBP*. Execute the bash script and in the end it shows the IP needed for *HBP* authentication (see below):

```
cd infrastructure/okd-dev.hbp.eu
bash setup-nest-desktop.sh
```

### Further usage

Scaling up the replicas (pods or nodes):

```
oc scale --replicas=2 dc nest-desktop
```

Monitor log of a pod (Get pod name: `oc get pod`):

```
oc exec <pod> -- nest-server log
```

### HBP Authentication and redirecting

To access to NEST Desktop on HBP infrastructure, an authentication of *HBP* membership is requested. You find the codes on <https://github.com/babsey/hbp-auth>.

Here are the steps how to setup authentication and redirecting to NEST Desktop properly. Before you have to modify the environment for *HBP* authentication, i.e. `OIDC_CLIENT_ID`, `OIDC_CLIENT_SECRET` and `CLUSTER_IP` of NEST Desktop (which is printed after setting up NEST Desktop).

```
cd projects/nest-desktop-dev
bash setup-nest-desktop-hbp-auth.sh
```

### Deploy Production

If NEST Desktop is ready for the production, meaning to deploy on `apps.hbp.eu`. Perform all steps same as in the development (`apps-dev.hbp.eu`).

### Acknowledgements

Thanks for the help to integrate NEST Desktop on HBP resources:

- Alberto Madonna (Concepting)
- Collin McMurtrie (Contacting)
- Fabrice Gaillard (Concepting of user authentication)
- Jonathan Villemaire-Krajden (Concepting)
- Martin Jochen Eppler (Contacting)

---

## Deploy NEST Desktop on **bwCloud**

---

The guide provides step-by-step documentation on how to deploy NEST Desktop on bwCloud resources. The *bwCloud* is a cloud assigned for the universities in Baden-Württemberg, Germany. For more information, follow the link: <https://www.bw-cloud.org/>.

The *bwCloud* is running on OpenStack, an open source platform using pooled virtual resources to build and manage clouds. For more information, please follow the link: <https://www.redhat.com/en/topics/openstack>.

### Requirements

- Packer (<https://www.packer.io/downloads.html>)
- Ansible (<https://releases.ansible.com/ansible/>)

### Deploy NEST Desktop

1. Download Openstack RC File from the dashboard, follow the steps: “Compute” -> “Access & Security” -> “API Access” -> “Download OpenStack RC File”

2. Source the RC file:

```
source Project_<userID>-openrc_v3.sh
```

3. Modify the configurations (i.e. source image and networks for the builder) in `infrastructure/bwCloud/nest-desktop.json`.

4. Build an image on *bwCloud*

```
cd infrastructure/bwCloud  
/path/to/packer build nest-desktop.json
```

### Acknowledgements

Thanks for the help to integrate NEST Desktop on *bwCloud*:

- Bernd Wiebelt
- Jonathan Bauer
- Michael Janczyk
- Manuel Messner
- Christopher Ill

---

## How to **develop** NEST Desktop

---

This is the developer guide, providing more detail on how to develop NEST Desktop.

---

### Table of content

- *The concept of the interface*
- *Prepare the environment for the development*
- *Working on source code of NEST Desktop*
- *Description of icons*
- *The semantic versioning*
- *Build and deploy NEST Desktop for Python Package Index*
- *User documentation workflow*
- *A list of sources containing in the code*



## CHAPTER 15

---

### Concept of the interface

---

#### **General layout concept of the interface.**

NEST Desktop contains three columns with clear function. Left column shows a navigation to route pages in the primary outlet. Center column renders main content of the page. Right column displays controller content for the modification.

#### **Pages**

NEST Desktop has three route pages (Project, Model, Settings). Colored buttons located in the left side of the page shows home icon for the start page. Other icons buttons renders the content of the navigation outlet. The brain icon refers to project page, equation icon to model page and wrench icon to setting page.

#### **Page colors**

Project button is summergreen, model button is crail and setting button is sycamore. These color palette were selected at the [colormind website](#).

#### **Navigation outlet (left)**

The navigation content displays a black toolbar in the header showing with the name of the page and an icon for the menu. Subjacent to the header it renders a list of items triggers loading pages in the content outlet.

#### **Primary outlet (center)**

The content outlet renders the page content via the URL. The project page displays network editor and activity

explorer. The model page shows the detailed information of models which can be used in NEST Simulator. The setting page shows an overview of all settings for various components of the app. The toolbar in the header stretches over center and right bars and it is color-coded by a loaded page.

### Controller bar (right)

The controller enables users to change values. The network controller displays list of nodes and connections.

### Right mouse button



When a component provides specific context menu triggering by right mouse button, an icon of mouse-right-button-click appears in left bottom of the page.



# CHAPTER 16

---

## Preparation

---

NEST Desktop is written in HTML5 but compiled by Angular, a web framework written in TypeScript. For Angular you need to install nodeJS (+ yarn).

I have prepared Singularity recipe that build an environment with required packages for the development. You can find the definition file in `singularity/nest-desktop-dev.def` for building Singularity container.

### Requirements

- Python 3.4 or higher
- NEST Server, NEST Simulator (Test app)
- nodeJS, npm, yarn (Build app)
- setuptools, wheel, twine (for PyPI)
- sphinx, sphinx rth theme (for Readthedocs)

### Source code

The source code of NEST desktop is hosted at <https://github.com/babsey/nest-desktop>

Clone NEST Desktop from the github repository:

```
git clone https://github.com/babsey/nest-desktop
```

### Environment for development

The definition file `nest-desktop/singularity/nest-desktop-dev.def` contains an adequate environment to develop and build NEST Desktop.

Build a singularity image:

```
singularity build nest-desktop-dev.sif nest-desktop/singularity/nest-desktop-dev.def
```

Go to shell of singularity container:

```
singularity shell nest-desktop-dev.sif
```



## CHAPTER 17

---

### Source code

---

#### Usage of yarn

Install node modules for NEST Desktop source code:

```
yarn install
```

Check if any node package are outdated:

```
yarn outdated
```

Upgrade outdated packages:

```
yarn upgrade
```

Serve Angular Live Development Server serving at <http://localhost:4200>

```
yarn start
```

#### Usage of npm

Install node modules for NEST Desktop source code:

```
npm install
```

Check if any node package is outdated:

```
npm out
```

Update node package:

```
npm up
```

Serve Angular Live Development Server serving at <http://localhost:4200>

```
npm start
```

### Setup

Install NEST Desktop from source code using `pip` (where it finds `setup.py`). Best method is to install it in user home using `pip install --user`.

```
python3 -m pip install --user --no-deps -e .
```

### Getting started

You can read *Getting started* in User Documentation to start NEST Desktop. Starting NEST Desktop `nest-desktop start` equivalent to the command:

```
python3 -m nest_desktop.app
```

## CHAPTER 18

---

### Icon description

---

#### **Model**

It refers to model management.

It shows only models which are implemented in NEST Desktop. If it is disabled (with a ban icon) it shows all models of NEST Simulator.

#### **Project**

The brain icon refers to the project to construct and simulate neuronal networks to explore network activity.

The eye icon refers to the lab book.

#### **Network**

The brain-edit icon refers to the network editor.

#### **Simulation**

The play icon refers to the activity explorer.

The play icon refers to the simulation.

#### **Activity**

The chart-line icon refers to line chart of analog signals.

The chart-bar icon refers to histogram chart.

The braille icon refers to the scatter animation of the activity.

#### **Settings**

The wrench icon refers to settings.



---

## The semantic versioning

---

During the course of the development the implementation of featured into NEST Desktop are validated its compatibility. In this concept a general rule of the semantic versioning NEST Desktop was introduced that the operational capability of the application can be guaranteed. The formal convention of the version releases for specifying compatibility in NEST Desktop uses a three-part number:

### **A major number**

It is incremented for the compatibility with the NEST Simulator. It implies that the major version of the NEST Desktop (2.x.x) has to match with the one of NEST Simulator (currently 2.x.x).

### **A minor number**

It is a breaking feature such as a new library or a minor changes of the data structure. It means that it could cause the compatibility issue and the user need to adapt to use the same minor version of NEST Desktop as well as of NEST Server.

### **A patch number**

It is a bugfix and non-breaking features added to the code. The user is able to work with different patch versions of NEST Desktop and NEST Server.

NEST Desktop checks the compatibility in the loading page and provides options to reset settings and databases. Otherwise, they might be incompatible.





## CHAPTER 20

---

### Python Package Index (PyPI)

---

To build and deploy NEST Desktop on PyPI is a crucial step for the development. With it, Docker hub can update NEST Desktop and NEST Server in babsey/nest-desktop image.

#### Steps to build and deploy

Current work directory: `nest-desktop`.

The Python Package Index **nest-desktop** includes an executive command `nest-desktop` and a Python library `nest_desktop`.

First update the version of `nest-desktop` in `src/packages.json` and in `nest_desktop/__init__.py`.

Then generate app package using yarn. It builds the folder `nest_desktop/app`:

```
yarn run build
```

Next, remove the folders:

```
rm -rf build/ dist/ nest_desktop.egg-info/
```

Then generate distribution packages of *nest-desktop* for PyPI:

```
python3 setup.py sdist bdist_wheel
```

Finally, upload *nest-desktop* to PyPI:

```
python3 -m twine upload dist/*
```



## CHAPTER 21

---

### User documentation workflow

---

We use Sphinx to generate documentation and Read the Docs to publish it. Sphinx uses reStructuredText. To learn more about the syntax, check out this quick reference.

#### **Peparation**

Current work directory: `nest-desktop/docs`. To install sphinx and readthedocs theme via pip:

```
python3 -m pip install sphinx sphinx_rtd_theme
```

#### **Rendering HTML offline**

Build sphinx documentation in `docs/_build` folder offline:

```
rm -r _build/; make html
```

#### **Pushing to Read the Docs**

It automatically builds docs for master when pulling commits to master. Docs for latest and stable depends on their github tags.



## CHAPTER 22

---

### Sources

---

#### Images

**mouse-button-right-click.png** <https://www.hiclipart.com/free-transparent-background-png-clipart-iplqp>

**nest.svg** <https://nest-simulator.org/>

**spike-transmission-carlos.svg** Carlos Toledo Suárez, a former PhD of Abigail Morrison lab, passed away in 2013)



---

## Bibliography

---

- [BCF] Bernstein Center Freiburg, Faculty of Biology, University of Freiburg, Freiburg, Germany
- [HCI] Department IV - Computer Science, Human-Computer Interaction, University of Trier, Trier, Germany
- [INM-6] Institute of Neuroscience and Medicine (INM-6), Jülich Research Center, Jülich, Germany
- [NMBU] Norwegian University of Life Sciences, As, Norway