

1 FishMet

FishMet — A simulation model of fish feeding and appetite

Synopsis

fishmet.exe [*OPTIONS*]

DESCRIPTION

FishMet, is a mechanistic, process-based agent-based simulation model of feeding, appetite and growth in fish. It has been engineered from the start to provide extensibility, interoperability and a plugin-like application within a larger digital twin system. FishMet is a discrete time model running over numerous time steps with the resolution of 1 s. The model works at the fine-grained level of individual feed items and individual fish decisions. This will potentially allow for complex simulations with variable feed, complex schedules and stochastic environment.

For more details of the model, see the **FishMet model User Manual**.

OPTIONS

-h, --help, /h or /help
print brief help

-g, --force-gui, -gui, /gui
force the model to start in the GUI mode.

-run, --run-model, /run
force run model at startup with the default parameters from the configuration file and then exit.

-q, --quiet, /q, /quiet
"quiet mode" with minimum output.

-b, --batch, /b, /batch
"batch mode" with no screen output, useful for running script. Note that GUI is always disabled in the batch mode.

ENVIRONMENT VARIABLES

Some global aspects of the program can also be controlled via the system environment variables.

FFA_MODEL_PARAMETER_FILE

Defines the name of the file that keeps [global model parameters](#). If this environment variable is not set, the default name `parameters.cfg` is used.

FFA_MODEL_GUI

if set to 1 or `yes` or `true` will force the model to start in the GUI mode.

FFA_MODEL_OUTPUT_DEST

sets the output destination as [output_dest](#) parameter in the model configuration file. The value set by the environment variable takes priority over the configuration file. Note that the outout destination can also be set using the [output_dest](#) parameter in the configuration file, but this environment variable takes priority.

FFA_MODEL_OUTPUT_TAG

sets the model tag that is added to all automatically generated files.

FFA_MODEL_OUTPUT_TAG_CONFIG_REV

causes to determine the model tag that is added to all automatically generated file names from the `configuration_version` parameter in the main parameter configuration file. See [configuration_version](#) parameter.

GUI_PLOT_AREA_WIDTH

sets the width of the graphical plotting area. The default value is 110. But if you have a very big computer screen, this may be increased. As an approximate orientation, `GUI_PLOT_AREA_WIDTH=150` is adequate for a 75dpi screen 1920x1200i, `GUI_PLOT_AREA_WIDTH=200` is okay for 2560x1440.

CONFIGURATION FILE

The configuration file is a text file with various model options that are set as `option_name = value` format. Any line starting from `#` is a comment and is ignored. This file is mandatory, the model program will not work without it.

The standard name of the configuration file is `parameters.cfg`

Program options**interface_graphical**

default program interface type, graphical `true` or command line `false`.

output_dest

Output destination (usually directory): all plot and output data files will be saved there. The default value is if this variable is absent or directory is not writeable is the current directory. This parameter affects the program behaviour only in the command line user interface and not in the graphical user interface. Note that fish object data save `fish_object_data` is saved exactly into the `fish_object_file` because this is non-human readable internal binary data not intended as the model output. The destination can refer to destination directory or output file name prefix. In the former case it should terminate with the operating system specific directory delimiter (e.g. `/` on GNU/Linux or `\` on Microsoft Windows). If the value of `output_dest` does not terminate this way, it will refer to the file prefix. It is important that the output destination can also be set with the [FFA_MODEL_OUTPUT_DEST](#) environment variable, which takes priority over this configuration file parameter. The following outputs are saved to the `output_dest`:

- output plots produced by `save command`;
- output arrays by `save model_output`;
- rate arrays by `save rate_output`;
- output statistics by `output stats` and `append stats`.

time_plots_unit_scale

Scale units for the time-based plots 1=sec, 2=min, 3=hours.

rate_unit_scale

Default time unit scale for calculating rates. Rate unit is set as digit: 1=s-1, 2=min-1, 3=h-1.

plot_scale_relative

define whether to use the raw absolute or relative for some of the time-based plots, e.g. stomach and midgut mass dynamics. `true` means relative scales, i.e. stomach or midgut food mass plot displays filling relative to the total capacity, `false` means that absolute food mass is used.

rate_interval

default discretization interval in minutes that is used to calculate rate, e.g. this interval is used to plot the ingestion rate

gui_plot_window_page_size

default page size (height, width) for the GUI plot window in pixels.

gui_plot_area_width

default width of the plot output area in pixels.

gui_plot_area_aspect

default aspect ratio (height/width) of the plot output area.

configuration_version

defines the configuration version tag that is added to all automatically generated output files. The tag should be defined using Subversion "\$Revision: 1234 \$" keyword. It is then parsed to exclude everything except the revision number, adding "conf_r" prefix. (so configuration_version = "\$Revision: 1234 \$" translates to "conf_r1234". Note that this behaviour is enabled if the environment variable FFA_MODEL_OUTPUT_TAG_CONFIG_REV is set to any value and disabled otherwise.

food_provision_file_name

This variable defines the file name that contains the food provision scheduling. This file should be in CSV format or a single-column plain text. In a multi-column CSV file, the last column is assumed to contain the feed scheduling. Scheduling is encoded by **zero** and **non-zero** values by minute (or by second if [food_provisioning_file_by_s](#) is set to TRUE). Here any non-zero value means that food is provided during the respective minute interval, zero means that food is not provided. The file is intended to describe the temporal pattern for 24 h, any longer sequence is discarded. All missing values used for filling the remaining minutes to complete the 24 h are zeroes, i.e. food not provided. The pattern is then propagated for all 24 h periods if [food_provision_file_repeat](#) is TRUE or applied once if [food_provision_file_repeat](#) is FALSE. Note that the file takes priority over the food ononing pattern defined in [food_provision_pattern](#). **Example:** food_provision_file_name = "../file_name.csv"

food_provision_file_repeat

A logical flag that defines if the feed scheduling pattern defined by the [food_provision_file_name](#) file will propagate to all 24 h periods or applied just once to the first such period. The default value is TRUE.

Example: food_provision_file_repeat = TRUE

food_provisioning_file_by_s

Global logical flag that defines if the feed scheduling pattern defined by the [food_provision_file_name](#) file represents the data for each time step (s) (if TRUE) or by minute (FALSE). The default value is FALSE i.e. data are given for each minute.

Example: food_provisioning_file_by_s = FALSE

stats_output_file

The name of the CSV file to save general output statistics for specific run of the model.

The default name is "fishmet_stats.csv"

Example: stats_output_file = "data.csv".

stats_output_long

This is a logical parameter that defines if the output stats are saved using the long or short format. In the first case, both the input parameters and the output stats are saved/appended into the CSV file. Note that the default value is TRUE.

Example: stats_output_long = TRUE

stats_output_prediction_stamp

This parameter define if the prediction timestamp is produced in the output stats file. This timestamp then takes the model execution timestamp and adds the time equivalent to the [run_model_hours](#) simulation duration. The default value is FALSE. All timestamps are produced in the following format: 2023/07/21 03:09:50.

Example: stats_output_prediction_stamp = TRUE

fish_object_file

The name of the binary file to save the fish object data, i.e. all data for the model fish allowing to resume simulation from a specific fish state (stomach and midgut content, appetite level etc). The default file name is fishmet_fish_state.bin. Note that the file name must have the .bin file extension, otherwise it is not accepted.

Example: fish_object_file = "persistent_data_1.bin"

stomach_emptying_matrix

The name of the CSV data file that defines the stomach emptying parameter matrix, an interpolation grid matrix that defines how long does it take to process full stomach calacity of the feed until no feed remains in the stomach. The matrix defined in this file determines stomach emptying time for a range of fish body mass (columns) and ambient temperatures (rows).

An example of the stomach emptying matrix is provided below:

,	50,	100,	300,	500
5,	24,	35,	75,	100
10,	15,	20,	50,	75
15,	9,	15,	40,	50
20,	5,	10,	30,	35
22,	4,	8,	29,	33

Here the rows define ambient temperatures (5,10,15,20,22 C) and columns refer to the fish body mass (50, 100, 300, 500 g).

Model parameters

The parameters of the model described below can be defined in the [parameter file](#) or set during the runtime using the command line or graphical user interface. They can also be defined in a batch script file.

run_model_hours

Total number of hours for which the simulation is run. It translates into the raw time steps internally as $T = H \times 3600$, where T is the number of time steps, H is the `run_model_hours` parameter and 3600 is seconds per hour. **Example:** `run_model_hours = 24`.

daytime_hours

Default daytime duration, hours. The night time duration is defined as $24 - \text{daytime_hours}$. Note that feeding activity occurs only during the day time and not at night. **Example:** `daytime_hours = 12`.

day_starts_hour

Default hour at which the daytime is normally started. It is also the initial offset for day at the start of the simulation. **Example:** `day_starts_hour = 6`.

temperature

Ambient temperature °C. **Example:** `temperature = 14`.

body_mass

Fish body mass at the start of the simulation, g. **Example:** `body_mass = 10.0`.

body_mass_override

Override [body_mass](#) value from fish object file (see [fish_object_file](#)) with input parameter value.

baseline_activity_day

Baseline locomotor activity, swimming speed SL/s, during the day as defined by [daytime_hours](#) parameter. + **Example:** `baseline_activity_day = 2.5`.

baseline_activity_night

Baseline locomotor activity, swimming speed, SL/s, during the night as defined by [daytime_hours](#) parameter. + **Example:** `baseline_activity_night = 0.5`.

stomach_capacity

fish **stomach mass**, max. filling capacity, g. **Example:** `stomach_capacity = 5.0`.

midgut_capacity

fish **midgut mass**, max. filling capacity, g. **Example:** `midgut_capacity = 10.0`.

stomach_midgut_automatic

logical flag determining, when TRUE, that [stomach_capacity](#) and [midgut_capacity](#) are continuously automatically recalculated based on the body mass [body_mass](#). Note that this can override any fixed parameter values set by the parameter values [stomach_capacity](#) and [midgut_capacity](#).

Example: `stomach_midgut_automatic = True`

absorption_ratio

Absorption ratio in the midgut, relative to the initial mass of food. **Example:** `absorption_ratio = 0.7`.

ingestion_delay

Delay after ingestion, min. Water uptake occurs during the delay. **Example:** `ingestion_delay = 30`.

water_uptake

Proportion of water uptake, relative to the initial food item mass. **Example:** `water_uptake = 0.20`.

water_uptake_a

Water uptake pattern over time is defined by the logistic equation $\frac{c}{1+ae^{-r^2t}}$, where c is the upper limit of the mass, a and r are logistic parameters and e is natural logarithm base. The a and r parameters of the equation refer to the `water_uptake_a` and `water_uptake_r` parameters. See also [water_uptake_r](#). **Example:** `water_uptake_a = 200.0`.

water_uptake_r

Water uptake pattern over time is defined by the logistic equation $\frac{c}{1+ae^{-r^2t}}$, where c is the upper limit of the mass, a and r are logistic parameters and e is natural logarithm base. The a and r parameters of the equation refer to the `water_uptake_a` and `water_uptake_r` parameters. See also [water_uptake_a](#). **Example:** `water_uptake_r = 0.01`.

digestion_delay

Delay of digestion, min. This refers to the time interval from the moment the food item gets into the mid-gut until absorption of the food starts. **Example:** `digestion_delay = 20`.

midgut_maxdur

The maximum duration a food item can be processed in the fish midgut, min. If it stays in the mid-gut for any longer time, it is evacuated. **Example:** `midgut_maxdur = 180`.

transport_pattern_t

Grid array defining the food transport pattern in the stomach. This array defines the time grid for interpolation. Note that normally it takes several hours until the food item fully disappears from stomach (gastric emptying time). This parameter can be set either in hours (small numbers) or in raw seconds (large numbers) and the correct time unit is normally autodetected by the program. See also [transport_pattern_r](#).

Example (s): `transport_pattern_t = 0 3000 15000 21600`

Example (hours): `transport_pattern_t = 0, 0.8, 4, 6`

transport_pattern_r

Grid array defining the food transport pattern in the stomach. This array defines the proportion of food mass left in stomach at each level of `transport_pattern_t` array. See also [transport_pattern_t](#).

Example: `transport_pattern_r = 1.0 0.6 0.01 0.0`.

appetite_night

Fish appetite at night. This value is normally low because the fish do not feed at night. If the value is absent or negative, fish appetite does not reduce at night.

Example: `appetite_night = 0.1`

appetite_factor_a

Appetite factor is defined by the logistic equation: $\frac{c}{1+ae^{-r^2t}}$. This parameter refers to a . See also [appetite_factor_r](#).

Example: `appetite_factor_a = 50000.0`.

appetite_factor_r

Appetite factor is defined by the logistic equation: $\frac{c}{1+ae^{-r^2t}}$. This parameter refers to r . See also [appetite_factor_a](#).

Example: `appetite_factor_r = 20.0`.

appetite_threshold_stomach

Protective appetite threshold for stomach: this is the maximum value of the stomach appetite signal when the overall fish appetite level depends only on stomach filling. Setting a sufficiently low value provides a way to protect stomach from overfilling.

Example: `appetite_threshold_stomach = 0.2`

appetite_energy_rate

Logistic appetite parameter describing defining the energy appetite component dependence on the energy budget.

Example: `appetite_energy_rate = 40.0`

appetite_energy_shift

Logistic appetite parameter describing defining the energy appetite component dependence on the energy budget.

Example: `appetite_energy_shift = 0.2`

activity_appetite_factor

Activity appetite factor determining how fish locomotor activity increases with increasing appetite.

Example: `activity_appetite_factor = 0.5`

stress

This is an array of time points that define intervention of stress. The array can be defined either in raw *s* time scale or in minutes depending on `stress_is_min` parameter. **Example:** `stress = c(1200, 6000)`.

stress_metabolic_cost

The maximum value of the metabolic cost of stress in units of resting metabolic rate (SMR). The actual metabolic cost at each time step scales depending on the time since stress intervention defined by `stress` array, following the stress pattern described by the `stress_grid_hour` and `stress_grid_fact` arrays.

Example: `stress_metabolic_cost = 0.5`

stress_is_min

Defines if stress intervention timing is defined in minutes (TRUE) or raw *s* (FALSE)).

Example: `stress_is_min = TRUE`

stress_grid_hour

Stress effect interpolation grid array. `stress_grid_hour` defines the time since the stress intervention occurs. The second array, `stress_grid_fact`, defines the intensity of the stress effect.

Example: `stress_grid_hour = c(0, 10, 20, 40, 50, 72)`

stress_grid_fact

Stress effect interpolation grid array. `stress_grid_fact` defines the intensity of the stress effect. For example, if it is equal to 1.0, appetite is completely suppressed. The other array, `stress_grid_hour` defines the time scale (h) for each point.

Example: `stress_grid_fact = c(1.0, 0.9, 0.7, 0.2, 0.1, 0.0)`

midgut_michaelis_r_max

Michaelis-Menten food absorption parameter in midgut, *r_max*, relative to the mass of the food item. Rate is per second, the basic discrete step of the model. E.g. 0.8 means that 80% of the food item mass can be absorbed at maximum. See also `midgut_michaelis_k`. **Example:** `midgut_michaelis_r_max = 0.9`.

midgut_michaelis_k

Michaelis-Menten food absorption parameter in midgut, *K_M*, relative to the total midgut capacity (e.g. 0.25 means a quarter of the total midgut mass). See also `midgut_michaelis_k`.

Example: `midgut_michaelis_k = 0.25`.

midgut_temp_fact_t

Temperature adjustment for absorption is controlled by the two parameters that define the temperature adjustment grid factor for the Michaelis-Menten equation: `midgut_temp_fact_t` and `midgut_temp_fact_m`. This parameter is the temperature grid for Michaelis-Menten absorption process.

Example: `midgut_temp_fact_t = c(5.0, 10.0, 15.0, 25.0)`.

midgut_temp_fact_m

Temperature adjustment for absorption is controlled by the two parameters that define the temperature adjustment grid factor for the Michaelis-Menten equation: `midgut_temp_fact_t` and `midgut_temp_fact_m`. This parameter is the temperature grid for Michaelis-Menten absorption process. This is the speedup factor grid for the Michaelis-Menten absorption process for the temperatures defined by `midgut_temp_fact_t`.

Example: `midgut_temp_fact_m = c(0.4, 0.6, 1.0, 8.0)`

smr_oxygen_temp

Interpolation X axis grid defining the SMR function on the temperature °C.

smr_oxygen_o2

Interpolation grid Y axis defining the SMR function on the temperature. SMR unit is $mg\ O_2\ kg^{-1}\ h^{-1}$.

branchial_energy_factor

A parameter defining branchial and urine (ZE+6UE) energy consumption as a factor to SMR, e.g. 0.3 means $(UE+ZE) = 0.3 \cdot SMR$. This is an alternative to [branchial_ammonia_rate](#). If both [branchial_energy_factor](#) and [branchial_ammonia_rate](#) are defined, the former parameter always takes priority.

Example: `branchial_energy_factor = 0.1`

branchial_ammonia_rate

A parameter defining branchial and urine (ZE+6UE) energy consumption as a fixed ammonia excretion rate, micro-mol per g body mass per hour. This is an alternative to [branchial_energy_factor](#). If both [branchial_energy_factor](#) and [branchial_ammonia_rate](#) are defined, the former parameter always takes priority.

Example: `branchial_ammonia_rate = 0.25`

sda_absorption_rate_max

Defines the specific dynamic action (SDA) that depends on the absorption rate. There are two parameters that define the linear relationship between absorption rate and SDA increase. This parameter defines the maximum absorption rate when the maximum SDA (defined by [sda_energy_factor_max](#)) is reached.

Example: `sda_absorption_rate_max = 0.0007`

sda_energy_factor_max

Defines the specific dynamic action (SDA) that depends on the absorption rate. There are two parameters that define the linear relationship between absorption rate and SDA increase. This parameter defines the maximum SDA reached at the maximum absorption rate defined by [sda_absorption_rate_max](#). For example, this parameter equal to 2.0 means that $SDA = SMR \cdot 2.0$ at the maximum absorption rate.

Example: `sda_energy_factor_max = 2.0`

food_item_mass

Basic dry mass of one food item (g). **Example:** `food_item_mass = 0.25`.

feed_gross_energy

Gross energy content of the feed, MJ/kg (= kJ/g) **Example:** `feed_gross_energy = 23.0`.

food_input_rate

Rate of food item input, per min. **Example:** `food_input_rate = 6.0`.

food_provision_pattern

Food provision patterning is determined by an array defining the time intervals (min) when food is provisioned and not provisioned e.g. a value of 10 230 means 10 min food is given followed by 230 min not given. Note also that any arbitrary food scheduling can be obtained by encoding periods of food provisioning and not provisioning in the [food_provision_file_name](#) parameter. **Example:** `food_provision_pattern = 10 230`.

feed_start_offset

The offset (delay, min) to start feeding at the beginning of the 24 hour period. The feeding pattern defined by [food_provision_pattern](#) starts every 24h period after this offset. For example, if the offset is set to 180 min, this means that the first feeding session defined by the [food_provision_pattern](#) begins 3h (180 min) after the start of the day.

Example: `feed_start_offset = 180`

stomach_emptying_matrix

The stomach emptying parameter matrix, an interpolation grid matrix that defines how long does it take to process full stomach capacity of the feed until no feed remains in the stomach. The matrix is defined in a file set by [stomach_emptying_matrix](#).

USER INTERFACE

The program has two interface modes: [command line \(CMD\)](#) and [graphical \(GUI\)](#). Default parameters of the model and the program are obtained from the configuration file `parameters.dat`.

COMMAND LINE INTERFACE AND SCRIPTING

By default it starts in the command line (terminal window) interface. This can be changed by the `interface_graphical` option in the configuration file (set to `true`).

Basic commands

The command line user interface involves typing commands in the program shell. Alternatively, a set of commands can be provided in a batch script file.

help

Get brief help on the commands. Several subcommands can be used: `parameters`, `plot`.

run

Run the model with the default parameters. Alternative alias: `start`.

reset

Reset all model parameters to the default values from the configuration file. Alternative alias: `restart`.

quit

exit the program.

set

Set any model parameter defined in [Model parameters](#). **Example:** `set run_model_hours=24`. Another use of the `set` command is to define the output format for saving plots (see [save](#) command): `set format` or `set save_format`, then define one of the possible output formats: PDF, PNG, SVG, BMP, GIF, EPS, WMF, CGM. This command is also used to define rate interval: `set rate_interval` as well as to define [override flags](#).

plot

Produce parameter plots or model output plots. To save a plot to a disk file, use the related [save](#) command. Rate plots can have different rate interval using `set rate_interval` command. **Example:** `plot stomach_transport`.

Subcommands:

- `stomach_transport` - Stomach transport curve;
- `absorption_mm` - Michaelis-Menten absorption;
- `appetite_factor` - Appetite factor;
- `smr_function` - SMR pattern function;
- `total_ingested` - Cumulative N food ingested;
- `mass_ingested` - Food mass ingested, g;
- `ingestion_rate` - Food ingestion rate;
- `absorption_rate` - Absorption rate;
- `not_ingested` - Cumulative N food not ingested;
- `encountered` - Cumulative N food encountered;
- `appetite` - Overall appetite level;
- `total_stomach` - Total mass of food in stomach;
- `total_midgut` - Total mass of food in midgut;
- `absorption` - Total cumulative absorption;
- `energy_balance` - Energy balance, kJ;
- `oxygen_uptake` - oxygen uptake, mg O₂ per kg per hour
- `body_mass` - Body mass, g;
- `growth_rate` - Growth rate, g/min;
- `sgr` - Specific growth rate, g/min;
- `activity` - Locomotor activity, SL/s;
- `evacuation` - Cumulative evacuation.

save

Save a plot defined as in the [plot](#) subcommand. This command also requires two additional parameters: plot type as in [plot](#) and the file name. **Example:** `save stomach_transport plot_01.pdf`.

Another use of this command is for saving model output data (CSV formatted table):

- `model_output`, `output_arrays` or `output_data` - model output arrays;
- `rate_output`, `output_rate`, `rate_data` - model rate data (using default rate interval, see `set rate_interval` [command](#) to change it).
- `fish_object_data` or `fish` - save full fish object data allowing to resume simulation using the [load](#) command. Note that the file name for the fish object is defined by [fish_object_file](#).

load

- `fish_object_data` or `fish` - load full fish object data allowing to resume simulation. Note that the file name for the fish object is defined by [fish_object_file](#).

show

This command is used to print specific model parameters.

- `show params` or `params`: print all parameters of the model;
- `show stomach_transport` with optional `raw` or `adjust`: show stomach transport unadjusted or adjusted to temperature and fish mass;
- `show stomach_emptying`: display the current stomach emptying matrix as obtained from the [stomach_emptying_matrix](#);
- `show version`: print the program version;
- `show timesteps`: print the total number of timesteps;
- `show stomach_transport raw`, `show stomach_transport adjust`: display the unadjusted and adjusted stomach transport grid arrays: [transport_pattern_t](#) and [transport_pattern_r](#);
- `show save_format` or `show format`: print the graphic output format (PDF, PNG, SVG, BMP, GIF, EPS, WMF, CGM, CSV).
- `show plot_scale_relative`: print if the time plots show raw data or relative values (e.g. stomach filling relative to the total stomach capacity).
- `show statistics` or `show stats`: show general model output statistics.

adjust stomach_transport

Adjust the stomach transport parameter arrays for the [body_mass](#) and ambient [temperature](#). See also [stomach emptying matrix](#).

output statistics

Initiate output of the general model statistics into [stats_output_file](#) csv output data file. Note that this results in writing the column headers and the first row of the data. All subsequent data output should use the [append statistics](#) command.

Warning: The `output` command will **overwrite any existing CSV file**, be careful!

append statistics

Append general statistics data for the current run of the model. Note that the first output should use the [output statistics](#) command.

shell

Execute some operating system specific command. This command can be useful, for example, for checking files from within the model program, moving files, deleting temporary files etc. Example: `shell dir /w` (this will show all files in the working directory on Windows without size, timestamp and other details).

Override flags:

The [load](#) command that loads the fish object data to resume simulation works such that when the simulation is resumed, fish parameters (e.g. [body_mass](#)) are obtained from the fish object data (see [fish_object_file](#)) rather than the model parameters, either defined in the parameter file or [set](#) command. This ensures smooth continuous simulation over many "resumes." However, there is a method to override the fish object data parameters with those explicitly defined by [set](#) or parameter file. It is done by setting the [body_mass_override](#) flag to `TRUE` with [set](#) command.

Example: `"set body_mass_override = True"`: the value of the `body_mass` will be obtained from input parameter rather than fish object file.

Scripting

Many model runs can be performed by a script that contains a sequence of commands. Any line starting with # is considered a comment and is ignored.

Here is an example of a script:

```
set run_model_hours = 100

# ---
set food_item_mass=0.05
set food_input_rate=30
run
save ingested zz_ingested_005.pdf
save appetite zz_appetite_005.pdf
save output_data zz_ingested_005.csv

# ---
set food_item_mass=0.1
set food_input_rate=15
run
save ingested zz_ingested_010.pdf
save appetite zz_appetite_010.pdf
save output_data zz_ingested_010.csv
```

To run the script issue this command on the terminal:

```
fishmet.exe < script_file.cmd
```

GRAPHICAL USER INTERFACE

The graphical interface uses the menus and buttons to control the program.

To start the mode in the GUI mode, use this command: `fishmet.exe -gui`

Note

There is a Windows VBS script `fishmet.vbs` to force the model in the GUI mode suppressing the terminal window completely.

EXIT STATUS

- 0 Success
- 1 Failure or error

BUGS

Todo

AUTHOR

Sergey Budaev <sergey.budaev@uib.no>

RESOURCES

Main web site: <http://fishmet.uib.no/>

COPYING

Todo

2 Index

–
--batch, 1
--gui, 1
--help, 1
--quiet, 1
--run-model, 1
-b, 1
-g, 1
-h, 1
-q, 1

A
absorption_ratio, 4
activity_appetite_factor, 6
adjust, 9
append
 statistics, 9
appetite_energy_rate, 5
appetite_energy_shift, 6
appetite_factor_a, 5
appetite_factor_r, 5
appetite_night, 5
appetite_threshold_stomach, 5

B
baseline_activity_day, 4
baseline_activity_night, 4
batch mode, 1
body_mass, 4
body_mass_override, 4
branchial_ammonia_rate, 7
branchial_energy_factor, 7

C
command
 adjust, 9
 help, 8
 load, 9
 plot, 8
 quit, 8
 reset, 8
 restart, 8
 run, 8
 save, 9
 set, 8
 shell, 9
 show, 9
 start, 8
command line options
 --batch, 1
 --gui, 1
 --help, 1
 --quiet, 1
 --run-model, 1

 -b, 1
 -g, 1
 -h, 1
 -q, 1
configuration_version, 3

D
day_starts_hour, 4
daytime_hours, 4
destination
 output, 1, 2
digestion_delay, 5

E
environment variables
 FFA_MODEL_GUI, 1
 FFA_MODEL_OUTPUT_DEST, 1
 FFA_MODEL_OUTPUT_TAG, 1
 FFA_MODEL_OUTPUT_TAG_CONFIG_REV, 2
 FFA_MODEL_PARAMETER_FILE, 1
 GUI_PLOT_AREA_WIDTH, 2

F
feed_gross_energy, 7
feed_start_offset, 7
FFA_MODEL_GUI, 1
FFA_MODEL_OUTPUT_DEST, 1
FFA_MODEL_OUTPUT_TAG, 1
FFA_MODEL_OUTPUT_TAG_CONFIG_REV, 2
FFA_MODEL_PARAMETER_FILE, 1
fish object, 9
 restart simulation, 3, 9
fish_object_file, 3
food_input_rate, 7
food_item_mass, 7
food_provision_file_name, 3
food_provision_file_repeat, 3
food_provision_pattern, 7
food_provisioning_file_by_s, 3

G
gastric emptying, 5
gui_plot_area_aspect, 3
GUI_PLOT_AREA_WIDTH, 2
gui_plot_area_width, 2
gui_plot_window_page_size, 2

H
help, 8

I
ingestion_delay, 5
interface_graphical, 2

L

load, [9](#)

M

midgut_capacity, [4](#)
midgut_maxdur, [5](#)
midgut_michaelis_k, [6](#)
midgut_michaelis_r_max, [6](#)
midgut_temp_fact_m, [6](#)
midgut_temp_fact_t, [6](#)

O

output, [1](#), [2](#)
 statistics, [9](#)
output_dest, [2](#)
override flag, [9](#)

P

plot, [8](#)
plot_scale_relative, [2](#)

Q

quiet mode, [1](#)
quit, [8](#)

R

rate_interval, [2](#)
rate_unit_scale, [2](#)
reset, [8](#)
restart, [8](#)
restart simulation, [3](#), [9](#)
run, [8](#)
run_model_hours, [4](#)

S

save, [9](#)
sda_absorption_rate_max, [7](#)
sda_energy_factor_max, [7](#)
set, [8](#)
shell, [9](#)
show, [9](#)
smr_oxygen_o2, [6](#)
smr_oxygen_temp, [6](#)
start, [8](#)
statistics, [9](#)
stats_output_file, [3](#)
stats_output_long, [3](#)
stats_output_prediction_stamp, [3](#)
stomach emptying, [7](#)
stomach emptying matrix, [3](#)
stomach_capacity, [4](#)
stomach_emptying_matrix, [3](#), [7](#)
stomach_midgut_automatic, [4](#)
stress, [6](#)
stress_grid_fact, [6](#)
stress_grid_hour, [6](#)
stress_is_min, [6](#)
stress_metabolic_cost, [6](#)

T

temperature, [4](#)
time_plots_unit_scale, [2](#)
timestamp, [3](#)
transport_pattern_r, [5](#)
transport_pattern_t, [5](#)

V

verbose mode, [1](#)

W

water_uptake, [5](#)
water_uptake_a, [5](#)
water_uptake_r, [5](#)