# Collector v1.5.0 setup and quickstart manual

# Installation of collector

Collector is distributed as a tgz compressed file.

Uncompress it and follow the installation instructions in the README file.

# Starting Collector web application server

Open a terminal window.

Move to the collector installation path. For example:

[[collector@collector](mailto:collector@collector) ~]$ cd /opt/collector

Start the web application by typing:

[[collector@collector](mailto:collector@collector) ~]$ startCollector

Now you will see on the terminal the Collector starting messages.

# Stopping collector web application

Open a terminal window.

Move to the collector installation path. For example:

[[collector@collector](mailto:collector@collector) ~]$ cd /opt/collector

Stop the web application by doing:

[[collector@collector](mailto:collector@collector) ~]$ stopCollector

# Troubleshooting starting web application

If you find difficulties starting the application you can try to stop previous collector instances that were already running:

[[collector@collector](mailto:collector@collector) ~]$ stopCollector

[[collector@collector](mailto:collector@collector) ~]$ startCollector

# Accessing Collector web application

Open a web browser and enter at the top web address line:

[http://localhost:900](http://localhost:9000/)1

# Collector Command line

Collector has a full command line interface.

To access it you have to open a terminal. Type “collector” to get a brief description of the commands available:

[[collector@collector](mailto:collector@collector) ~]$ collector

Collector command line commands:

collector listprotocols

* Lists the protocols available in the system. In the current implementation:

|  |  |  |  |
| --- | --- | --- | --- |
| job\_filtering\_id | job\_filtering\_description | filter\_description | curation\_order |
| 1 | No filtering | NoFiltering | 1 |
| 2 | LipRo5/Atom Validation + Activity | Activity | 1 |
| 2 | LipRo5/Atom Validation + Activity | LipinskiRo5 | 2 |
| 2 | LipRo5/Atom Validation + Activity | ValidateAtoms | 3 |
| 3 | LipRo5/Atom Validation + IC50 | IC50 | 1 |
| 3 | LipRo5/Atom Validation + IC50 | LipinskiRo5 | 2 |
| 3 | LipRo5/Atom Validation + IC50 | ValidateAtoms | 3 |
| 4 | LipRo5/Atom Validation + Inhibition | Inhibition | 1 |
| 4 | LipRo5/Atom Validation + Inhibition | LipinskiRo5 | 2 |
| 4 | LipRo5/Atom Validation + Inhibition | ValidateAtoms | 3 |
| 5 | LipRo5/Atom Validation + Ki | Ki | 1 |
| 5 | LipRo5/Atom Validation + Ki | LipinskiRo5 | 2 |
| 5 | LipRo5/Atom Validation + Ki | ValidateAtoms | 3 |

Every filtering protocol is identified by job\_filtering\_id. The curation\_order defines the order by which filters are applied. For example, the protocol 2 applies sequentially the filters

* Activity
* LipinskiRo5
* ValidateAtoms

The “No filtering” is protocol that applies no filtering; it only extracts the raw data from the source.

collector newjobuniprotid –-protocolid <protocol\_id> --uniprotid <uniprotaccession> --jobdescription <job\_description>

Defines a new job based on:

* <protocol\_id>: the protocol to apply (obtained in listprotocols call)
* the <uniprotaccession> of the target
* <job\_description>: a descriptive text of the job

collector executejob –jobdescription <job\_description>

Executes the job with <job\_description>

collector listjobexecutions --jobid <job\_id>

Lists all the executions of the given job\_id

collector export --raw **-–jobexecutionid <job\_execution\_id> --datatoexport activities|compounds –exportformat sdf|csv –filename <filename>**

Exports the data obtained in the job\_execution\_id depending on:

* activities: extracts the detailed activity data
* compounds: extracts the detailed activity data aggregated at compound level (median of all the activities reported for the same compound)
* sdf or csv: SDF 2D format or plain text tab-separated plain text file.
* <filename>: Filename to export.