# SimSoup User Manual

Version 1.0 - 20 July 2008

## 1 Introduction

This is the user manual for the SimSoup artificial chemistry simulator. It applies to SimSoup version 0.4.

SimSoup enables a Chemistry to be defined in terms of Molecule Types and the possible Interactions between them. A simulation run involves setting up a number of Molecules of various types in a Reactor, and then allowing Interactions to take place between them. The progress of the simulation can be viewed on various graphical displays.

Although the motivation for development of SimSoup is to enable investigation of 'metabolism first' theories of the Origin of Life, the basic design supports modeling the structure and dynamic behaviour of any network.

This manual focuses on how to use SimSoup. For further details of the model itself and the associated thinking see the following papers available at `http://www.simsoup.info/Publications.html`

- Evolution Without Smart Molecules
- SimSoup: An Artificial Chemistry Model for Investigation of the Evolution of Metabolic Networks.

## 2 Summary of Features

SimSoup version 0.4 includes the features listed below. Items that are new in this release version are highlighted in ***italic***.

- **A Chemistry:** Including Molecule Types and Interaction Types
- **A Reactor:** In which interactions take place between Molecules of the various types

- **Graphical Views of the Chemistry and Reactor:** This includes real time display of the current state of the Reactor and the Chemistry. In each case a network is displayed, along with detailed information in list format
- **Graphical Display of Simulation Statistics:** Statistics showing the behaviour of the simulation over time can be displayed in two formats:-
  - **Data Series Plots:** These show the real-time behaviour of a range of variables that are monitored as the simulation runs. The Data Series to be displayed are selectable from various lists
  - **Manhattan Plot:** This shows the variability in the composition of the material in the Reactor over time
- ***Trackers and Cycle Detection:*** A Tracker is an object that can be attached to a Molecule. As Molecules take part in Interactions, the Trackers are passed from Reactant Molecule to Product Molecule. This enables cycles to be detected
- **Mass Conservation:** Each Molecule Type has a mass, and only Interactions that conserve mass are allowed
- ***Heat and Thermodynamics:*** Reaction rates are based on considerations of energy and thermodynamics. The rate at which the Reactor releases heat can be displayed as a data series plot
- **Scenario Display:** A display of the current scenario and status information on its progress
- **Action Requests:** These are events that can be scheduled to take place at particular times. For example, an Action Request can be setup to cause a Molecule of a particular type to be added to the Reactor
- ***Input Processor:*** This enables a simulation scenario to be defined in a file and loaded to SimSoup. Any errors in the file are detected and reported to the user
- **System Monitor:** A (partly hidden) screen - mainly for diagnostic purposes.

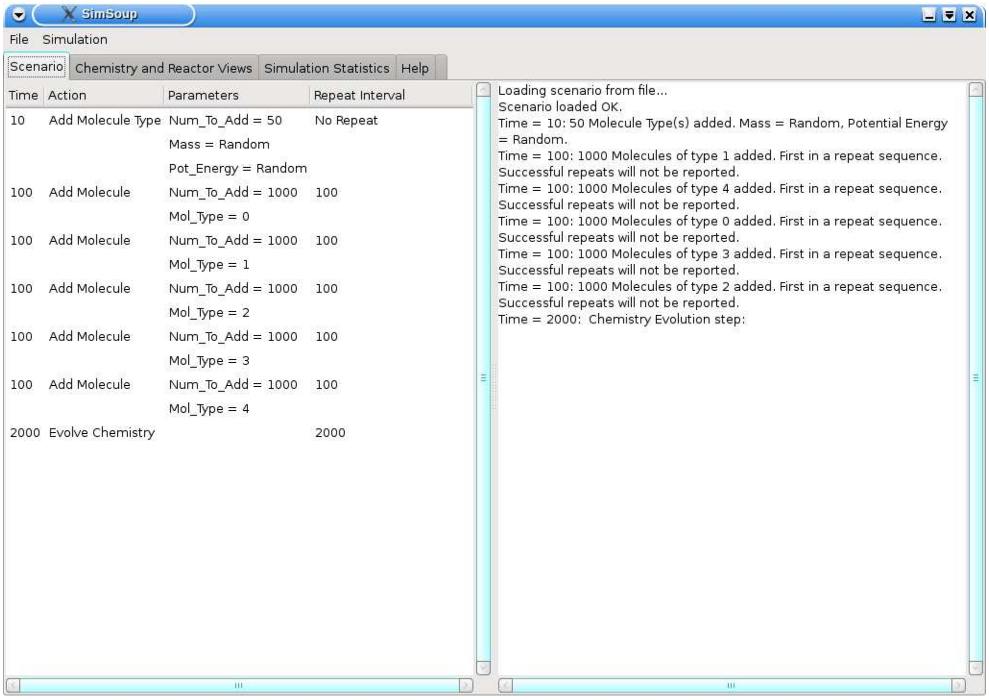
# 3 The User Interface

The SimSoup user interface consists of a simple menu system and a set of tabs (or screens) showing information about the simulation.

## 3.1 The Menu System

The SimSoup menu system has the following options:-

- **File Menu:** Items are as follows:-
  - **Open Scenario:** Load a scenario from an input file. If the scenario loads successfully with no errors, then the scenario details are displayed on the left pane of the Scenario tab (see section 3.2.1)
  - **Close Scenario:** Closes the current scenario. Any statistics collected while running the scenario are deleted
- **Simulation Menu:** Items are as follows:-
  - **Run Simulation:** Starts the simulation
  - **Stop Simulation:** Stops the simulation



| Time | Action | Parameters | Repeat Interval |
|---|---|---|---|
| 10 | Add Molecule Type | Num_To_Add = 50<br>Mass = Random<br>Pot_Energy = Random | No Repeat |
| 100 | Add Molecule | Num_To_Add = 1000<br>Mol_Type = 0 | 100 |
| 100 | Add Molecule | Num_To_Add = 1000<br>Mol_Type = 1 | 100 |
| 100 | Add Molecule | Num_To_Add = 1000<br>Mol_Type = 2 | 100 |
| 100 | Add Molecule | Num_To_Add = 1000<br>Mol_Type = 3 | 100 |
| 100 | Add Molecule | Num_To_Add = 1000<br>Mol_Type = 4 | 100 |
| 2000 | Evolve Chemistry |  | 2000 |

```
Loading scenario from file...
Scenario loaded OK.
Time = 10: 50 Molecule Type(s) added. Mass = Random, Potential Energy
= Random.
Time = 100: 1000 Molecules of type 1 added. First in a repeat sequence.
Successful repeats will not be reported.
Time = 100: 1000 Molecules of type 4 added. First in a repeat sequence.
Successful repeats will not be reported.
Time = 100: 1000 Molecules of type 0 added. First in a repeat sequence.
Successful repeats will not be reported.
Time = 100: 1000 Molecules of type 3 added. First in a repeat sequence.
Successful repeats will not be reported.
Time = 100: 1000 Molecules of type 2 added. First in a repeat sequence.
Successful repeats will not be reported.
Time = 2000:  Chemistry Evolution step:
```

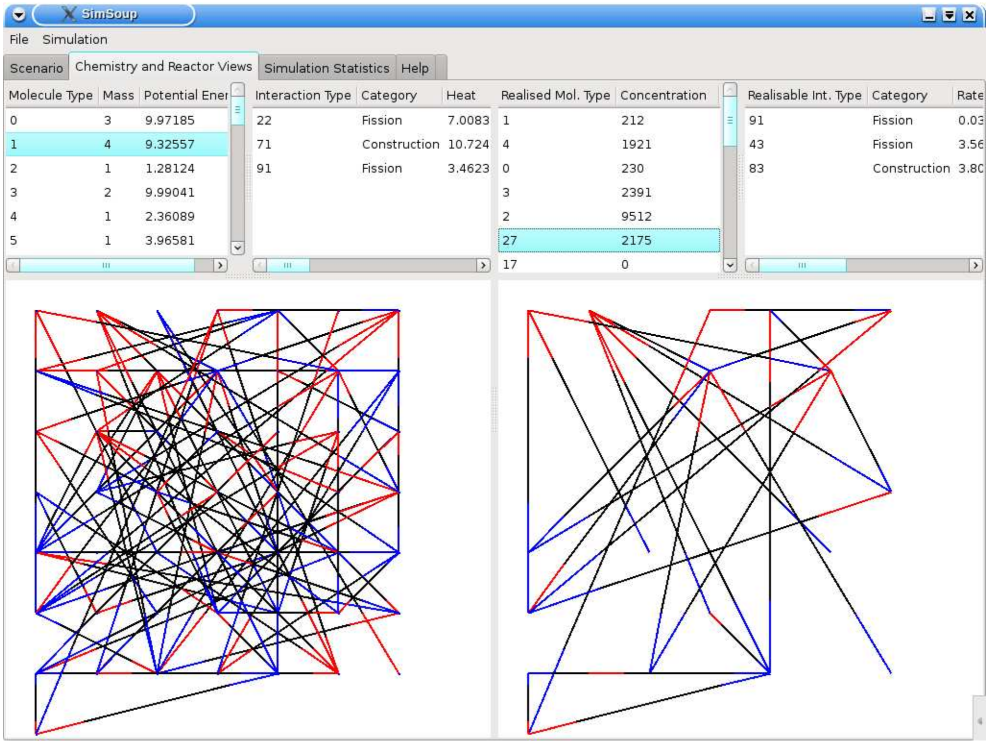
Figure 1: The Scenario Tab

## 3.2 SimSoup Tabs

SimSoup has five tabs (screens). These are described below.

### 3.2.1 The Scenario Tab

Figure 1 shows the Scenario tab. The left hand pane shows the current scenario setup. See section 5.2 regarding the meaning of the information shown. The right hand pane shows output produced by the input processor when loading the current scenario, followed by a record of Action Requests executed while running the scenario.



| Molecule Type | Mass | Potential Ener |
|---|---|---|
| 0 | 3 | 9.97185 |
| 1 | 4 | 9.32557 |
| 2 | 1 | 1.28124 |
| 3 | 2 | 9.99041 |
| 4 | 1 | 2.36089 |
| 5 | 1 | 3.96581 |

| Interaction Type | Category | Heat |
|---|---|---|
| 22 | Fission | 7.0083 |
| 71 | Construction | 10.724 |
| 91 | Fission | 3.4623 |

| Realised Mol. Type | Concentration |
|---|---|
| 1 | 212 |
| 4 | 1921 |
| 0 | 230 |
| 3 | 2391 |
| 2 | 9512 |
| 27 | 2175 |
| 17 | 0 |

| Realisable Int. Type | Category | Rate |
|---|---|---|
| 91 | Fission | 0.03 |
| 43 | Fission | 3.56 |
| 83 | Construction | 3.80 |

Figure 2: The Chemistry and Reactor Views Tab

### 3.2.2 The Chemistry and Reactor Views Tab

Figure 2 shows the Chemistry and Reactor Views tab. The Chemistry View is on the left of the screen, the Reactor View is on the right.

**The Chemistry View:** At the top left are the Molecule Type List and the Interaction Type List. The Molecule Type List shows all the Molecule Types that are defined in the Chemistry. The screenshot shows that Molecule Type 1 is currently selected. Its Mass is 4, and its Potential Energy is 9.32557.

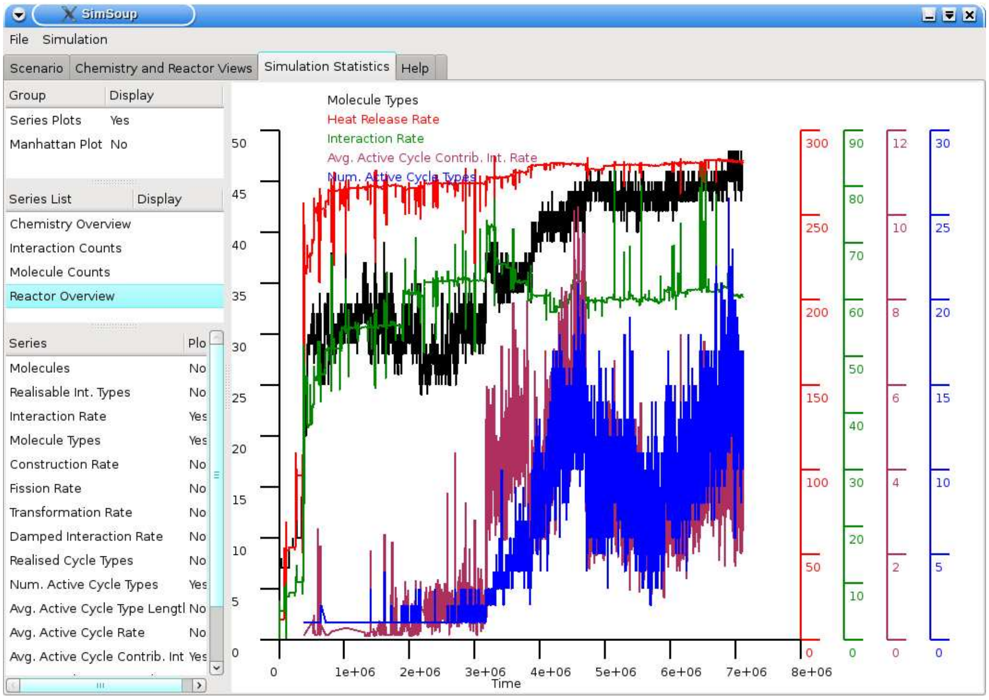
The Interaction Type List on the right of the Molecule Type List shows the Interaction Types that involve the selected Molecule Type as either a Product or a Reactant. Two Fission Interaction Types and one Construction are shown. The first in the list is Interaction Type 22, which is a Fission. Its associated Heat release is 7.0083. By scrolling right, the Forward Activation Energy, Reverse Activation Energy, Activated Complex Energy, Forward Rate Constant, Reverse Rate Constant and Reactants/Products can also be seen.

The Chemistry Network Display at the bottom left part of the screen shows a graphical representation of the network of Interaction Types that are detailed in the Interaction Type List. The red end(s) of an Interaction Type correspond to the Reactant(s). The blue end(s) correspond to the Product(s).

**The Reactor View:** The Reactor Network Display to the right of the Chemistry Network Display shows the Interaction Types that are currently realisable (possible) in the Reactor.

Above the Reactor Network Display are the Realised Molecule Type List and the Realisable Interaction Type List. The Realised Molecule Type List identifies the Molecule Types that have so far been present in the Reactor at any time during the current scenario run. In the screenshot, Realised Molecule Type 27 is selected.

The Realisable Interaction Type List shows the Interaction Types that are currently Realisable (possible) for the selected Realised Molecule Type. The screenshot shows that the selected Molecule Type can be involved in three Interaction Types, including 83, which is a Construction. In addition to the Interaction Type and Category, the current rate of Interactions within the Reactor, and the Reactant(s) and Product(s) for the Interaction Type can be seen by scrolling right.



| Group | Display |
|---|---|
| Series Plots | Yes |
| Manhattan Plot | No |

| Series List | Display |
|---|---|
| Chemistry Overview | |
| Interaction Counts | |
| Molecule Counts | |
| Reactor Overview | |

| Series | Plo |
|---|---|
| Molecules | No |
| Realisable Int. Types | No |
| Interaction Rate | Yes |
| Molecule Types | Yes |
| Construction Rate | No |
| Fission Rate | No |
| Transformation Rate | No |
| Damped Interaction Rate | No |
| Realised Cycle Types | No |
| Num. Active Cycle Types | Yes |
| Avg. Active Cycle Type Lengtl | No |
| Avg. Active Cycle Rate | No |
| Avg. Active Cycle Contrib. Int | Yes |

Figure 3: The Simulation Statistics tab showing Data Series Plots

### 3.2.3 The Simulation Statistics Tab

Statistics can be displayed in two formats: Data Series Plots and the Manhattan Plot.

**Data Series Plots:** Figure 3 shows the Simulation Statistics tab displaying a set of Data Series Plots. The main part of the display shows up to five plots of user selectable simulation variables. In this case, the number of Molecule Types (black) present in the Reactor, the Interaction Rate (green), the average rate of Interactions that are contributing to active Cycles (brown), the Heat Release Rate (red) and the number of Active Cycle Types (blue) are plotted against time.

To the left of the plot area (from top to bottom) are:-

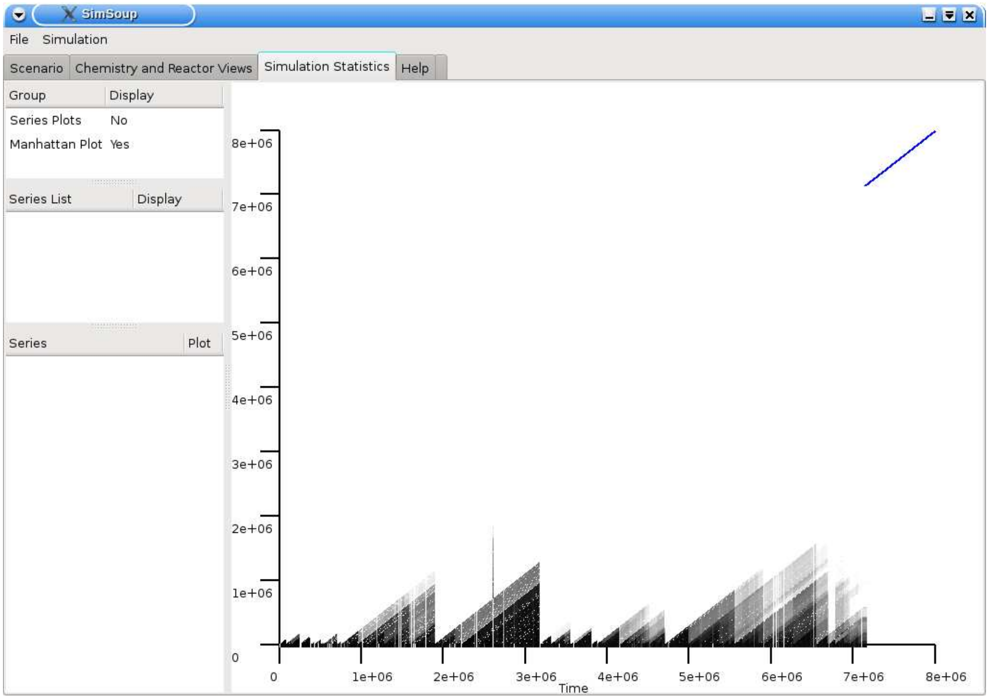
- **The Display Group Area:** This shows the two types of display available. In this case the Series Plots display is selected. To display the Manhattan Plot, double click on 'Manhattan Plot'

- **The Series List Area:** This shows the lists of data series that are available for display:-

  - **Chemistry Overview:** Numbers of various Interaction Types in the Chemistry
  - **Interaction Counts:** Rates of the various Interaction Types in the Reactor
  - **Molecule Counts:** Numbers of Molecules of the various types in the Reactor
  - **Reactor Overview:** Measures of overall Reactor activity.

  To display the various series within a particular series list, click on that list.

- **The Series Area:** This displays the data series in the currently selected series list. To add or remove a series to / from the display, double click on that series. Up to five series can be displayed simultaneously.



| Group | Display |
|---|---|
| Series Plots | No |
| Manhattan Plot | Yes |

Figure 4: The Simulation Statistics tab showing the Manhattan Plot

**The Manhattan Plot:** Figure 4 shows the Simulation Statistics tab displaying a Manhattan Plot. The tone of each point indicates the difference between the Reactor Compositions (in terms of the numbers of Molecules of each type) at time t (x axis) and a time $\Delta t$ (y axis) earlier. Black indicates zero difference. White indicates a difference greater than $\frac{1}{10}D_{max}$, where $D_{max}$ is the maximum possible difference. The dark triangles indicate periods during which the Reactor Composition is roughly constant. The right hand edge of a triangle indicates a time at which the composition changes sharply.

### 3.2.4 The Help Tab

This has the following ‘sub’ tabs:-

- Quick Start
- About
- Licence

### 3.2.5 The System Monitor Tab

This is primarily for diagnostics during SimSoup development, and the tab name is not shown. However, the tab can be accessed (its the little one to the right of the Help tab). If you want to take a peek at the SimSoup internals then go ahead!

# 4 Action Requests

An Action Request is a request for an event to take place at a particular time or times. For example, an Action Request can be setup to cause a Molecule of a particular type to be added to the Reactor. The design of Action Requests is extensible; a common framework has been produced, making it straightforward to add new Action Requests.

The Action requests currently supported by SimSoup are:-

- **Add Molecule Type:** Add a Molecule Type to the Chemistry
- **Add Molecule:** Add a Molecule to the Reactor
- **Add Construction:** Add a Construction (one of the three forms of Interaction Type) to the Chemistry
- **Add Fission:** Add a Fission (the second form of Interaction Type) to the Chemistry
- **Add Transformation:** Add a Transformation (the third form of Interaction Type) to the Chemistry
- **Evolve Chemistry:** See below.

The Evolve Chemistry Action Request entails a little more that the others. It supports an experiment in which the network of Interaction Types in the Chemistry 'evolves'. At each 'mutation step', a Construction or Fission is either added or removed at random (if this is possible within certain feasibility constraints). After a short period, a 'fitness' function is tested to see the 'effectiveness' of the mutation. The fitness function is based on the rate at which the Interactions taking place in the Reactor are releasing heat. This can be thought of as analogous to the rate of metabolism in an organism. If the heat release has reduced, then the mutation is considered to be 'bad', and is reversed; otherwise the mutation is retained.

# 5 The Input Processor

The SimSoup input processor allows the user to specify a scenario in terms of Action Requests. An example scenario input can be found in the file 'SimSoup_Demo_Input' in the directory 'Scenario_Input' in the SimSoup distribution.

## 5.1 Input Syntax

The input file has the following syntax rules:-

- The file includes statements each of which must be terminated with a semicolon
- A statement may span multiple lines
- Each statement consists of:-
  - A keyword identifying the statement type
  - A series of Attribute identifiers followed by one or more Attribute values. Attribute identifiers begin with an @ character
- Comments are allowed. A comment begins with two slash characters (//); subsequent input text up to the end of the line is ignored
- Comments can occur within an input statement.

## 5.2 Input Semantics

The file must also follow semantic rules in relation to statement type keywords, Attribute identifiers, and the number and format of Attribute values.

### 5.2.1 Statement Type Keywords

The allowed statement type keywords and the corresponding Action Requests are:-

- Add_MolType_Req (Add Molecule Type)
- Add_Molecule_Req (Add Molecule)
- Add_Constr_Req (Add Construction)
- Add_Fission_Req (Add Fission)

- Add_Transfm_Req (Add Transformation)
- ChemEvol_Req (Evolve Chemistry)

### 5.2.2 Attribute Keywords, Meanings and Formats

The allowed Attribute keywords and associated meanings and formats are shown below.

- **@ACE**
  - **Meaning:** Specifier for the Activated Complex Energy for an Interaction Type. Activated Complex Energy is the energy that the activated complex formed by the energised Reactant(s) must reach for the reaction to proceed. The specifier can indicate either the actual energy, or that the energy should be selected at random
  - **Format:-**
    - Actual energy: Floating point number
    - Random selection: The text "Random"
- **@Mass**
  - **Meaning:** Mass specifier for a Molecule Type. Can indicate either the actual mass, or that the mass should be selected at random
  - **Format:-**
    - Actual mass: Positive integer
    - Random selection: The text "Random"
- **@MolType**
  - **Meaning:** Molecule Type requirement specifier. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random
  - **Format:-**
    - Actual Molecule Type: Non-negative integer
    - Random selection: The text "Random"
- **@Num_To_Add**

  - **Meaning:** Number of required items to be added. The type of the items depends on the type of Action Request in which the attribute appears
  - **Format:** Positive integer

- **@Pot_Energy**

  - **Meaning:** The potential energy of a Molecule Type. Can indicate either the actual energy, or that the energy should be selected at random
  - **Format:-**
    - Actual Energy: Floating point number
    - Random selection: The text "Random"

- **@Product**

  - **Meaning:** Molecule Type requirement specifier for a (single) Product. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random
  - **Format:** Same as for @MolType

- **@Product1**

  - **Meaning:** Molecule Type requirement specifier for first Product. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random
  - **Format:** Same as for @MolType

- **@Product2**

  - **Meaning:** Molecule Type requirement specifier for second Product. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random
  - **Format:** Same as for @MolType

- **@Reactant**

  - Meaning: Molecule Type requirement specifier for a (single) Reactant. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random

  - Format: Same as for @MolType

- **@Reactant1**

  - **Meaning:** Molecule Type requirement specifier for first Reactant. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random
  - **Format:** Same as for @MolType

- **@Reactant2**

  - **Meaning:** Molecule Type requirement specifier for second Reactant. Can indicate either the actual Molecule Type, or that a Molecule Type is to be selected at random
  - **Format:** Same as for @MolType

- **@Repeat**

  - **Meaning:** Specifies whether the Action Request should be executed once only, or at repeated intervals
  - **Format:-**
    - Once only execution: The text “No”
    - Repeated execution: Positive integer that specifies the repeat interval

- **@Time**

  - Meaning: The time at which the Action Request should be executed
  - Format: Non-negative integer

### 5.2.3 Statements and Their Associated Attributes

Each statement must have Attributes that are appropriate for the statement type. All statements have @Time and @Repeat Attributes. Other Attributes for statements vary by statement type. The order of Attributes within a statement is unimportant. The statements and their required Attributes are as follows:-

- **Add_ChemEvol_Req:-**

  - @Time
  - @Repeat

- **Add_Constr_Req:-**
  - @Time
  - @Num_To_Add
  - @Reactant1
  - @Reactant2
  - @Product
  - @ACE
  - @Repeat

- **Add_Fission_Req:-**
  - @Time
  - @Num_To_Add
  - @Reactant
  - @Product1
  - @Product2
  - @ACE
  - @Repeat

- **Add_Molecule_Req:-**
  - @Time
  - @Num_To_Add
  - @MolType
  - @Repeat

- **Add_MolType_Req:-**
  - @Time
  - @Num_To_Add
  - @Mass
  - @Pot_Energy

  - @Repeat

- **Add_Transfm_Req:-**
  - @Time
  - @Num_To_Add
  - @Reactant
  - @Product
  - @ACE
  - @Repeat

# 6 Command Line Execution and Output

You can run SimSoup from a console command line. For example:-

```
chris@jupiter:>simsoup
```

Standard output produced by SimSoup can then be seen in the console, or redirected to a file. Eg:-

```
chris@jupiter:>simsoup > SimSoup_Output
```

The following categories of output are produced:-

- Output produced during input processing. This is more verbose than the output seen on the Scenario tab. Look at this output if you have difficulty processing an input file

- Output produced by the execution of 'Evolve Chemistry' Action Requests.

There is a third category of output, which ideally you should never see. This output is produced when a run-time check (or *assertion*) performed by SimSoup fails. In such cases, SimSoup writes a line of output indicating the nature of the error and then terminates. If SimSoup terminates unexpectedly, then please send details of the command line output and the input file to the address given on the SimSoup website.