

You are in: [Distributed State Machine](#) > [Distributed State Machines](#) > [Dev Guide](#) > Command Line Tools

## Command Line Tools



[Like](#) | Updated May 18 by [David A. Wood](#) | Tags: *None*

The following command line tools are available from the command line after having [installed](#) the release into a DSM\_HOME directory and adding DSM\_HOME/bin to your PATH environment variable.

## Rules Engine

The DSM Rules Engine can be run from the command line and only requires the name space, instance id and rules file to be provide. The usage is as follows:

**Usage:** DSM\_HOME/bin/dsmengine -namespace <app namespace> -instance <engine instance id> <rule file>

- [-namespace](#)<application namespace> - namespace used by coordinating tuple applications using this namespace. Required.
- [-instance](#) <instance name> - name given to this instance of the engine. The instance name is used as an 'address' in the tuples to designate the destination for tuples contained in *Channel* tuples. Required.
- rule file - this is the last argument and specifies the name of the file contain the tuple processing rules. Required.

In addition, the following DSM properties are used (either as system properties or in dsm.properties):

- *receiver.network.port* property. This is the port on which the engine will accept tuple changes.
- *application.registry* property. This sets sets the type of application registry to use.

See the [Configuration section](#) for more details.

Note that -D system properties arguments must be enclosed in double quotes on Windows (i.e. ... "-Dfoo=bar" ...).

# DSM Rules Parser

After creating a [high-level DSM rules](#) file you will want to check it for syntax errors and perhaps a correct translation to the [DSMSQL rule language](#). To run the parser on a file

**Usage:** DSM\_HOME/bin/dsmparse <file>

## Tuple Administration

Sends transport tuple to an application identified by a given name space and the 0th value in the tuple.

**Usage:** tuple insert <namespace> <tablename> <column 0 name>=<value 0> <column 1 name>=<value 1> ...

the column names must match those defined in the target table. The types of values are inferred from the value. The 0th value must be the instance identifier of the tuple application to which the tuple will be delivered.

Example: DSM\_HOME/bin/tuple *insert router path src=1.2.3.4 dest=5.6.7.8* - sends to application with id '1.2.3.4' in the 'router' namespace, a tuple insertion of tuple (1.2.3.4, 5,6.7.8) to the table named 'path'

## Comments

*There are no comments.*