# Overview

A MIST/C template has been provided, stripped of everything except the core structure.

When various components are injected into this template, the code (effectively just the model) can be dropped directly into the MIST framework.

This template contains various C-style comments with a special syntax that allows the translator tool conform to MIST structure.

The syntax is ‘<’ TAG\_NAME ‘>’ for tag opening and ‘</’ TAG\_NAME ‘>’ for tag closing. Tags with this structure should be present both in the MATLAB model comments and in the MIST template comments because they will be used by the translator tool.

General translation rules:

1. Code not belonging to any tag will be ignored.
2. Nested tags are not supported.
3. Rules not explicitly stated in the following table are not supported.

Below you can find the formal specification for each tag.

# Tags formal specification

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ALL\_OTHER\_PARAMETERS | All the left-hand sides of assignment under this tag will be copied under the corresponding MIST section, comma separated.  If they are multiple-valued in MATLAB a progressive index will be created and appended to the base name in MIST.  The size of the multi-valued field should be computable within the same line. Accepted multi-valued fields:   1. [1,2,3,4] matrix definition within the same line 2. <VARIABLE\_NAME>(1:4) matrix slicing with evident index values (the indexes should be numeric values)  |  |  | | --- | --- | | MATLAB | MIST | | % <ALL\_OTHER\_PARAMETERS>  parameters = …  computed\_parameters = …  Nav = …  PK\_ETA = param(1:3) | //<ALL\_OTHER\_PARAMETERS>  parameters,  computed\_parameters,  Nav  PK\_ETA\_01,  PK\_ETA\_02,  PK\_ETA\_03  //</ALL\_OTHER\_PARAMETERS> | |
| SPECIES\_NAMES | All the left-hand sides of assignment under this tag will be copied under the corresponding MIST section, comma separated.  If they are multiple-valued in MATLAB a progressive index will be created and appended to the base name in MIST   |  |  | | --- | --- | | MATLAB | MIST | | % <SPECIES\_NAMES>  n\_state\_variables = 82;  index\_PK\_AUC = 1;  params = […]  % </SPECIES\_NAMES> | // <SPECIES\_NAMES>  n\_state\_variables,  index\_PK\_AUC,  params\_01,  params\_02,  …  params\_N  // </SPECIES\_NAMES> | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FLUXES  FLUXES\_SECTION:name1  FLUXES\_SECTION:name2  …  FLUXES\_SECTION:nameN | Each FLUXES\_SECTION found in MATLAB will be translated between FLUXES tag in MIST ordered by name lexicographycally.  Translation rules for code found in FLUXES\_SECTION:   1. Each variable defined in MATLAB should be defined   in-place in MIST   1. Each variable used but not defined:    1. If belongs to SPECIES\_NAMES should be translated as Ith(y, <VARIABLE\_NAME>)    2. Otherwise will be translated as PM(<VARIABLE\_NAME>)   If the variable is accessed with a single index then the MIST variable will be <VARIABLE\_NAME>\_<ACCESS\_INDEX>  and the ending translation will be  PM(<VARIABLE\_NAME>\_<ACCESS\_INDEX>)   |  |  | | --- | --- | | MATLAB  (assuming PK\_A2 is inside SPECIES\_NAMES) | MIST | | % <FLUXES>  % <FLUXES\_SECTION:0\_pk>  PK\_D1FRAC = PK\_THETA(27)  PK\_OCC = time/24 ;  PK\_Ccent = PK\_A2/PK\_OCC + X\*100  % </FLUXES\_SECTION:0\_pk>  % </FLUXES> | // <FLUXES>  // <FLUXES\_SECTION:0\_pk>  realtype PK\_D1FRAC = UP(PK\_THETA\_27)  realtype PK\_OCC = UP(time)/24 ;  realtype PK\_Ccent = Ith(y,PK\_A2)/PK\_OCC + UP(X)\*100  // </FLUXES\_SECTION:0\_pk>  // </FLUXES> | |
| INIT\_COND\_COLUMN\_NAME | The first the left-hand side of the first assignment under this tag will replace the whole comment that contains this tag in MIST  Example where /\*<INIT\_COND\_COLUMN\_NAME>  </INIT\_COND\_COLUMN\_NAME>\*/ was found as a parameter of ADD\_COLUMN and GET\_COLUMN in MIST   |  |  | | --- | --- | | MATLAB | MIST | | % < INIT\_COND\_COLUMN\_NAME>  A = 1;  B = 2;  % </INIT\_COND\_COLUMN\_NAME> | /\* INIT\_CONDS \*/  ADD\_COLUMN(A, N\_SPECIES);  #define IM(n) GET\_COLUMN(A, n) | |
| PARAMETERS\_COLUMN\_NAME | The first left-hand side of the first assignment under this tag will replace the whole comment that contains this tag in MIST  Example where /\*<PARAMETERS\_COLUMN\_NAME>  </PARAMETERS\_COLUMN\_NAME>\*/ was found as a parameter of ADD\_COLUMN and GET\_COLUMN in MIST   |  |  | | --- | --- | | MATLAB | MIST | | % <PARAMETERS\_COLUMN\_NAME>  A = 1;  B = 2;  % </PARAMETERS\_COLUMN\_NAME> | /\* INIT\_CONDS \*/  ADD\_COLUMN(A, N\_SPECIES);  #define IM(n) GET\_COLUMN(A, n) | |
| NUMBER\_OF\_OTHER\_PARAMETERS | The number of comma-separated values identified under the section ALL\_OTHER\_PARAMETERS in MATLAB will replace the whole comment that contains this tag in MIST  Example where /\*<NUMBER\_OF\_OTHER\_PARAMETERS>  </ NUMBER\_OF\_OTHER\_PARAMETERS >\*/ was found after U01 in MIST   |  |  | | --- | --- | | MATLAB | MIST | |  | #define U01 10 | |
| NUMBER\_OF\_SPECIES | The number of comma-separated values identified under the section SPECIES\_NAMES in MATLAB will replace the whole comment that contains this tag in MIST  Example where /\*<NUMBER\_OF\_SPECIES>  </NUMBER\_OF\_ SPECIES>\*/ was found after N\_SPECIES in MIST   |  |  | | --- | --- | | MATLAB | MIST | |  | #define N\_SPECIES 10 | |
| ODES  ODES\_SECTION:name1  ODES\_SECTION:name2  …  ODES\_SECTION:nameN | Each ODES\_SECTION found in MATLAB will be translated between ODES tag in MIST ordered by name lexicographycally.  Translation rules for code found in ODES\_SECTION:   1. Each line of MATLAB code should be an assignment. 2. The left-hand side of each assignment should be a specific value of a vector, accessed through an index variable. 3. The left-hand side MIST translation will be  Ith(ydot, <INDEX\_VARIABLE\_NAME>) 4. The right-hand side MIST translation will be equal to the MATLAB right-hand side.  |  |  | | --- | --- | | MATLAB | MIST | | % <ODES>  % <ODES\_SECTION:0\_pk>  dydt(index\_PK\_AUC) = v001;  dydt(index\_PK\_A1) = v002;  dydt(index\_PK\_A2)= v003-v004+v005-v006;  dydt(index\_PK\_A3) = v004 - v005;  % </ODES\_SECTION:0\_pk>  % </ODES> | // <ODES>  // <ODES\_SECTION:0\_pk>  Ith(ydot, index\_PK\_AUC) = V001;  Ith(ydot, index\_PK\_A1) = v002;  Ith(ydot, index\_PK\_A2)= v003-v004+v005-v006;  Ith(ydot, index\_PK\_A3) = v004 - v005;  // </ODES\_SECTION:0\_pk>  // </ODES> | |