

General Structure

```
=====
```

BEGSIM

```
Comments
End Comments
```

```
SPECIES
End SPECIES
```

```
Solution Classes
End Solution Classes
```

```
CHEMISTRY
END CHEMISTRY
```

```
SIMULATION PARAMETERS
END SIMULATION PARAMETERS
```

ENDSIM

```
=====
```

A mozart preprocessor input file must have the above general structure.
Note how each key pairs are of the form

```
keyword
end keyword
```

except for the begsim, endsim pair.

The keywords begsim, species, end species, Solution Classes, end solution classes, chemistry, end chemistry, simulation parameters, end simulation parameters and endsim are mandatory and must be ordered as above.

Its hard to discern from this skeleton example but the mozart preprocessor is in general case insensitive. For instance, in example 1 you could have the SPECIES line be input as SpEcieS and the preprocessor would still interpret this as the species keyword. Furthermore, in general, white space is ignored. In example 1 there are several blank lines. The preprocessor always ignores blank lines. Just as you could input the SPECIES line as SpEcieS you could also input this line as S pE cie S and it would be a valid, although less readable input.

This is a good point to bring up some basic syntax rules of the mozart preprocessor.

- (1) Input lines are limited to 120 characters. White space at the beginning of a line counts in the character limit. Input lines greater than 120 characters are truncated and can cause preprocessing errors.
- (2) Input lines are rather free form; they may start in any column.
- (3) Blank lines are ignored
- (4) Lines in which the first character is an "*" are considered comment lines and as such are also ignored.


```
(2) New merged code version
(3) maccm3 inputs
(4) New isoprene chemistry
added species: CH3OH, C2H5OH, GLYALD, HYAC, EO2, EO, HYDRALD
```

If we had entered the line :

```
"(3) maccm3 inputs"
```

as

```
(3) maccm3 inputs
```

then the document file output would look like :

```
This is a mozart2 simulation with :
(1) The new Lin and Rood advection routine
(2) New merged code version
(3)maccm3inputs
(4) New isoprene chemistry
added species: CH3OH, C2H5OH, GLYALD, HYAC, EO2, EO, HYDRALD
```

That's it for the comments section.

The Species section

SPECIES

```
Solution
End Solution
```

```
Fixed
End Fixed
```

```
Groups
End Groups
```

```
Col-int
End Col-int
```

End SPECIES

This is the general form of the species section. There are 4 sub-sections: solution, fixed, groups, and col-int. The solution and fixed sub-sections are required, the groups and col-int are optional. The solution and fixed species must be specified first and in that order. In the following we will go into detail sub-section by sub-section.

The solution sub-section (mandatory)

Solution

O3, O, O1D -> O, N2O, N, NO, NO2, NO3, HNO3, HO2NO2, N2O5, CH4, CH3O2
CH3OOH, CH2O, CO, OH, HO2, H2O2, C3H6, ISOP -> C5H8, PO2 -> C3H6OHO2, CH3CHO
POOH -> C3H6OHOOH, CH3CO3, CH3COOH, PAN -> CH3CO3NO2
ONIT -> CH3COCH2ONO2, C2H6, C2H4, C4H10, MPAN -> CH2CCH3CO3NO2
ISOPO2 -> HOCH2COOCH3CHCH2, MVK -> CH2CHCOCH3, MACR -> CH2CCH3CHO
MACRO2 -> CH3COCHO2CH2OH, MACROOH -> CH3COCHOHCH2OH
MCO3 -> CH2CCH3CO3, C2H5O2, C2H5OOH, C10H16
C3H8, C3H7O2, C3H7OOH, CH3COCH3, ROOH -> CH3COCH2OOH
CH3OH, C2H5OH, GLYALD -> HOCH2CHO, HYAC -> CH3COCH2OH, EO2 -> HOCH2CH2O2
EO -> HOCH2CH2O, HYDRALD -> HOCH2CCH3CHCHO
RO2 -> CH3COCH2O2, CH3COCHO, Rn, Pb
ISOPNO3 -> CH2CHCCH3OOCH2ONO2, ONITR -> CH2CCH3CHONO2CH2OH
XO2 -> HOCH2COOCH3CHCHOH, XOOH -> HOCH2COOHCH3CHCHOH, ISOPOOH -> HOCH2COOHCH3CHCH2
H2, O3S -> O3, O3INERT -> O3
End Solution

Solution function

Specify all primary solution species. As a part of this input the molecular weight of each solution species is specified.

Syntax and limits

-
- (1) 300 solution species limit per simulation
 - (2) Solution species are limited to eight alphanumeric characters
 - (3) Solution species are case sensitive
 - (4) Solution species "aliases" are limited to 32 alphanumeric characters

The mozart model employs both mass mixing ratio and volumetric mixing units internally. Specifically, only the chemistry modules utilize volumetric mixing ratio. Thus mozart has to convert to and from volumetric mixing ratios. This requires the molecular weight of each solution species. Thus one way or another all solution species need to have their symbolic name relate to their chemical composition as represented by the standard chemical periodic table. For instance, in the above input ozone is specified as O3 not o3 since the in the periodic table O represents atomic oxygen whereas o is not represented. Numbers following valid periodic table elements act exactly as you would expect - they indicate quantity. Invalid periodic elements are ignored in the molecular weight specification; they do not cause a preprocessor error halt.

Please note the aliasing mechanism in the solution species subsection. For example, the entry PAN -> CH3CO3NO2. Here the solution species labeled PAN actually represents the compound CH3CO3NO2. Why is this necessary? Remember

due to code history the preprocessor allows only eight characters for the each solution species.

This is the most critical section of all. Make a subtle logic error here and it can be hard to track down latter. One of the most common errors is to misrepresent the the chemical formulation of a compound. In that case the true molecular mass will be erroneous and you may well get actual volumetric mixing ratios in your outputs that don't correspond to known limits.

Note that CO and Co are not the same species; the first is carbon monoxide and the second is cobalt. While HNO4 and HO2NO2 have the same molecular weight they are two separate species to the preprocessor.

The fixed sub-section (mandatory)

```
Fixed
M, N2, O2, H2O
End Fixed
```

```
Fixed function
-----
```

Specify all "fixed" species. Fixed species are species that participate in chemical reactions but are not solution species. Separate routines exist to specify these species values at each time step during a simulation.

```
Syntax and limits
-----
```

- (1) 300 fixed species limit per simulation
- (2) Fixed species are limited to eight alphanumeric characters
- (3) The symbol "M" represents total atmospheric density and must appear in the Fixed sub-section

The example fixed sub-section is fairly typical with molecular nitrogen, molecular oxygen, and water vapor declared as fixed species as well as the mandatory total atmospheric density. Note that unlike the solution species fixed species do not have any periodic table matching requirements and their molecular weight is not computed.

The groups sub-section (optional)

```
Groups
OX = O3 + O1D + O
End Groups
```

```
Groups function
-----
```

Specify all groups or "families". Groups are composed of solution species. Groups are optional and are often used to reduce the number of species in a actual simulation. Group members, those solution species appearing to the right

of the equal operator, are removed from the species that are directly solved. In their place is the group.

Syntax and limits

- (1) 20 groups limit per simulation
- (2) Groups are limited to eight alphanumeric characters
- (3) Groups must be specified one to each input line (120 char limit)
- (4) Although Group members can be weighted the preprocessor does NOT properly handle non-unity weights
- (5) All Group members must be solution species; they must be entered exactly as specified in the solution species sub-section
- (6) Group and group members are separated by a single equal operator
- (7) Group members are separated by the "+" character

The example groups sub-section specifies a single group, OX, composed of equal parts O3, O1D, and O. In the actual simulation only OX is transported and chemically active. A routine must be supplied that specifically partitions OX into the three members. Although they are not explicitly simulation species, O3, O1D, and O are formed and used inside the chemistry modules and can be output.

The col-int sub-section (optional)

Col-int

```
O3 = 0.  
O2 = 0.  
End Col-int
```

Col-int function

The col-int subsection, short for column integral, defines those solution species for which a vertical column integral is required.

Syntax and limits

- (1) 300 col-int entries per simulation
- (2) All col-int entries must be either solution or fixed species; they must be entered exactly as specified in the solution or fixed species subsection
- (3) Each col-int entry must be assigned an upper boundary value
- (4) Each col-int entry and upper boundary value are separated by the = operator
- (5) Although NOT used by the simulation the upper boundary value must be a valid fortran syntax number

The example col-int subsection specifies that O3 and O2 will have column integrals formed in the mozart simulation. If no col-int species are defined then mozart standard code in the photolysis routine will not call any column integral routines. The entries in the col-int example are placed on per line for clarity. They could have been specified in one line via :

```
O3=0., O2=0.
```

The Solution Classes section

Solution Classes

```
Explicit
End Explicit
```

```
Implicit
End Implicit
```

```
Rodas
End Rodas
```

```
End Solution Classes
```

This is the general form of the solution classes section. There are 3 subsections: explicit, implicit, and rodas. All solution species with potential group modifications must be included in one and only one of the three subsections. The order in which the solution classes are specified does not matter.

The explicit subsection (optional)

```
Explicit
  CH4, N2O, CO, Rn, Pb, H2, O3INERT, O3S
End Explicit
```

Explicit class function

Specify all species to be solved via the forward Euler or explicit algorithm. Be careful with this solution class. Note that all the above species listed in this class are presumed to have rather gentle chemistry with longer lifetimes. It would be disastrous to the simulation to place a highly reactive species such as OH in the explicit solution class.

Syntax and limits

-
- (1) 300 species limit per simulation
 - (2) All explicit class members must be solution or group species; they must be entered exactly as specified in the solution or groups subsection of the species section
 - (3) One may use the entry "All" to place all solution/group species in the explicit solution class as in :

```
Explicit
  All
End Explicit
```

The implicit subsection (optional)

Implicit

OX, N, NO, NO2, NO3, HNO3, HO2NO2, N2O5, CH3O2
CH3OOH, CH2O, OH, HO2, H2O2, C3H6, ISOP, PO2, CH3CHO
POOH, CH3CO3, CH3COOOH, PAN, ONIT, C2H6, C2H4, C4H10, MPAN
ISOPO2, MVK, MACR, MACRO2, MACROOH
MCO3, C2H5O2, C2H5OOH, C10H16
C3H8, C3H7O2, C3H7OOH, CH3COCH3, ROOH
CH3OH, C2H5OH, GLYALD, HYAC, EO2
EO, HYDRALD, RO2, CH3COCHO, ISOPNO3, ONITR
XO2, XOOH, ISOPOOH
End Implicit

Implicit class function

Specify all species to be solved via the backward Euler or implicit algorithm.
This is the "work horse" solution class. If in doubt put a species in this
class.

Syntax and limits

The limits and syntax rules for the implicit class as the same
as those for the explicit class.

Note that in this simulation that OX is declared as an implicit solution
species. Furthermore note that neither O3 or O1D or O are included in
any of the solution classes since they will all be removed in the simulation
species list and replaced with the group OX.

The rodas subsection (optional)

Rodas
End Rodas

Rodas class function

Specify all species to be solved via the implicit Rosenbrock solver Rodas.
This is the "cadillac " solution class. Again as with the implicit solver
class you may put anything in this class. This class is about twice as
expensive as the implicit class and should really only be used for situations
where the implicit solver exhibits a large number of failure to converge
messages. If you use this class be careful with the per species relative
error criterion. Setting to strigent a criteria, generally $< 1.e-3$, can
cause the computational time to increase greatly.

Syntax and limits

The limits and syntax rules for the rodas class as the same
as those for the explicit class.

The example has does not use the rodas solver for any species. Mozart has executed hundreds of simulation years in production with scientifically acceptable results without ever using the rodas solver.

The Chemistry section

CHEMISTRY

Photolysis
End Photolysis

Reactions
End Reactions

Heterogeneous
End Heterogeneous

Ext Forcing
End Ext Forcing

END CHEMISTRY

Above is the general form of the chemistry section. There are 4 subsections: photolysis, reactions, heterogeneous, and ext forcing. Interestingly, none of the subsections are mandatory. In that case you will in effect be doing a passive tracer simulation with no chemistry. Most simulations will have entries in all the subsections. The Chemistry, End Chemistry section delimiters are always required, even if there is no real chemistry. If they exist photolysis reactions must be specified first in the Photolysis subsection and gas phase reactions second in the Reactions subsection. Otherwise subsection order is immaterial.

The following details general characteristics of both photo and gas phase reactions(Reactions section).

Reactants and products are limited to eight characters. Reactants are restricted to be either solution species as possibly modified by groups or fixed species. Any reactant not in the solution or fixed groups will cause a preprocessor error halt. Any eight character alphanumeric string is allowed for products. Products that are neither solution or fixed species are flagged in the document file. They are enclosed in the {} pair as in {XYX}, assuming XYX is not a valid solution or fixed species.

Reactants may not have an explicit coefficient. Reactants always have implied unity coefficients. Reactants are separated from each other by the "+" operator. Products are separated from each other by either the "+" or "-" operator. It seems strange but you may in essence have negative products. Blame this on complex hydrocarbon chemistry. Products may have any valid fortran number as a coefficient. Again no explicit coefficient is an implied unity coefficient. A coefficient, reactant product pair is

separated by the "*" operator as in $2.5*OH$. The coefficients are checked for validity and can cause an error halt. Reactions that do not fit on one input line may be continued on subsequent lines. The first non-blank character of the continued lines must be a product separator; either "+" or "-". A reaction line must not end with a product separator. You can not break an input reaction in the reactants and must have at least one product on the first input line.

Reactants are separated from products with either the "->" or "=" operator. Either is allowed in a given reaction. Specifics regarding reactant limits and gas phase reaction rate constants will be covered in the photolysis and reactions subsections below. All reactions may have up to 16 products.

The "tags" at the beginning of a reaction line, enclosed by the [] pair, are useful for coding actual reaction rate routines. These tags can be used to identify individual reactions in the reaction matrix. Thus you can potentially change the order of reactions, add or delete reactions and not have to worry about the mapping of a given reaction to a location in the actual fortran reaction array. Tags are optional. Tags are limited to 16 characters not including the enclosing [] pair.

The combined photolysis and gas phase reaction count is limited to 900.

The photolysis subsection (optional)

Photolysis

```
[jo2] O2 + hv -> 2*O
[jo1d] O3 + hv -> O1D + O2
[jo3p] O3 + hv -> O + O2
[jn2o] N2O + hv -> O1D + N2
[jno] NO + hv -> N + O
[jno2] NO2 + hv -> NO + O
[jn2o5] N2O5 + hv -> NO2 + NO3
[jhno3] HNO3 + hv -> NO2 + OH
[jno3] NO3 + hv -> .89*NO2 + .11*NO + .89*O3
[jho2no2] HO2NO2 + hv -> NO2 + HO2
[jch3ooh] CH3OOH + hv -> CH2O + HO2 + OH
[jch2o_a] CH2O + hv -> CO + 2 * HO2
[jch2o_b] CH2O + hv -> CO + H2
[jh2o] H2O + hv -> OH + HO2
[jh2o2] H2O2 + hv -> 2*OH
[jch3cho] CH3CHO + hv -> CH3O2 + CO + HO2
[jpooh] POOH + hv -> CH3CHO + CH2O + HO2 + OH
[jch3co3h] CH3COOOH + hv -> CH3O2 + OH + CO2
[jpan] PAN + hv -> .6*CH3CO3 + .6*NO2 + .4*CH3O2 + .4*NO3
[jmpan] MPAN + hv -> MCO3 + NO2
[jmacr_a] MACR -> 1.34 * HO2 + .66 * MCO3 + 1.34 * CH2O + 1.34 * CH3CO3
[jmacr_b] MACR -> .66 * OH + 1.34 * CO
[jmrvk] MVK + hv -> .7 * C3H6 + .7 * CO + .3 * CH3O2 + .3 * CH3CO3
[jc2h5ooh] C2H5OOH + hv -> CH3CHO + HO2 + OH
[jc3h7ooh] C3H7OOH + hv -> 0.82 * CH3COCH3 + OH + HO2
[jrooh] ROOH + hv -> CH3CO3 + CH2O + OH
[jacet] CH3COCH3 + hv -> CH3CO3 + CH3O2
[jmgly] CH3COCHO + hv -> CH3CO3 + CO + HO2
[jxooh] XOOH + hv -> OH
[jonitr] ONITR + hv -> HO2 + CO + NO2 + CH2O
```

```
[jisopoo] ISOPOOH + hv -> .402 * MVK + .288 * MACR + .69 * CH2O + HO2
[jhyac] HYAC + hv -> CH3CO3 + HO2 + CH2O
[jglyald] GLYALD + hv -> 2 * HO2 + CO + CH2O
End Photolysis
```

Photolysis function

Specify all photolysis reactions. Photolysis reactions have only one and only one true reactant. They must have the "hv" symbol as a symbolic second reactant.

Syntax and limits

- (1) Up to 900 photo reactions limit per simulation
- (2) Reactants must be solution or fixed species
- (3) Reactants understood to have unity coefficients; no reactant coefficients allowed
- (4) 16 products allowed
- (5) "hv" symbol required as second reactant
- (6) No explicit rate constant may be assigned; see rate constant information in the reactions section below

The reactions subsection (optional)

Reactions

```
[usr1] O + O2 + M -> O3 + M
      O + O3 -> 2*O2 ; 8e-12, -2060
[old_n2] O1D + N2 -> O + N2 ; 1.8e-11, 110
[old_o2] O1D + O2 -> O + O2 ; 3.2e-11, 70
[ox_l1] O1D + H2O -> 2*OH ; 2.2e-10
      N2O + O1D -> 2*NO ; 6.7e-11
      N2O + O1D -> N2 + O2 ; 4.9e-11
[ox_p1] NO + HO2 -> NO2 + OH ; 3.5e-12, 250
      NO + O3 -> NO2 + O2 ; 3e-12, -1500
      NO2 + O -> NO + O2 ; 5.6e-12, 180
      NO2 + O3 -> NO3 + O2 ; 1.2e-13, -2450
      NO3 + HO2 -> OH + NO2 ; 2.3e-12, 170.
[usr2] NO2 + NO3 + M -> N2O5 + M ; 2.e-30,4.4, 1.4e-12,.7, .6
[usr3] N2O5 + M -> NO2 + NO3 + M
      N2O5 + H2O -> 2*HNO3 ; 0.
[usr4] NO2 + OH + M -> HNO3 + M ; 2.4e-30,3.1, 1.7e-11,2.1, .6
[usr5] HNO3 + OH -> NO3 + H2O
      NO3 + NO -> 2*NO2 ; 1.5e-11, 170
[usr6] NO2 + HO2 + M -> HO2NO2 + M ; 1.8e-31,3.2, 4.7e-12,1.4, .6
      HO2NO2 + OH -> H2O + NO2 + O2 ; 1.3e-12, 380
[usr7] HO2NO2 + M -> HO2 + NO2 + M
      CH4 + OH -> CH3O2 + H2O ; 2.45e-12, -1775
      CH4 + O1D -> .75 * CH3O2 + .75 * OH + .25 * CH2O + .4 * HO2 + .05 * H2 ; 1.5e-10
[ox_p2] CH3O2 + NO -> CH2O + NO2 + HO2 ; 3.e-12, 280
      CH3O2 + CH3O2 -> 2 * CH2O + 2 * HO2 ; 5.e-13,-424
      CH3O2 + CH3O2 -> CH2O + CH3OH ; 1.9e-14,706
      CH3O2 + HO2 -> CH3OOH + O2 ; 3.8e-13, 800
      CH3OOH + OH -> .7 * CH3O2 + .3 * OH + .3 * CH2O + H2O ; 3.8e-12, 200
      CH2O + NO3 -> CO + HO2 + HNO3 ; 6.0e-13, -2058
```

```

CH2O + OH -> CO + H2O +HO2          ; 1e-11
[usr8] CO + OH -> CO2 + HO2
H2 + O1D -> HO2 + OH                  ; 1.1e-10
O + OH -> HO2 + O2                    ; 2.2e-11, 120
HO2 + O -> OH + O2                    ; 3e-11, 200
[ox_l2] OH + O3 -> HO2 + O2           ; 1.5e-12, -880
[ox_l3] HO2 + O3 -> OH + 2*O2         ; 2e-14, -680
[usr9] HO2 + HO2 -> H2O2
H2O2 + OH -> H2O + HO2                ; 2.9e-12, -160
OH + HO2 -> H2O + O2                  ; 4.8e-11, 250
OH + OH -> H2O + O                    ; 4.2e-12, -240
H2 + OH -> H2O + HO2                  ; 5.5e-12, -2000
[usr10] C3H6 + OH + M -> PO2 + M       ; 8.e-27,3.5, 3.e-11,0, .5
[ox_l4] C3H6 + O3 -> .54*CH2O + .19*HO2 + .33*OH          ; 6.5e-15, -1900
          + .08*CH4 + .56*CO + .5*CH3CHO + .31*CH3O2 + .25*CH3COOH
C3H6 + NO3 -> ONIT                     ; 4.6e-13,-1156
[ox_p3] PO2 + NO -> CH3CHO + CH2O + HO2 + NO2 ; 4.2e-12, 180
PO2 + HO2 -> POOH + O2                 ; 7.5e-13, 700
POOH + OH -> .5*PO2 + .5*OH + .5*HYAC + H2O ; 3.8e-12, 200
CH3CHO + OH -> CH3CO3 + H2O           ; 5.6e-12, 270
CH3CHO + NO3 -> CH3CO3 + HNO3         ; 1.4e-12, -1900
[ox_p4] CH3CO3 + NO -> CH3O2 + CO2 + NO2 ; 8.1e-12, 270
[usr11] CH3CO3 + NO2 + M -> PAN + M     ; 8.5e-29,6.5, 1.1e-11,1., .6
CH3CO3 + HO2 -> .7 * CH3COOOH + .3 * CH3COOH + .3 * O3 ; 4.3e-13, 1040
CH3CO3 + CH3O2 -> .9*CH3O2 + CH2O + .9*HO2 + .9*CO2 + .1*CH3COOH ; 1.3e-12,640
*-----
* note the reaction immediately below is "commented out" and will not
* be in the reaction mechanism
*-----
* CH3COOOH + OH -> CH3CO3 + H2O ; 1e-12
CH3COOOH + OH -> .5*CH3CO3 + .5*CH2O + .5*CO2 + H2O ; 1e-12
[usr12] PAN + M -> CH3CO3 + NO2 + M
CH3CO3 + CH3CO3 -> 2*CH3O2 + 2*CO2 ; 2.5e-12, 500
[ox_l5] ISOP + O3 -> .4 * MACR + .2 * MVK + .07 * C3H6 + .27 * OH ; 1.05e-14, -2000
          + .06 * HO2 + .6 * CH2O + .3 * CO + .1 * O3
          + .2 * MCO3 + .2 * CH3COOH
OH + C2H6 -> C2H5O2 + H2O ; 8.7e-12, -1070
[ox_p5] C2H5O2 + NO -> CH3CHO + HO2 + NO2 ; 2.6e-12, 365
C2H5O2 + HO2 -> C2H5OOH + O2 ; 7.5e-13, 700
C2H5O2 + CH3O2 -> .7 * CH2O + .8 * CH3CHO + HO2 ; 2.e-13
          + .3 * CH3OH + .2 * C2H5OH
C2H5O2 + C2H5O2 -> 1.6 * CH3CHO + 1.2 * HO2 + .4 * C2H5OH ; 6.8e-14
C2H5OOH + OH -> .5 * C2H5O2 + .5 * CH3CHO + .5 * OH ; 3.8e-12, 200
[usr13] OH + C2H4 + M -> .75*EO2 + .5*CH2O + .25*HO2 + M ; 1.e-28,.8, 8.8e-12,0.,
.6

```

End Reactions

Reactions function

Specify all gas phase reactions. Gas phase reactions must have one reactant and may have up to three reactants. All reactants must be either solution or fixed species. If a reaction has three reactants then at most two can be solution species and at least one must be a fixed species. For example, if OH and HO2 are solution species, the following is an invalid gas phase reaction :

OH + OH + HO2 -> H2O2

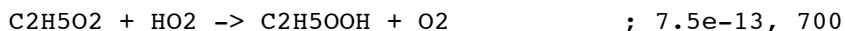
If you really must have three solution species reactants then contact Stacy Walters at ncar; stacy@ucar.edu

Gas phase reactions may specify rate constants. Rate constants are delimited from reaction products by the ";" character. There are two types of rate constants; arrenhius and troe.

The general arrenhius rate constant is of the form :

$$a_0 * \exp(b_0/t)$$

where a0 and b0 are constants to be specified and t is temperature. a0 and b0 are input as in :

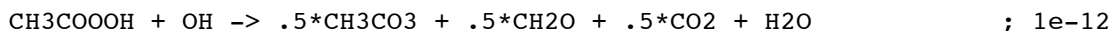


where a0 = 7.5e-13 and b0 = 700

a0 and b0 are checked for fortran numeric validity. They may be positive or negative.

NOTE: The JPL book provides A and (E/R) in $k(T) = A \cdot \exp((-E/R) (1/T))$, whereas here $b_0 = (-E/R)$

A temperature independent arrenhius rate only has the a0 term as in :



where a0 = 1e-12

The general troe rate constant is of the form :

$$\alpha^{**x} / (1 + \beta^{**2})$$

where

alpha = k0*M/kinf
beta = log10(alpha)
M = total atmospheric density (molecules/cm**3)
x = "exponential factor"
k0 = a0*(300/t)**a1
kinf = b0*(300/t)**b1
t = temperature (degrees Kelvin)

a0,a1,x,b0,b1 are rate constant inputs to be specified in that order as in :



Here a0 = 8.5e-29, a1 = 6.5, b0 = 1.1e-11, b1 = 1., and x = .6

Whether arrenhius or troe rates, the reaction rate constants must be input on only the first line of a reaction.

Gas phase reactions with no specified rate constant are labeled as "user defined" in the document file and as such their rate

constant must be supplied in a user supplied subroutine(mo_usrxrt.F90). Failure to supply a rate constant for such a reaction can lead to a bogus simulation. At best the simulation will rapidly break down with some sort of runtime exception. At worst the simulation will complete without incident, however the results will be erroneous.

Each input rate constant parameter, a0,a1,x,b0,b1, is limited to 16 characters. Parameters are delimited by the "," character.

Syntax and limits

- (1) Up to 900 gas phase reactions limit per simulation
- (2) Reactants must be solution or fixed species
- (3) Reactants understood to have unity coefficients; no reactant coefficients allowed
- (4) 16 products allowed

The heterogeneous subsection (optional*)

Heterogeneous

H2O2, HNO3, CH2O, CH3OOH, POOH, CH3COOOH, HO2NO2, ONIT, MVK, MACR, C2H5OOH
C3H7OOH, ROOH, CH3COCHO, Pb, MACROOH, XOOH, ONITR, ISOPOOH
CH3OH, C2H5OH, GLYALD, HYAC, HYDRALD, CH3CHO, ISOPNO3

End Heterogeneous

Heterogeneous function

Specify all solution species that are removed by wet deposition(washout).

Although this section is optional species not specifically listed in the Heterogeneous subsection will not be have any wet removal in the simulation.

Syntax and limits

- (1) Up to 300 heterogeneous entries
- (2) Only solution species allowed
- (3) Each individual species must appear only once

The ext forcing subsection (optional*)

Ext Forcing

NO, CO, CH4

End Ext Forcing

Ext forcing function

Specify all solution species that have "external" or insitu chemical forcing.

Although this section is optional species not specifically listed in the Ext forcing subsection will not be have any external forcing in the simulation.

Syntax and limits

- (1) Up to 300 ext forcing entries
- (2) Only solution species allowed
- (3) Each individual species must appear only once

The Simulation Parameters section

SIMULATION PARAMETERS

Spatial Dimensions
End Spatial Dimensions

Numerical Control
End Numerical Control

Surface Flux
End Surface Flux

Surface Deposition
End Surface Deposition

Version Options
End Version Options

Outputs
End Outputs

END SIMULATION PARAMETERS

Above is the general form of the simulation parameters section. This is a long section with 6 subsections: Spatial dimensions, Numerical control, Surface Flux, Surface Deposition, Version Options, and Outputs. Spatial dimensions and Version Options are mandatory. Outputs are highly recommended if you want to archive your simulation results. Surface Flux and Surface Deposition specify which species have a surface emission and/or dry deposition. Subsection order is unrestrained.

The spatial dimensions subsection (mandatory)

Spatial Dimensions
Longitude points = 128
Latitude points = 64

```
Vertical points = 34
End Spatial Dimensions
```

Spatial dimensions function

Specify the size of the three spatial dimensions.

Syntax and limits

The numeric values entered are checked for fortran integer validity. The values must be positive integers. All three dimensions must be specified; order is immaterial. The vertical points must be equal to or less than the actual number of vertical levels in the dynamical netcdf input files.

If running in hybrid mode, using both OpenMP and MPI, or in pure MPI mode then latitude points must be ≥ 4 and an even multiple of the mpi task count. For example, with the above specification of 64 latitudes you can not run moztart with 3 or 5 mpi tasks. In fact with the above specification for latitude points you can only run moztart on $2*n$ mpi tasks where $n = 1, \dots, 4$. If running in hybrid or in pure OpenMP mode the number of longitudes is constrained to be a multiple of the longitude_tiles variable. In a typical hybrid moztart simulation the longitude_tiles variable is set to 8. Thus 128 longitude points will not cause a runtime error whereas a value of say 127 would be very hard to accomodate.

These restrictions do not produce preprocessing errors but will cause the actual simulation to halt before doing any real work.

The Version Options subsection (mandatory)

Version Options

```
machine = Intel
architecture = hybrid
* vec_ftns = on
multitask = on
namemod = on
modules = on
* cpp_opts = "-P -C -I. -traditional"
```

End Version Options

Version Options function

Specify target execution environment architecture details.

Note the two entries that are "commented" out. Many vendors supply special routines for transcendental functions such as exp, log, sqrt, The vec_ftns parameter allows for such function activation in the executable. Currently the only such environment where this provision is implemented is on AIX based systems from IBM. The vec_ftns default to off.

The `cpp_opts` parameter is actually intended for use by the preprocessor and allows one to specify options to be used by the `cpp` command. The commented out `cpp_opts` setting is for use on a typical RedHat Linux system.

The surface flux subsection (optional*)

Surface Flux

NO, N2O, CH4, CH2O, CO, C3H6, ISOP, C10H16, C2H4, C2H6, C4H10
C3H8, CH3COCH3, Rn, H2, CH3OH

End Surface Flux

Surface Flux function

Specify species with a surface emission.

Although this section is optional `mozart` will, by default, assign a zero surface emission to all species not specifically listed in the Surface Flux subsection.

Syntax and limits

- (1) up to 300 entries
- (2) solution species only
- (3) entries delimited by the "," character

The surface deposition subsection (optional*)

Surface Deposition

OX, NO2, HNO3, CH4, CH3OOH, CH2O, CO, H2O2, POOH
CH3COOOH, PAN, MPAN, C2H5OOH, ONIT
C3H7OOH, ROOH, CH3COCHO, CH3COCH3, Pb, O3INERT, O3S, H2
ONITR, MACROOH, XOOH, ISOPOOH
CH3CHO, NO, HO2NO2

CH3OH, C2H5OH, GLYALD, HYAC, HYDRALD

End Surface Deposition

Surface Deposition function

Specify species with dry deposition at the lower boundary.

Although this section is optional `mozart` will, by default, assign a zero dry deposition velocity to all species not specifically listed in the Surface Deposition subsection.

Syntax and limits

- (1) up to 300 entries
- (2) solution species only

(3) entries delimited by the "," character

Note that OX appears in the Surface deposition subsection since this simulation has OX as a group representing the sum of O3, O1D, and O.

The outputs subsection (optional)

Outputs

File

Transported Species = avrg
End Transported Species

Group Members = avrg
End Group Members

Surface Flux = avrg
End Surface Flux

Deposition velocity
End Deposition velocity

Washout Rates = avrg
End Washout Rates

External Forcing = avrg
End External Forcing

Production = avrg
End Production

Loss = avrg
End Loss

Deposition flux
End Deposition flux

Massdiags = avrg
End Massdiags

End File
End Outputs

Outputs function

Specifies archival output for chemical species and many related diagnostics.

This a long subsection that has several of its own subsections. Outputs are specified file by file with a maximum of 10 output files; note the single File, End File pair in the preceding example. Files are understood to be ordered in the way they are input.

Within each File there are the following subsections :

Transported Species, Group Members, Surface Flux, Deposition velocity, Washout Rates, External Forcing, Production, Loss, Deposition, and Massdiags

First one general characteristic of all the subsections within the File subsection before going into each subsection in detail. Each subsection keyword may be assigned a time characteristic; either inst or avrg. Inst denotes instantaneous output and avrg denotes time averaged output. A keyword without a timing qualifier, such as Deposition flux in the above outputs example, defaults to instantaneous output. You may mix timing qualifiers within a File as is done in the example above. There are no restrictions for ordering the File subsections.

Since there is ample provision for output you should exercise some precaution when specifying simulation output. All output files are in netcdf format and as such are output by a single processor on the master mpi task. Voluminous output can both slow a simulation down and gobble up disk space rapidly. A typical mozart simulation writes between 2 to 4 gigabytes of archived data per simulation month.

Now for the **File subsections**.

```
=====
Transported Species = avrg
  All
End Transported Species
```

Transported Species function

Specifies which solution species to output in the file and the timing qualifier for all such listed species.

Note that instead of listing all the solution species individually the "All" token is used which will economically do the same thing.

Syntax, limits, and notes

- ```

```
- (1) solution species only
  - (2) entries delimited by the "," character
  - (3) All entry outputs every solution species
  - (4) Outputs are three dimensional (longitude, latitude, level)

```
=====
Group Members = avrg
 All
End Group Members
```

Group Members function

-----

Specifies which group species to output in the file and the timing qualifier for all such listed species. In this case the amount of input saved is small and an alternative specification would be :

```
Group Members = avrg
 03, 01D, 0
End Group Members
```

Syntax,limits, and notes

-----

- (1) group species only
- (2) entries delimited by the "," character
- (3) All entry outputs every group species
- (4) Outputs are three dimensional (longitude,latitude,level)

=====  
Surface Flux = avrg  
NO, N2O, CH4, CH2O, CO, C3H6, ISOP, C10H16, C2H4, C2H6, C4H10  
C3H8, CH3COCH3, Rn, H2, CH3OH  
End Surface Flux

Surface Flux function

-----

Specifies which solution species surface emission to output in the file and the timing qualifier for all such listed species. Note that in this case each individual species is listed since most of the species have a default surface emission of zero.

Syntax,limits, and notes

-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are two dimensional (longitude,latitude)

=====  
Deposition velocity  
OX, NO2, HNO3, CH4, CH3OOH, CH2O, CO, H2O2, POOH  
CH3COOOH, PAN, MPAN, C2H5OOH, ONIT  
C3H7OOH, ROOH, CH3COCHO, CH3COCH3, Pb, O3INERT, O3S, H2  
ONITR, MACROOH, XOOH, ISOPOOH  
CH3CHO, NO, HO2NO2, GLYALD, HYAC, CH3OH, C2H5OH, HYDRALD  
End Deposition velocity

Deposition velocity function

-----

Specifies which solution species dry deposition velocity to output in the file and the timing qualifier for all such listed species. Note that in this case each individual species is listed since many of the species have a default dry deposition velocity of zero.

Syntax,limits, and notes

-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are two dimensional (longitude,latitude)

=====  
Washout Rates = avrg  
H2O2, HNO3, CH2O, CH3OOH, POOH, CH3COOOH, HO2NO2, ONIT

MVK, MACR, C2H5OOH, C3H7OOH, ROOH, CH3COCHO, Pb  
MACROOH, XOOH, ONITR, ISOPOOH, GLYALD, HYAC, CH3OH, C2H5OH, HYDRALD  
End Washout Rates

Washout rates function

-----

Specifies which solution species wet removal rate constants to output in the file and the timing qualifier for all such listed species. Note that in this case each individual species is listed since many of the species are not removed by washout. Species with washout or wet removal are specified in the Heterogeneous subsection of the chemistry section( see pp.doc.5 ).

Syntax,limits, and notes

-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are three dimensional (longitude,latitude,level)

=====  
External Forcing = avrg  
NO, CH4, CO  
End External Forcing

External Forcing function

-----

Specifies solution species with external or insitu forcing to output in the file and the timing qualifier for all such listed species. Note that in this case each individual species is listed since most of the species do not have external forcing. Species with external forcing are specified in the Ext Forcing subsection of the chemistry section( see pp.doc.5 ).

Syntax,limits, and notes

-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are three dimensional (longitude,latitude,level)

=====  
Production = avrg  
OX  
End Production

Production function

-----

Specifies the chemical production for solution species to output in the file and the timing qualifier for all such listed species.

Syntax,limits, and notes

-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are three dimensional (longitude,latitude,level)

=====  
Loss = avrg  
    OX  
End Loss

Loss function  
-----

Specifies the chemical loss for solution species to output  
in the file and the timing qualifier for all such listed species.

Syntax,limits, and notes  
-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are three dimensional (longitude,latitude,level)

=====  
Deposition flux  
    OX, O3S, CO, NO, NO2, HNO3, PAN, MPAN  
End Deposition flux

Deposition function  
-----

Specifies which solution species dry deposition surface flux to output  
in the file and the timing qualifier for all such listed species.  
Note that in this case each individual species is listed since  
most of the species have a default dry deposition surface flux of zero.

Syntax,limits, and notes  
-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) All entry outputs every solution species
- (4) Outputs are two dimensional (longitude,latitude)

=====  
Massdiags = avrg  
    OX  
End Massdiags

Massdiags function  
-----

Specifies which solution species mass diagnostics to output  
in the file and the timing qualifier for all such listed species.  
This output subsection is a little deceptive. Mass diagnostics  
outputs 8 three dimensional fields for each entry. The fields

are :

(1) horizontal x mass flux, (2) horizontal y mass flux, (3) vertical mass flux  
and process cell mass changes for

(4) advection, (5) surface pressure change, (6) convection, (7) diffusion, (8) chemistry

This output can be very useful for detailed mass diagnostics but can produce a great deal of output since each entry spawns eight output fields.

Syntax, limits, and notes

-----

- (1) solution species only
- (2) entries delimited by the "," character
- (3) Outputs are three dimensional (longitude, latitude, level)