

# Text User Interface of Wildkatze

## Dravvya Simulation and Engineering

Wildkatze solver is console based application, this is one mode of input and interaction with it. The other mode is using the Wildkatze-GUI. These commands are grouped in two major categories. One set of commands drive the solver, for example one can ask solver to iterate or to export certain data etc, another set is where user changes settings of simulation object, phase, feature or of a model.

## 1 Control Commands

### **setdir directory\_path**

Sets directory path to place where simulation tree and mesh data is present. Wildkatze loads data from this directory. For current directory just use ./ as the path.

### **fluent2bmesh mesh\_name**

Fluent2bmesh converts FLUENT mesh file to FVUS file format, mesh\_name is mesh name without .msh extension. To convert mesh from STAR-CCM+ format to FVUS, there exists another program for it.

### **setmesh mesh\_name**

Sets the mesh file name (without .bmesh and .info.bmesh). This is the file which solver loads. Simulation tree has no information about the mesh file name. This is done to allow same simulation file to be run with different meshes with same region names and boundary names.

### **setsim simtree\_name**

Setsim sets simulation tree name that solver shall use. Note that user shall set directory, mesh name and simulation tree before processing any commands. This way solver knows which simulation tree to work with.

### **readsimtree**

Reads the simulation tree and mesh from file names already specified.

### **update-simtree**

Updates the simulation tree from solver models and settings.

### **save-simtree**

Saves simtree that could be loaded and modified by GUI.

### **process-file file\_name**

Process-file runs commands listed in a file. This is to help writing all the common Wildkatze commands at one place. Most tutorials come with a *run.txt* for example:

```
setdir ./
setmesh triVortex
setsim trivortex
readsimtree
init-solution
```

This sets up calculation for iterations.

### **init-solution**

Initializes the solver. This has to be done before iterations can take place.

### **Iterate number\_of\_iterations**

Runs number of iterations or time-steps specified by the user.

*Note:* to pause the solver or to stop solver from iterating, user shall create *wildkatze.pause* file in the folder where simulation is running. Wildkatze checks for this file, before starting next iterations and stops when this file is present. Also make sure to remove *wildkatze.pause* before running iterations again.

### **save-restart restart\_file\_name**

Save-restart exports restart data in *restart\_file\_name*. This consists of object options (for example model settings), field variables etc.

### **read-restart restart\_file\_name**

Read-restart reads data from *restart\_file\_name*. This typically should be done after the *init-solution*, which is responsible for allocating field variables. This over-writes the field variables by these saved values.

### **export-mesh mesh\_file\_name**

Export-mesh saves mesh, in case there is mesh movement and nodes have to be updated. This is used in case of sliding mesh simulations where mesh moves.

### **export-ensight file\_name**

Export-ensight exports post processing data in Ensignt Gold format. Note that for transient data export for animation, use the GUI to set up export at certain frequency.

### **export-residuals**

Command exports residuals

### **export-monitors**

Exports field variables at point probes etc.

## 2 Buffer Pointers

Since there exist many regions, boundaries, setting objects etc, it is not good idea to write their path name again and again. For this reason, Wildkatze has buffer pointers to region, boundary, setting option etc. that could be set once and following commands could be applied to it.

### **region region\_name**

Sets region with region\_name to buffer pointer.

### **rface-list**

Lists the boundary faces for the region at buffer pointer (set by *region region\_name*).

### **fvfaces boundary\_name**

Sets the boundary from region in region buffer to boundary buffer.

## 3 Setting Object

Setting Object is a place where various settings related to model, phase or simulation are stored in one object. This is typically visible on the GUI.

These are places where Setting Object is found:

- Simulation Options: settings related to simulation type (steady/unsteady), number of iterations per time step etc.
- Phase: material properties
- Model Control: model options related to the phases declared in phase sets, plus options for default phase (outside of phases in phase set)
- Boundaries Options: for each model given its region set and phase set, there exists a Setting Object for each boundary. This is the means of providing boundary conditions.
- Model Features: model features come with their own Setting Object where feature control is provided.

Now we will see how these settings are accessed and changed during the simulation through TUI (Text User Interface), where GUI is not available.

### **model-list**

Lists the models. One shall note that each model is identified with its hash tag. Hash tag is unique to each model while solver can have multiple model with same name. For example here *PressureBasedFlowModel-plus-90* is its hashtag. This is the tag user shall use to access this model and its options.

### **phase-list**

Lists phases present.

## options

This command sets the Simulation Options to the current options buffer.

## op-list

Lists the settings present in the options pointed by the buffer pointer. Here is output of last two commands:

```
wildkatze> options
  Setting current options to simulation options.
wildkatze> op-list

simulation-type    transient { steady, transient, transient-fractional }
iteration          Integer 1.00000000000000e+00 1.00000000000000e+00
timeIteration      Integer 1.00000000000000e+00 1.00000000000000e+00
current-inner-iteration Integer 0.00000000000000e+00 0.00000000000000e+00
max-iterations     Integer 1.00000000000000e+00 1.00000000000000e+00
max-iteration-per-timestep Integer 5.00000000000000e+00 5.00000000000000e+00
simulation-time    Real 0.00000000000000e+00 0.00000000000000e+00
time-step-size0    Real 1.00000000000000e-04 1.00000000000000e-04
time-step-size1    Real 1.00000000000000e-04
time-step-size2    Real 1.00000000000000e-03
variable-time-step Integer 0.00000000000000e+00 0.00000000000000e+00
minimum-time-step Real 5.00000000000000e-07 5.00000000000000e-07
maximum-time-step Real 5.00000000000000e-04 5.00000000000000e-04
courant           Real 1.00000000000000e-01 1.00000000000000e-01
using-expert-driver Integer 0.00000000000000e+00 0.00000000000000e+00
fractional-after-iteration Integer 1.00000000000000e+01 1.00000000000000e+01
implicit-step-frequency Integer 1.00000000000000e+01 1.00000000000000e+01
fractional-solver-correctors Integer 2.00000000000000e+00 2.00000000000000e+00

wildkatze> █
```

Figure 1: output of **op-list** command

At this point one can see that these settings are of two types:

- **Enumerated Type:** example `simulation-type` with options from `steady`, `transient` and `transient-fractional` with current selected value of `transient`.
- **Real Type:** example `time-step-size0` for current time step. Also note that Integer and Bool type are stored as Real but converted to integer values when needed (1 is considered true while 0 is considered false).

Now lets see how to change these options on command prompt.

**op-double** setting value

***op-double time-step-size0 1E-5***

Sets the time step size to 1E-5 for the options pointed by buffer pointer.

**op-type** setting type\_value

This command is used to set the enumerated settings option. For example to change the `simulation-type` from `transient` to `steady`, we can use:

***op-type simulation-type steady***

Once we know how to change a setting for Options Object (Settings Object), all we need to do is to set the buffer pointer to the Options Object to whose value we wish to change.

### **phase-material-options phase\_name**

Puts the phase buffer pointer to phase with name of phase\_name. Here is an example of changing material properties of phase-1 we defined earlier:

#### ***phase-material-options phase-1***

Sets options buffer to *phase-1*. Now we can see what settings it has by using *op-list* command:

```
wildkatze> phase-list
Phase list:
  phase-1 .
  default-phase .

wildkatze> phase-material-options phase-1
Setting options for phase-1
wildkatze> op-list

material-type      Constant-Properties { Constant-Properties }
density-value-constant  Real 1.000000000000000e+00 1.000000000000000e+00
viscosity-value-constant  Real 1.538000000000000e-05 1.538000000000000e-05
specific-heat-value-constant  Real 1.006430000000000e+03 1.006430000000000e+03
thermal-conductivity-value-constant  Real 2.420000000000000e-02 2.420000000000000e-02
molecular-weight      Real 2.896600000000000e+01 2.896600000000000e+01

wildkatze> □
```

Figure 2: output of **op-list** command

Now we look at how we can change the options related to models.

### **model-options model\_hashtag**

Sets models (with model\_hashtag) options to options buffer:

```
wildkatze> model-list
Model Set:  model-set-1
-  WallDistanceModel-plus-78 ( phase-set-1 )
Model Set:  model-set-2
-  PressureBasedFlowModel-plus-90 ( phase-set-1 )
-  KEpsilonTurbulenceModel-plus-604 ( phase-set-1 )

wildkatze> model-options PressureBasedFlowModel-plus-90
Setting options for PressureBasedFlowModel-plus-90
wildkatze> □
```

Figure 3: output of **model-options** command

### **model-phase-options model\_hashtag**

Sets models (with model\_hashtag) options from given phase to options buffer.

```
wildkatze> model-list
Model Set: model-set-1
- WallDistanceModel-plus-78 ( phase-set-1 )
Model Set: model-set-2
- PressureBasedFlowModel-plus-90 ( phase-set-1 )
- KEpsilonTurbulenceModel-plus-604 ( phase-set-1 )

wildkatze> model-phase-options PressureBasedFlowModel-plus-90 phase-1
Setting options for PressureBasedFlowModel-plus-90( phase-1 )
wildkatze> op-list

model-type SIMPLE { SIMPLE }
pressure-interpolation standard { standard, bodyforce-weighted }
flux-dissipation delta-pressure { delta-pressure, standard }
flux-dissipation-scaling Real 1.0000000000000e+00 1.0000000000000e+00
density-interpolation first-order-upwind { first-order-upwind, second-order-upwind }
convection-scheme Bounded-Central { FirstOrderUpwind, SecondOrderUpwind, CUBISTA, Bounded-Central }
grad-scheme lsqr-green-gauss { green-gauss, least-square, lsqr-green-gauss }
inner-iterations Integer 1.0000000000000e+00 1.0000000000000e+00
extra-continuity-iterations Integer 0.0000000000000e+00 0.0000000000000e+00
momentum-temporal-extrapolation Integer 0.0000000000000e+00 0.0000000000000e+00
temporal-scheme Three-Time-Level { Euler, Three-Time-Level }
```

Figure 4: output of `model-phase-options` command

### `model-phase-bnd-options model_name phase_name region_name boundary_name`

Sets models options for boundary of a given region for specified phase.

**Note:** `model_name` is used here and not `model.hashtag`.

```
wildkatze> model-phase-bnd-options PressureBasedFlowModel phase-1 fluid inlet
Setting storage for fluid(inlet) for phase phase-1
Setting options for phase-1
wildkatze> op-list
```

Figure 5: output of `model-phase-bnd-options` command