Sirius: Concept of Operations

# General Concept

The newly developed Sirius software package for traffic simulation and analysis is to be used as

* Standalone program invoked from command line with necessary parameters:

$ sirius.jar <command> [parameters]

* Web service implemented through a servlet using the same set of parameters:

http://<web address>/sirius.jsp?cmd=<command>[&p=<parameters>]

The Sirius package is to rely on a database in two possible ways:

* Database integrated into the package – Java based Derby DBMS;
* Database separated from the package – PostgreSQL.

Both, the command line program as well as the servlet can work with the database in either of these two modes. The database type, the database host and the connection port are specified in the environment variables SIRIUS\_DB, SIRIUS\_DB\_HOST and SIRIUS\_DB\_PORT respectively. When these environment variables are not set, the default values pointing to localhost and Derby are used.

# Commands

The list of command is a subject of discussion and potential change. The Sirius commands fall into the three groups:

* Working with scenarios;
* Working with traffic data;
* Administrative commands.

## Commands Working with Scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Parameters** | **Returns** | **Description** |
| import (i) | * Scenario file name/ URL | Scenario ID | Parses the XML with scenario and inserts the scenario information into the database making new copies of the data objects. |
| update (u) | * Scenario file name/ URL | Scenario ID | Differs from the ‘import’ command in how in handling already existing scenario objects: it updates them rather than creating new instances. |
| export (e) | * Scenario ID | Name/ URL of the scenario file | Generates XML file conforming sirius.xsd from the scenario data in the database. |
| calibrate (c) | * Scenario ID * Date range | New scenario ID | Performs calibration for the given scenario and given date range, and generates new scenario with calibrated parameters and, if necessary, imputed inputs. |
| simulate (s) | * Scenario ID * Settings file name/ URL (optional) | Run number(s) | Runs a single or a batch of simulations generating raw data, which is recorded in the database. |
| * Scenario file name * Settings file name/ URL (optional) |
| simulate\_output (so) | * Scenario ID * Settings file name/ URL (optional) | Name/ URL of the output file | Runs a single or a batch of simulations generating raw data, which is dumped into the XML file conforming sirius\_output.xsd, and is not recorded in the database. |
| * Scenario file name * Settings file name/ URL (optional) |
| simulate\_process (sp) | * Scenario ID * Settings file name/ URL (optional) | Run number(s) | Runs a single or a batch of simulations generating raw data, which is recorded in the database, then immediately it runs the data grinding function – see ‘process’ command below. |
| * Scenario file name * Settings file name/ URL (optional) |
| list\_scenarios (ls) |  | List of scenario IDs with additional information if any | Lists all scenarios existing in the database. |
| list\_runs (lr) | * Scenario ID | List of Run numbers with corresponding date/time information | Lists all existing run numbers from the database for the given scenario. |

## Commands Working with Data

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Parameters** | **Returns** | **Description** |
| load (l) | * Data format (e.g. ‘PeMS’) * Data file/ URL |  | Loads traffic data in the known format into the data tables. |
| process (p) | * Scenario ID * Run number(s) (optional) |  | Performs data aggregation in 1, 5, 15 minute, 1 hour and 1 day intervals, and computes performance measures. |
| output (o) | * Scenario ID * Run number(s) (optional) * Aggregation: ‘raw’ | ‘1min’ | ‘5min’ | ‘15min’ | ‘1hour’ | ‘1day’ | ‘total’ | Name/ URL of the output file | Generates XML file in the format conforming sirius\_output.xsd for the data in the database in the given aggregation level. |
| list\_aggregations (la) | * Scenario ID * Run number(s) (optional) | List of computed aggregations – subset of {‘raw’, ‘1min’, ‘5min’, ‘15min’, ‘1hour’, ‘1day’, ‘total’} | Lists computed aggregation levels for the given data. |
| link\_data\_total (ldt) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_data\_total’ table into CSV file. |
| link\_data\_detailed (ldd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_data\_detailed’ table into CSV file. |
| link\_performance\_total (lpt) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_performance\_total’ table into CSV file. |
| link\_performance\_detailed (lpd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_performance\_detailed’ table into CSV file. |
| route\_data\_total (rdt) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_data\_total’ table into CSV file. |
| route\_data\_detailed (rdd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘route\_data\_detailed’ table into CSV file. |
| route\_performance\_total (rpt) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_performance\_total’ table into CSV file. |
| route\_performance\_detailed (rpd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘link\_performance\_detailed’ table into CSV file. |
| node\_data (nd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested node data into CSV file. |
| signal\_data (sd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘signal\_data’ table into CSV file. |
| signal\_phase\_performance (spp) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested data from ‘signal\_phase\_performance’ table into CSV file. |
| detection\_data (dd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested sensor data into CSV file. |
| probe\_data (pd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested probe data into CSV file. |
| controller\_data (cd) | * Scenario ID * Run number(s) (optional) | Name/ URL of the resulting CSV file | Dumps the requested controller data into CSV file. |
| report (r) | * Request file/ URL | * Name/ URL of the report PDF * Name/ URL of the report PPT | Generates the requested report out of the database data in PDF and PPT format. |

## Administrative Commands

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Parameters** | **Returns** | **Description** |
| init |  |  | Initializes the database. If the database exists, it is dropped and recreated. |
| clear\_data (cld) | * Scenario ID * Run number(s) (optional) * Aggregation: ‘raw’ | ‘1min’ | ‘5min’ | ‘15min’ | ‘1hour’ | ‘1day’ | ‘total’ (optional) |  | Erases data for given runs in specified aggregations. |
| clear\_processed (clp) | * Scenario ID * Run number(s) (optional) |  | Erase all data resulting from the ‘process’ command for given runs. |
| clear\_scenario (cls) | * Scenario ID |  | Erase the given scenario and all the related data. |
| clear\_all (cla) |  |  | Erases everything from the database. |
| version (v) |  |  | Returns the schema and build versions. |

# Settings File

TBD

# Request File

TBD

# Typical Workflow

Below is the sample workflow, where Sirius software is used.

1. The scenario is created in the Scenario Editor (formerly, Network Editor). Initially a scenario consists of just a road network and sensors.
2. The scenario is imported into the Sirius database (import command). The scenarios already in the database can be exported or updated (export, update command).
3. The sensor data that may be used for calibration is loaded into the Sirius database (calibrate command).
4. Calibration for the requested days is performed resulting in new scenarios.
5. At this point, these new scenarios can be further edited in the Scenario Editor.
6. Now we are ready to run simulations, deterministic and stochastic. Simulations can be executed for the scenarios in the database and those specified just by the file. The file option is temporary – to allow us running the simulations until the database schema is finalized (simulate command).
7. The simulation output (raw data) goes either to the database or to the XML file without being recorded in the database. The XML file option for the output is required to allow Sirius be integrated with the third-party Traffic Management Systems (simulate\_output command).
8. After a simulation or a simulation batch concludes, its output (raw) data should be aggregated in 1, 5, 15 minute, 1 hour and 1 day intervals; and the performance measures at the same aggregation levels should be computed (simulate\_process, process commands).
9. Output XML file can be generated with data at different aggregation levels (output command).
10. Reports are generated according to the request specification from data in the database (report command).
11. Data from the database can be dumped into CSV files (list\_\* commands).