Appendix A

Input Format Notation

Appendix A

Input Format Notation

Notation	Description
{ Option }	Character string options are indicated by enclosing braces. Options are chosen by entering word(s) within the braces, exactly as shown. Only one option should be chosen for each data entry.
[Optional]	Enclosing brackets indicate optional characters or words. These characters can be entered into the input file to improve its readability or to specify optional features.
{{ Contains }}	Indicates the option contains a particular word. For example "Fractured Tuff" contains the word "Fractured" thus indicating a dual-porosity type rock/soil.
< *Data Types* >	Indicates repeated formatting.
Char[a]	Character string data type, referenced by superscript "a".
Integer[a]	Integer data type (no character data or decimal points) reference by superscript "a".
Real[a]	Real data type (decimal points and exponential notation are acceptable), reference by superscript "a".
#	A pound symbol in the first column indicates a comment line and will be ignored during execution. Comment lines may be placed inside or outside card structures. All lines outside of the card structures are ignored during execution.
~ Card Name	A tilde symbol in the first column indicates the start of a new card.
,	Data entries are comma delimited. Commas shown in the line format structures must be entered as shown, including a closing comma at the end of each line. Characters following the last comma of a data line are ignored during execution.

Units[a] (m)	Indicates the SI unit for the input data item referenced by superscript “a”.
|	Indicates a choice between more than one options
Format:	Indicates line formatting instructions and the beginning of a new input line. Each format statement requires a new input line.
Endcard:	Indicates end of a card.
For: Integer Instructions **Endfor:** Integer	Indicates instruction looping.
If: Name: Card = { Opt_1 } Instructions1 **Elseif:** Name: Card = { Opt_2 } Instructions2 **Elseif:** Instructions3 **Endif:**	Indicates decision logic.
IfDef: Opt_1 Instructions1 **ElseifDef:** Opt_2 Instructions2 **ElseDef:** Instructions3 **EndifDef:**	Indicates C preprocessor options and logic.
Note:	Indicates formatting information.

Appendix B

STOMP Input Control Card Formats

Appendix B

STOMP Input Control Card Formats

Contents

B.1 Atmospheric Conditions Card

Card Title[a] { ~Atmospheric [Conditions Card] }
Format: *Char*[a]

If: Operational Mode Option Card = { **STOMP-WAE-B** }

Atmospheric Start Time: Month[a], Atmospheric Start Time: Day[b],
Atmospheric Start Time: Year[c], Atmospheric Start Time: Time (military format)[d],
Wind Speed Measurement Height[e], Units[f] (m),
Air Temperature/Relative Humidity Measurement Height[g], Units[h] (m),
Local Longitude[i], Units[j] (deg),
Local Meridian[k], Units[l] (deg),
Format: *Char*[a], *Integer*[b], *Integer*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h], *Real*[i], *Char*[j], *Real*[k], *Char*[l],

If: Atmospheric Conditions Read From External File

External File with Atmospheric Conditions[a],
Format: *Char*[a],

Elseif: Atmospheric Conditions Read From Input File

Number of Atmospheric Condition Times[a],
Format: *Integer*[a],

For: Number of Atmospheric Condition Times
Atmospheric Condition Time[a], Units[b] (s),
Atmospheric Condition Temperature[c], Units[d] (K),
Atmospheric Condition Pressure[e], Units[f] (kg/(m s^2)),
Atmospheric Condition Water-vapor Relative Humidity[g],
Atmospheric Condition Net Solar Radiation[h], Units[i] (kg/m^3),
Atmospheric Condition Wind Speed[j], Units[k] (m/s),
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Real*[h], *Char*[i], *Real*[j], *Char*[k],
Endfor:

Endif:

Endif:

Endcard: Atmospheric Conditions Card

B.1.1 Atmospheric Conditions Card Examples

Extracted from a STOMP-WAE-B input file:

```
~Atmospheric Conditions Card
June,29,2000,00:00:00,2.0,m,2.0,m,119.627,deg,120.0,deg,
file,hms2july,
```

Extracted from a STOMP-WAE-B input file:

```
#-------------------------------------------------------
~Atmospheric Conditions Card
#-------------------------------------------------------
July,21,1997,00:00:00,15.0,m,2.0,m,112.95,deg,43.53,deg,112.95,deg,0.1,mm,0.1,mm,
19993,  # of atmospheric conditions
file,idaho_atm.dat,
```

Extracted from a STOMP-WAE-B input file:

```
#-------------------------------------------------------
~Atmospheric Conditions Card
#-------------------------------------------------------
May,23,1962,00:00:00,0.914,m,15.24,m,120.0,deg,46.57,deg,120.0,deg,4.9E-04,m,4.9E-04,m,
13,
0.000,day,58.000,F, 98199.213,Pa,0.560,0.000,langley/min,12.000,mi/hr,
0.042,day,58.000,F, 98165.444,Pa,0.560,0.000,langley/min, 5.000,mi/hr,
0.083,day,58.000,F, 98165.444,Pa,0.550,0.000,langley/min,10.000,mi/hr,
0.125,day,54.000,F, 98131.675,Pa,0.820,0.000,langley/min, 6.000,mi/hr,
0.167,day,52.000,F, 98131.675,Pa,0.910,0.000,langley/min,10.000,mi/hr,
0.208,day,52.000,F, 98131.675,Pa,0.890,0.020,langley/min,10.000,mi/hr,
0.250,day,51.000,F, 98165.444,Pa,0.970,0.070,langley/min, 6.000,mi/hr,
0.292,day,52.000,F, 98165.444,Pa,1.000,0.100,langley/min, 5.000,mi/hr,
0.333,day,52.000,F, 98165.444,Pa,1.000,0.160,langley/min,10.000,mi/hr,
0.375,day,52.000,F, 98165.444,Pa,0.950,0.230,langley/min, 8.000,mi/hr,
0.417,day,53.000,F, 98199.213,Pa,0.930,0.560,langley/min, 5.000,mi/hr,
0.458,day,56.000,F, 98199.213,Pa,0.850,0.380,langley/min, 0.000,mi/hr,
0.500,day,55.000,F, 98199.213,Pa,0.850,0.310,langley/min, 6.000,mi/hr,
```

B.2 Aqueous Relative Permeability Card

Card Title[a] { ~Aqueous Rel [ative Permeability Card] }
Format: *Char*[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section B.2.1

Elseif:

For: Number of Rock/Soil Types or Scaling Groups

If: Anisotropy Option is specified in combination with the Mualem permeability function option, the following two real parameters have to be added to the end of the input line (see example in B2.1):
Horizontal Pore-Scale Parameter (0.5), Horizontal Pore-Scale Parameter (0.5),
Endif:

If: Polmann Anisotropy Option is specified, the following eight real parameters have to be added to the end of the input line (see example in B2.1):
<LnKs> Mean of lnKs with Ks in cm/s,
σ^2_{LnKs} Variance of lnKs with Ks in cm/s,
p Slope of the β versus lnKs regression line with Ks in cm/s,
ζ Parameter with Ks in cm/s,
λ Vertical correlation lengths for lnKs with Ks in cm/s,
Mean slope, β, for lnKs versus ψ with Ks in cm/s,
Upper Anisotropy Ratio Limit,
Lower Anisotropy Ration Limit,
Endif:

Aqueous Relative Permeability Card (cont'd)

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
Rock/Soil or Scaling Group Name[a],
Permeability Function Option[b], (Polmann Option and Anisotropy Option (in combination with Mualem only) may be specified)
{ Constant | Mualem [Modified] [Irreducible] |
Burdine | Fatt and Klikoff | Corey |
Haverkamp | Tauma and Vauclin |
Tabular [Linear | Spline] [Water Content | Saturation] }
If: Permeability Function Option = { Constant }
If: Rock/Soil or Scaling Group Name = {{ Fractured }} {{ DP }}
Matrix Aqueous Relative Permeability[c],
Fracture Aqueous Relative Permeability[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Else:
Aqueous Relative Permeability[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
If: Saturation Function Option = {{ van Genuchten }}
and Rock/Soil or Scaling Group Name = {{ Fractured }} {{ DP }}
Matrix van Genuchten m parameter[c],
Fracture van Genuchten m parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
and Rock/Soil or Scaling Group Name = {{ Fractured }} {{ DP }}
Matrix Brooks and Corey λ parameter[c],
Fracture Brooks and Corey λ parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],
Format: *Char[a], Char[b], Real[c],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Modified Mualem }
If: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c], Pore-Scale Parameter[d] (0.5),
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c], Pore-Scale Parameter[d] (0.5),
Format: *Char[a], Char[b], Real[c], Real[d],*
Endif:
Elseif: Permeability Function Option = { Irreducible Mualem }
If: Saturation Function Option = {{ van Genuchten }}

van Genuchten m parameter[c], Irreducible Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c], Irreducible Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Endif:
Elseif: Permeability Function Option = { Fatt and Klikoff } { Corey }
Format: *Char[a], Char[b],*
Elseif: Permeability Function Option = { Haverkamp }
A Parameter[c], γ Parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Permeability Function Option = { Tauma and Vauclin }
α Parameter[c], β Parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Permeability Function Option = { Tabular Water Content | [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a], Char[b], Integer[c],*
For: Number of Table Entries
Aqueous Moisture Content[a], Aqueous Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor: Number of Table Entries
Elseif: Permeability Function Option = { Tabular [Saturation] [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a], Char[b], Integer[c],*
For: Number of Table Entries
Aqueous Saturation[a], Aqueous Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor: Number of Table Entries
Endif:

Aqueous Relative Permeability Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WA** } { **STOMP-WAE** } { **STOMP-WAE-R** } { **STOMP-WAE-Sc** } { **STOMP-WS** } { **STOMP-WS-Sc** } { **STOMP-WAS** } { **STOMP-WASE** }
Rock/Soil Name[a],
Permeability Function Option[b], (Polmann Anisotropy Option may be specified)
{ Constant | Mualem [Irreducible] | Burdine | Fatt and Klikoff | Corey | Haverkamp | Tauma and Vauclin |
Tabular [Linear | Spline] [Water Content | Saturation] }
If: Permeability Function Option = { Constant }
If: Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix Aqueous Relative Permeability[c],
Fracture Aqueous Relative Permeability[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Else:
Aqueous Relative Permeability[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
If: Saturation Function Option = {{ van Genuchten }}
and Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix van Genuchten m parameter[c],
Fracture van Genuchten m parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
and Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix Brooks and Corey λ parameter[c],
Fracture Brooks and Corey λ parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],
Format: *Char[a], Char[b], Real[c],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Irreducible Mualem }
If: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c], Irreducible Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c], Irreducible Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Endif:
Elseif: Permeability Function Option = { Fatt and Klikoff } { Corey }
Format: *Char[a], Char[b],*
Elseif: Permeability Function Option = { Haverkamp }

Aqueous Relative Permeability Card (cont'd)

A Parameter[c], γ Parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Permeability Function Option = { Tauma and Vauclin }
α Parameter[c], β Parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Permeability Function Option = { Tabular Water Content | [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a], Char[b], Integer[c],*
For: Number of Table Entries
Aqueous Moisture Content[a], Aqueous Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor: Number of Table Entries
Elseif: Permeability Function Option = { Tabular [Saturation] [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a], Char[b], Integer[c],*
For: Number of Table Entries
Aqueous Saturation[a], Aqueous Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor: Number of Table Entries
Endif:

Elseif: Operational Mode Option = { **STOMP-WO** } **{ STOMP-WOA }**
{ STOMP-WOA-Sc } { STOMP-WOAE } { **STOMP-WOD** } { **STOMP-WOM** }
Rock/Soil Name[a],
Permeability Function Option[b]
{ Constant | Mualem | Burdine }
If: Permeabililty Function Option = { Constant }
Aqueous Relative Permeability[c],
Format: *Char[a], Char[b], Real[c],*
Else:
If: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],
Format: *Char[a], Char[b], Real[c],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Endif:
Endif:

Aqueous Relative Permeability Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** }
Rock/Soil Name,
Permeability Function Option[b], (Polmann Anisotropy Option may be specified)
{ Constant | Mualem [Irreducible] | Burdine | Fatt and Klikoff | Corey | Free Corey | Haverkamp | Tauma and Vauclin | Tabular [Linear | Spline] [Water Content | Saturation] }
If: Permeability Function Option = { Constant }
If: Rock/Soil Name= {{ Fractured }} {{ DP }}
Matrix Aqueous Relative Permeability[c],
Fracture Aqueous Relative Permeability[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Else:
Aqueous Relative Permeability[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
If: Saturation Function Option = {{ van Genuchten }} and Rock/SoilName = {{ Fractured }} {{ DP }}
Matrix van Genuchten m parameter[c],
Fracture van Genuchten m parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }} and Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix Brooks and Corey λ parameter[c],
Fracture Brooks and Corey λ parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],
Format: *Char[a], Char[b], Real[c],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Irreducible Mualem }
If: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c], Irreducible Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c], Irreducible Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Endif:
Elseif: Permeability Function Option = { Fatt and Klikoff } { Corey }
Format: *Char[a], Char[b],*
Elseif: Permeability Function Option = { Haverkamp }
A Parameter[c], γ Parameter[d],

Aqueous Relative Permeability Card (cont'd)

Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Permeability Function Option = { Tauma and Vauclin }
α Parameter[c], β Parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
Elseif: Permeability Function Option = { Free Corey }
Endpoint Gas Permeability[c], Exponent Gas Relative Permeability[d],
Residual Aqueous Saturation[e], Residual Gas Saturation[f],
Format: *Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],*
Elseif: Permeability Function Option = { Tabular Water Content | [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a], Char[b], Integer[c],*
For: Number of Table Entries
Aqueous Moisture Content[a], Aqueous Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor: Number of Table Entries
Elseif: Permeability Function Option = { Tabular [Saturation] [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a], Char[b], Integer[c],*
For: Number of Table Entries
Aqueous Saturation[a], Aqueous Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor: Number of Table Entries
Endif:

Endfor: Number of Rock/Soil Types

Endif:

Endcard: Aqueous Relative Permeability Card

B.2.1 Aqueous Relative Permeability Card Examples

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
Sand,Mualem,,

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
20/30 Ottawa Sand,Mualem,0.56,

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
IJK Indexing,Mualem,,

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
IJK Indexing,Mualem,file:data_m.dat,

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
Sand,Fatt And Klikoff,

Extracted from a STOMP-W input file:
#R1 is a scaling group
~Aqueous Relative Permeability Card
R1,Mualem,,

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
Backfill,Mualem w/Polmann,,15.76,3.56,1.1e4,1.84e4,30.0,0.00371,57.710,30.819,
H2 Sand,Mualem w/Polmann,,-14.59,1.50,-7.2e4,6.55e4,50.0,0.00620,48.363,5.682,
H1 Gravelly Sand,Mualem w/Polmann,,14.85,1.94,2.6e4,2.504,30.0,0.00368,17.866,7.92,

Extracted from a STOMP-W input file:
~Aqueous Relative Permeability Card
Backfill,Anisotropy Mualem,,0.5,0.5,

Extracted from a STOMP-W input file:
~Aqueous Rel
SM-ML1,Burdine,,
SW1,Burdine,,
SP3,Burdine,,
SM-SP1,Burdine,,
SP2,Burdine,,
SP1,Burdine,,
US,Touma and Vauclin,1.0,2.0,

Extracted from a STOMP-WCMSE input file:
~Aqueous Relative Permeability Card
Medium,Free Corey,1.0,3.0,0.12,0.02,

B.3 Aqueous Species Card

Card Title[a] { ~Aqueous Species [Card] }
Format: *Char*[a],

If: Operational Mode = { **STOMP-W-R** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }

Number of Aqueous Species[a],
Aqueous Molecular Diffusion Coefficient[b], Units[c] (m),
[Activity Coefficient Model Option[d],]
{ Davies Equation | Pitzer Equation | Constant | B Dot }
Format: *Integer*[a], *Real*[b], *Char*[c], [*Char*[d],]

For: Number of Aqueous Species

Aqueous Species Name[a],
If: Activity Coefficient Model Option ≠ { Constant }
Species Charge[b],
Species Diameter[c], Units[d] (m),
Species Molecular Weight[e], Units[f] (gm/mol),
Format: *Char*[a], *Real*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f],
Elseif: Activity Coefficient Model Option = { Constant }
Species Activity Coefficient[b],
Format: *Char*[a], *Real*[b],
Endif:

Endfor: Number of Aqueous Species

Endcard: Aqueous Species Card

B.3.1 Aqueous Species Card Examples

Extracted from a STOMP-W-R input file:

```
~Aqueous Species Card
21,0.e-9,cm^2/s,Constant Activity,1.0,
CO2,0.,3.0,A,44.0098,kg/kmol,
CO3--,-2.,4.5,A,60.0092,kg/kmol,
Cl-,-1.,3.0,A,35.4527,kg/kmol,
Co(OH)2,0.,3.0,A,92.9479,kg/kmol,
Co(OH)3-,-1.,0.0,A,0.0,kg/kmol,
Co++,2.,6.0,A,58.9332,kg/kmol,
CoNta-,-1.,0.0,A,247.0638,kg/kmol,
CoNta2----,-4.,0.0,A,435.1944,kg/kmol,
CoOH+,1.,0.0,A,75.9405,kg/kmol,
CoOHNta--,-2.,0.0,A,264.0711,kg/kmol,
H+,1.,9.0,A,1.0079,kg/kmol,
H2Nta-,-1.,0.0,A,190.146,kg/kmol,
H3Nta,0.,0.0,A,191.1543,kg/kmol,
HCO3-,-1.,4.0,A,61.0171,kg/kmol,
HNta--,-2.,0.0,A,189.1385,kg/kmol,
NH3,0.,3.0,A,17.0306,kg/kmol,
NH4+,1.,2.5,A,18.0385,kg/kmol,
Nta---,-3.,0.0,A,188.1306,kg/kmol,
Na+,1.,4.0,A,22.9898,kg/kmol,
O2,0.,3.0,A,31.99880,kg/kmol,
OH-,-1.,3.5,A,17.0073,kg/kmol,
```

Extracted from a STOMP-WCS-R input file:

```
~Aqueous Species Card
10,1.e-9,cm^2/s,Constant Activity,1.0,
Ca++,2.0,6.0,A,40.0780,kg/kmol,
CO3--,-2.0,4.5,A,60.0092,kg/kmol,
H+,1.0,9.0,A,1.0079,kg/kmol,
OH-,-1.0,3.5,A,17.0073,kg/kmol,
CaCO3(aq),0.0,3.0,A,100.0872,kg/kmol,
CaHCO3+,1.0,4.0,A,101.0951,kg/kmol,
CaOH+,1.0,4.0,A,57.0853,kg/kmol,
HCO3-,-1.0,4.0,A,61.0171,kg/kmol,
H2CO3,0.0,4.0,A,62.0250,kg/kmol,
Ca(OH)2,0.0,4.0,A,74.0926,kg/kmol,
```

B.4 Boundary Conditions Card

Card Title[a] {~Boundary [Conditions Card] }
Format: *Char*[a]

Number of Boundary Condition Domains[a],
Format: *Integer*[a],

For: Number of Boundary Condition Domains

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-Sc**} { **STOMP-WAE** } { **STOMP-WAE** } { **STOMP-WAE-Sc** }
Boundary Surface Direction Option[a],
{ Bottom } { South } { West } { East } { North } { Top } { File }
Else:
Boundary Surface Direction Option[a],
{ Bottom } { South } { West } { East } { North } { Top }
Endif:

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-Sc**}
If: Surface Direction Option = { File }
File Name[b],
Note: File name contains x,y, and z node indices and surface indicator
(-3 for Bottom; -2 for South; -1 for West; 1 for East; 2 for North; 3 for Top).
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Saturated |Unit Gradient |
Free Gradient | Outflow |Hydraulic Gradient | Initial Condition |
[X-Y-Z] Seepage Face | Falling Head | Falling Pond }
For: Number of Solutes
Solute Transport Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
| Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
Inflow-Outflow Aqueous | Inflow-Outflow Volumetric }
Endfor: Number of Solutes
Format: *Char*[a], *Char*[b], *Char*[c], <*Char*[d],>
Else:
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Saturated |Unit Gradient
Free Gradient | Outflow |Hydraulic Gradient | Initial Condition |
[X-Y-Z] Seepage Face | Falling Head | Falling Pond }
For: Number of Solutes
Solute Transport Boundary Type Option[c],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
| Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
Inflow-Outflow Aqueous | Inflow-Outflow Volumetric }
Endfor: Number of Solutes
Format: *Char*[a], *Char*[b], <*Char*[c],>
Endif:

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-W-R** }
If: Surface Direction Option = { File }
File Name[b],
Note: File name contains x,y, and z node indices and surface indicator
(-3 for Bottom; -2 for South; -1 for West; 1 for East; 2 for North; 3 for Top).
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Saturated |Unit Gradient
Free Gradient | Outflow |Hydraulic Gradient | Initial Condition |
[X-Y-Z] Seepage Face | Falling Head | Falling Pond }
For: Number of Solutes
Solute Transport Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
| Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
Inflow-Outflow Aqueous | Inflow-Outflow Volumetric }
Endfor: Number of Solutes
Species Transport Boundary Type Option[e],
{ Aqueous Conc. | Zero Flux | Outflow | Initial Condition | Inflow Aqueous }
Format: *Char*[a], *Char*[b], *Char*[c], <*Char*[d],> *Char*[e]
Number of Reactive Species[a],
For: Number of Reactive Species
Species Name[b],
Endfor: Number of Reactive Species
Format: *Integer*[a], <*Char*[c],>
Else:
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Saturated |Unit Gradient
Free Gradient | Outflow |Hydraulic Gradient | Initial Condition |
[X-Y-Z] Seepage Face | Falling Head | Falling Pond }
For: Number of Solutes
Solute Transport Boundary Type Option[c],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
| Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
Inflow-Outflow Aqueous | Inflow-Outflow Volumetric }
Endfor: Number of Solutes
Species Transport Boundary Type Option[d],
{ Aqueous Conc. | Zero Flux | Outflow | Initial Condition | Inflow Aqueous }
Format: *Char*[a], *Char*[b], <*Char*[c],> *Char*[d],
Number of Reactive Species[a],
For: Number of Reactive Species
Species Name[b],
Endfor: Number of Reactive Species
Format: *Integer*[a], <*Char*[c],>
Endif:

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WA** }
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Saturated | Unit Gradient |
Hydraulic Gradient | Initial Condition | Seepage Face |
Gas-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Unit Gradient |
|Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
|Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c],* <*Char[d]*,>

Elseif: Operational Mode Option = { **STOMP-WAE** } { **STOMP-WAE-Sc** }
If: Surface Direction Option = { File }
File Name[b],
Note: File name contains x,y, and z node indices and surface indicator
(-3 for Bottom; -2 for South; -1 for West; 1 for East; 2 for North; 3 for Top).
Energy Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |
| Outflow | Initial Condition | Ground | Convective | Convective-Radiative }
Aqueous-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux | Saturated | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition | Seepage Face }
Gas-Phase Boundary Type Option[e],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[f],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],* <*Char[f]*,>
Else:
Energy Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
| Outflow | Initial Condition | Ground | Convective | Convective-Radiative | }
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Saturated | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition | Seepage Face }
Gas-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[e],

Boundary Conditions Card (cont'd)

{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d],* <*Char[e],*>
Endif:

Elseif: Operational Mode Option = { **STOMP-WAE-B** }
If: Surface Direction Option = { File }
File Name[b],
Note: File name contains x,y, and z node indices and surface indicator
(-3 for Bottom; -2 for South; -1 for West; 1 for East; 2 for North; 3 for Top).
Energy Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |
| Outflow | Initial Condition | Ground | Convective | Convective-Radiative |
Bare Shuttleworth-Wallace | Shuttleworth-Wallace }
Aqueous-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux | Saturated | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition | Seepage Face }
Gas-Phase Boundary Type Option[e],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[f],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],*<*Char[f],*>
Else:
Energy Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
| Outflow | Initial Condition | Ground | Convective | Convective-Radiative |
Bare Shuttleworth-Wallace | Shuttleworth-Wallace }
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Saturated | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition | Seepage Face }
Gas-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d],* <*Char[e],*>
Endif:

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WO** } { **STOMP-WOM** }
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition }
NAPL Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | NAPL Conc. |
Zero Flux | Outflow | Initial Condition |
Inflow Volumetric | Inflow Aqueous | Inflow NAPL }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c],* <*Char[d],*>

Elseif: Operational Mode Option Card = **{ STOMP-WOA } { STOMP-WOA-Sc }**
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Unit Gradient | Hydraulic Gradient | Initial Condition | Fluctuating Water Table }
Gas-Phase Boundary Type Option[c],
{ Dirichlet Outflow | Dirichlet | Neumann | Zero Flux |
Unit Gradient |Hydraulic Gradient | Initial Condition }
NAPL Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. | NAPL Conc. |
Zero Flux | Outflow | Initial Condition | Inflow Volumetric |
Inflow Aqueous | Inflow Gas | Inflow NAPL }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d],* <*Char[e],*>

Elseif: Operational Mode Option = **{ STOMP-WOAE }**
Energy Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Outflow | Initial Condition }
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |
Unit Gradient | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux |
Unit Gradient | Hydraulic Gradient | Initial Condition }
NAPL Boundary Type Option[e],
{ Dirichlet | Neumann | Zero Flux |
Unit Gradient | Hydraulic Gradient | Initial Condition }

Boundary Conditions Card (cont'd)

For: Number of Solutes
Solute Transport Boundary Type Option[f],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. | NAPL Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],* <*Char[f],*>

Elseif: Operational Mode Option = { **STOMP-WOD** }
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition }
NAPL Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Outflow |
Unit Gradient | Hydraulic Gradient | Initial Condition }
Dissolved Oil Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous }
For: Number of Solutes
Solute Transport Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | NAPL Conc. |
Zero Flux | Outflow | Initial Condition |
Inflow Volumetric | Inflow Aqueous }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d],* <*Char[e],*>

Elseif: Operational Mode Option = { **STOMP-WS** } { **STOMP-WS-Sc }**
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Saturated | Unit Gradient |
Free Gradient | Hydraulic Gradient | Initial Condition }
Salt Boundary Type Option[c],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
[Inflow] Relative Saturation }
For: Number of Solutes
Solute Transport Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c],* <*Char[d],*>

Elseif: Operational Mode Option = { **STOMP-WAS** }
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux | Unit Gradient |
Saturated | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |

Hydraulic Gradient | Initial Condition }
Salt Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux | Gas Conc. |
Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
[Inflow] Relative Saturation }
For: Number of Solutes
Solute Transport Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d],* <*Char[e],*>

Elseif: Operational Mode Option = { **STOMP-WASE** }
Energy Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Outflow | Initial Condition | Ground | Convective | Convective Radiative }
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux | Saturated |
Unit Gradient | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
Salt Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux | Gas Conc. |
Outflow | Initial Condition | Inflow Volumetric | Inflow Aqueous |
[Inflow] Relative Saturation }
For: Number of Solutes
Solute Transport Boundary Type Option[f],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],* <*Char[f],*>

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-Sc** }
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Saturated | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
Salt Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
Outflow | Initial Condition |
Inflow Volumetric Conc. | Inflow Aqueous Conc. |
Inflow Mass Fraction | Inflow Relative Saturation |
Aqueous Relative Saturation | Aqueous Mass Fraction }

Boundary Conditions Card (cont'd)

For: Number of Solutes
Solute Transport Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char*[a], *Char*[b], *Char*[c], *Char*[d], <*Char*[e],>

Elseif: Operational Mode Option = { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }
Aqueous-Phase Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Saturated | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
Salt Boundary Type Option[d],
{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
Outflow | Initial Condition |
Inflow Volumetric Conc. | Inflow Aqueous Conc. |
Inflow Mass Fraction | Inflow Relative Saturation |
Aqueous Relative Saturation | Aqueous Mass Fraction }
For: Number of Solutes
Solute Transport Boundary Type Option[e],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char*[a], *Char*[b], *Char*[c], *Char*[d], <*Char*[e],>
Species Transport Boundary Type Option[d],
{ Aqueous Conc. | Zero Flux | Outflow | Initial Condition | Inflow Aqueous }
Format: *Char*[a], *Char*[b], <*Char*[c],> *Char*[d],
Number of Reactive Species[a],
For: Number of Reactive Species
Species Name[b],
Endfor: Number of Reactive Species
Format: *Integer*[a], <*Char*[c],>

Elseif: Operational Mode Option = { **STOMP-WCSE** }
Energy Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Outflow | Initial Condition | Ground | Convective | Convective Radiative }
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet | Neumann | Zero Flux |
Saturated | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[d],
{ Dirichlet | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
Salt Boundary Type Option[e],

Boundary Conditions Card (cont'd)

{ Volumetric Conc. | Aqueous Conc. | Zero Flux |
Outflow | Initial Condition |
Inflow Volumetric Conc. | Inflow Aqueous Conc. |
Inflow Mass Fraction | Inflow Relative Saturation |
Aqueous Relative Saturation | Aqueous Mass Fraction }
For: Number of Solutes
Solute Transport Boundary Type Option[f],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],* <*Char[f],*>

Elseif: Operational Mode Option = { **STOMP-WCMSE** }
Note: Salt Equation is currently not solved.
Energy Boundary Type Option[b],
{ Dirichlet | Neumann | Zero Flux |
Outflow | Initial Condition | Ground | Convective | Convective Radiative }
Aqueous-Phase Boundary Type Option[c],
{ Dirichlet [Outflow] [Inflow] | Neumann | Zero Flux |
Unit Gradient | Hydraulic Gradient | Initial Condition }
Gas-Phase Boundary Type Option[d],
{ Dirichlet [Outflow] [Inflow] | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
Liquid-CO_2 Boundary Type Option[e],
{ Dirichlet [Outflow] [Inflow] | Neumann | Zero Flux |
Hydraulic Gradient | Initial Condition }
For: Number of Solutes
Solute Transport Boundary Type Option[f],
{ Volumetric Conc. | Aqueous Conc. | Gas Conc. |
Zero Flux | Outflow | Initial Condition }
Endfor: Number of Solutes
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],* <*Char[f],*>
Endif:

If: Boundary Surface Direction Option[a],
{ Bottom } { South } { West } { East } { North } { Top }
I-Start Index[a], I-End Index[b], J-Start Index[c], J-End Index[d],
K-Start Index[e], K-End Index[f], Number of Boundary Times[g],
Format: *Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g],*
Elseif: Boundary Surface Direction Option[a], { File }
Number of Boundary Times[a],
Format: *Integer[a],*
Endif:

Boundary Conditions Card (cont'd)

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-Sc** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous Boundary Type Option =
{ Dirichlet } { Zero Flux } { Outflow }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option =
{ Hydraulic Gradient } { [X-Y-Z] Seepage Face }
Base Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option =
{ Falling Head } { Falling Pond }
Ponding Height[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
If: Aqueous Boundary Type Option = { X-Y-Z Seepage Face }
X-Direction Gradient[e], Units[f],
Y-Direction Gradient[f], Units[g],
Z-Direction Gradient[h], Units[i],
For: Number of Solutes
If: Solute Transport Boundary Type Option =
{ [Inflow] Volumetric Conc. }
Solute Volumetric Conc[j], Units[k] (1/m^3),
Elseif: Solute Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[j], Units[k] (1/m^3),
Else:
Null[j], Null[k],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
< *Real[j], Char[k],* >
Else:
For: Number of Solutes
If: Solute Transport Boundary Type Option =
{ [Inflow] Volumetric Conc. }
Solute Volumetric Conc[e], Units[f] (1/m^3),
Elseif: Solute Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[e], Units[f] (1/m^3),
Else:
Null[e], Null[f],
Endif:
Endfor: Number of Solutes

Boundary Conditions Card (cont'd)

Format: *Real[a], Char[b], Real[c], Char[d],* < *Real[e], Char[f],* >
Endif
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-W-R** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous Boundary Type Option =
{ Dirichlet } { Zero Flux } { Outflow }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option =
{ Hydraulic Gradient } { Seepage Face }
Base Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option =
{ Falling Head } { Falling Pond }
Ponding Height[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
If: Aqueous Boundary Type Option = { X-Y-Z Seepage Face }
X-Direction Gradient[e], Units[f],
Y-Direction Gradient[f], Units[g],
Z-Direction Gradient[h], Units[i],
For: Number of Solutes
If: Solute Transport Boundary Type Option =
{ [Inflow] Volumetric Conc. }
Solute Volumetric Conc[j], Units[k] (1/m^3),
Elseif: Solute Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[j], Units[k] (1/m^3),
Else:
Null[j], Null[k],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
< *Real[j], Char[k],* >
For: Number of Species
If: Species Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Species Aqueous-Phase Conc[l], Units[m] (mol/m^3),
Else:
Null[l], Null[m],
Endif:
Endfor: Number of Species
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
< *Real[j], Char[k],* >, < *Real[l], Char[m],* >
Else:
For: Number of Solutes

Boundary Conditions Card (cont'd)

If: Solute Transport Boundary Type Option =
{ [Inflow] Volumetric Conc. }
Solute Volumetric Conc[e], Units[f] (1/m^3),
Elseif: Solute Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[e], Units[f] (1/m^3),
Else:
Null[e], Null[f],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d],* < *Real[e], Char[f],* >
For: Number of Species
If: Species Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Species Aqueous-Phase Conc[g], Units[h] (mol/m^3),
Else:
Null[g], Null[h],
Endif:
Endfor: Number of Species
Format: *Real[a], Char[b], Real[c], Char[d],* < *Real[e], Char[f],* >, < *Real[g], Char[h],* >
Endif

Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WA** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
{ Seepage Face }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
Aqueous Dissolved-Air Relative Saturation[e],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[f], Units[g] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[f], Units[g] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Gas Aqueous Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
Water-Vapor Relative Humidity[h],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[i], Units[j] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[i], Units[j] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas-Phase Volumetric Conc[i], Units[j] (1/m^3),
Else:
Null[i], Null[j],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g], Real[h],* < *Real[i], Char[j],* >
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WAE** } { **STOMP-WAE-Sc** }
If: Boundary Times and Conditions are Read in from File:
File Read Indication[a], { File }
File Name[b],
Format: Char[a], Char[b],
Else:
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),

If: Energy Boundary Type Option = { Shuttleworth-Wallace }
Volumetric Aqueous Flux[c], Units[d] (C),
For: Number of Plant Varietals
Leaf Area Index[e],
Endfor:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[f], Units[g] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[f], Units[g] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
{ Seepage Face }
Base Aqueous Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
Aqueous Dissolved-Air Relative Saturation[h],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[i], Units[j] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[i], Units[j] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[i], Units[j] (Pa),
Else:
Null[i], Null[j],
Endif:
Water-Vapor Relative Humidity[k],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[l], Units[m] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[l], Units[m] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[l], Units[m] (1/m^3),
Else:
Null[g], Null[h],
Endif:
Endfor: Number of Solutes

Format: *Real[a], Char[b], Real[c], Char[d],< Real[e],> Real[f], Char[g], Real[h], Real[i], Char[j], Real[k], < Real[l], Char[m], >*

Elseif: Energy Boundary Type Option = { Convective } { Convective-Radiative }
If: Energy Boundary Type Option = { Convective }
Conv. Temp.[c], Units[d] (C), Conv. Heat Transfer Coeff.[e], Units[r] (kg/C s^3),
Elseif: Energy Boundary Type Option = { Convective-Radiative }
Conv. Temperature[c], Units[d] (C), Rad. Temperature[e], Units[r] (C),
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[g], Units[h] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[g], Units[h] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient } { Seepage Face }
Base Aqueous Pressure[g], Units[h] (Pa),
Else:
Null[g], Null[h],
Endif:
Aqueous Dissolved-Air Relative Saturation[i],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[j], Units[k] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[j], Units[k] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[j], Units[k] (Pa),
Else:
Null[j], Null[k],
Endif:
Water-Vapor Relative Humidity[l],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[m], Units[n] (1/m^3),
Else:
Null[m], Null[n],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k], Real[l], < Real[m], Char[n], >*
Else:

Boundary Conditions Card (cont'd)

If: Energy Boundary Type Option = { Dirichlet }
Temperature[c], Units[d] (C),
Elseif: Energy Boundary Type Option = { Neumann }
Energy Flux[c], Units[d] (W/m^2),
Elseif: Energy Boundary Type Option = { Ground }
Air Temperature[c], Units[d] (C),
Else:
Null[c], Null[d],
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[e], Units[f] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[e], Units[f] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
{ Seepage Face }
Base Aqueous Pressure[e], Units[f] (Pa),
Else:
Null[e], Null[f],
Endif:
Aqueous Dissolved-Air Relative Saturation[g],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[h], Units[i] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[h], Units[i] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[i], Units[i] (Pa),
Else:
Null[h], Null[i],
Endif:
Water-Vapor Relative Humidity[j],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[k], Units[l] (1/m^3),
Else:
Null[k], Null[l],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i], Real[j],* < *Real[k], Char[l],* >
Endif:
Endfor: Number of Boundary Times
Endif:

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WAE-B** }
If: Boundary Times and Conditions are Read in from File:
File Read Indication[a], { File }
File Name[b],
Format: Char[a], Char[b],
Else:
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Energy Boundary Type Option = { Bare Shuttleworth-Wallace }
Volumetric Aqueous Flux[c], Units[d] (C),
Elseif: Energy Boundary Type Option = { Shuttleworth-Wallace }
Volumetric Aqueous Flux[c], Units[d] (C),
For: Number of Plant Species
Leaf Area Index[e],
Plant Area Index[f],
Endfor:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[f], Units[g] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[f], Units[g] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
{ Seepage Face }
Base Aqueous Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
Aqueous Dissolved-Air Relative Saturation[h],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[i], Units[j] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[i], Units[j] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[i], Units[j] (Pa),
Else:
Null[i], Null[j],
Endif:
Water-Vapor Relative Humidity[k],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[l], Units[m] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[l], Units[m] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[l], Units[m] (1/m^3),
Else:
Null[g], Null[h],

Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d],< Real[e],> Real[f], Char[g], Real[h], Real[i], Char[j], Real[k], < Real[l], Char[m], >*
Elseif: Energy Boundary Type Option = { Convective } { Convective-Radiative }
If: Energy Boundary Type Option = { Convective }
Conv. Temperature[c], Units[d] (C), Conv. Heat Transfer Coeff.[e], Units[r] (kg/C s^3),
Elseif: Energy Boundary Type Option = { Convective-Radiative }
Conv. Temperature[c], Units[d] (C), Rad. Temperature[e], Units[r] (C),
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[g], Units[h] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[g], Units[h] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient } { Seepage Face }
Base Aqueous Pressure[g], Units[h] (Pa),
Else:
Null[g], Null[h],
Endif:
Aqueous Dissolved-Air Relative Saturation[i],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[j], Units[k] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[j], Units[k] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[j], Units[k] (Pa),
Else:
Null[j], Null[k],
Endif:

For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[m], Units[n] (1/m^3),
Else:
Null[m], Null[n],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k], Real[l], < Real[m], Char[n], >*

Boundary Conditions Card (cont'd)

Else:
If: Energy Boundary Type Option = { Dirichlet }
Temperature[c], Units[d] (C),
Elseif: Energy Boundary Type Option = { Neumann }
Energy Flux[c], Units[d] (W/m^2),
Elseif: Energy Boundary Type Option = { Ground }
Air Temperature[c], Units[d] (C),
Else:
Null[c], Null[d],
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[e], Units[f] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[e], Units[f] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
{ Seepage Face }
Base Aqueous Pressure[e], Units[f] (Pa),
Else:
Null[e], Null[f],
Endif:
Aqueous Dissolved-Air Relative Saturation[g],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[h], Units[i] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[h], Units[i] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[i], Units[i] (Pa),
Else:
Null[h], Null[i],
Endif:
Water-Vapor Relative Humidity[j],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[k], Units[l] (1/m^3),
Else:
Null[k], Null[l],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i], Real[j],* < *Real[k], Char[l],* >
Endif:
Endfor: Number of Boundary Times
Endif:

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WO** } { **STOMP-WOM** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
Aqueous Dissolved Oil Mass Fraction[e],
If: NAPL Boundary Type Option = { Dirichlet } { Zero Flux }
NAPL Pressure[f], Units[g] (Pa),
Elseif: NAPL Boundary Type Option = { Neumann }
NAPL Volumetric Flux[f], Units[g] (m/s),
Elseif: NAPL Boundary Type Option = { Hydraulic Gradient }
Base NAPL Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { [Inflow] Volumetric Conc. }
Solute Volumetric Conc[h], Units[i] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[h], Units[i] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] NAPL Conc. }
Solute NAPL Volumetric Conc[h], Units[i] (1/m^3),
Else:
Null[h], Null[i],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],* < *Real[h], Char[i],* >
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc }**
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option =
{ Hydraulic Gradient } { Fluctuating Water Table }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
Aqueous Dissolved Air Mass Fraction[e],
Aqueous Dissolved Oil Mass Fraction[f],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[g], Units[h] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[g], Units[h] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[g], Units[h] (Pa),
Else:
Null[g], Null[h],
Endif:
Water Vapor Relative Humidity[i],
Oil Vapor Relative Humidity[j],
If: NAPL Boundary Type Option = { Dirichlet } { Zero Flux }
NAPL Pressure[k], Units[l] (Pa),
Elseif: NAPL Boundary Type Option = { Neumann }
NAPL Volumetric Flux[k], Units[l] (m/s),
Elseif: NAPL Boundary Type Option = { Hydraulic Gradient }
Base NAPL Pressure[k], Units[l] (Pa),
Else:
Null[k], Null[l],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { [Inflow] Volumetric Conc. }
Solute Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] Gas Conc. }
Solute Gas-Phase Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] NAPL Conc. }
Solute NAPL Volumetric Conc[m], Units[n] (1/m^3),
Else:

Boundary Conditions Card (cont'd)

Null[m], Null[n],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Char[l],* < *Real[m], Char[n],* >
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WOAE }**
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Energy Boundary Type Option = { Dirichlet }
Temperature[c], Units[d] (C)
Elseif: Energy Boundary Type Option = { Neumann }
Energy Flux[c], Units[d] (W/m^2)
Else:
Null[c], Null[d],
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[e], Units[f] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[e], Units[f] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[e], Units[f] (Pa),
Else:
Null[e], Null[f],
Endif:
Aqueous Dissolved Air Mass Fraction[g],
Aqueous Dissolved Oil Mass Fraction[h],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[i], Units[j] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[i], Units[j] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[i], Units[j] (Pa),
Else:
Null[i], Null[j],
Endif:
Water Vapor Relative Humidity[k],
Oil Vapor Relative Humidity[l],
If: NAPL Boundary Type Option = { Dirichlet } { Zero Flux }
NAPL Pressure[m], Units[n] (Pa),
Elseif: NAPL Boundary Type Option = { Neumann }
NAPL Volumetric Flux[m], Units[n] (m/s),
Elseif: NAPL Boundary Type Option = { Hydraulic Gradient }
Base NAPL Pressure[m], Units[n] (Pa),
Else:
Null[m], Null[n],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[o], Units[p] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }

Solute Gas-Phase Volumetric Conc[o], Units[p] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { NAPL Conc. }
Solute NAPL Volumetric Conc[o], Units[p] (1/m^3),
Else:
Null[o], Null[p],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Real[i], Char[j], Real[k], Real[l], Real[m], Char[n],* < *Real[o], Char[p],* >
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WOD** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type = { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[c], Units[d] (Pa),
Else: Null[c], Null[d],
Endif:
If: NAPL Boundary Type Option = { Dirichlet } { Zero Flux }
NAPL Pressure[e], Units[f] (Pa),
Elseif: NAPL Boundary Type Option = { Neumann }
NAPL Volumetric Flux[e], Units[f] (m/s),
Elseif: NAPL Boundary Type Option = { Hydraulic Gradient }
Base NAPL Pressure[e], Units[f] (Pa),
Else: Null[e], Null[f],
Endif:
If: Dissolved Oil Boundary Type Option = { [Inflow] Volumetric Conc. }
Dissolved Oil Volumetric Conc[g], Units[h] (kg/m^3),
Elseif: Dissolved Oil Boundary Type Option =
{ Aqueous Conc. } { Inflow Aqueous }
Dissolved Oil Aqueous-Phase Volumetric Conc[g], Units[h] (kg/m^3),
Else: Null[g], Null[h],
Endif:
For: Number of Solutes:
If: Solute Transport Boundary Type Option = { [Inflow] Volumetric Conc. }
Solute Volumetric Conc[i], Units[j] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[i], Units[j] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { NAPL Conc. }
Solute NAPL Volumetric Conc[i], Units[j] (1/m^3),
Else: Null[i], Null[j],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],* <*Real[i], Char[j],*>
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WS** } { **STOMP-WS-Sc** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
If: Salt Boundary Type Option = { Inflow Volumetric Conc. } { Volumetric Conc. }
Salt Volumetric Conc[e], Units[f] (kg/m^3),
Elseif: Salt Boundary Type Option =
{ Aqueous Conc. } { Inflow Aqueous }
Salt Aqueous-Phase Volumetric Conc[e], Units[f] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Relative Saturation }
Salt Relative Saturation[e], Null[f],
Else:
Null[e], Null[f],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { [Inflow] Volumetric Conc. }
Solute Volumetric Conc[g], Units[h] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { [Inflow] Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[g], Units[h] (1/m^3),
Else:
Null[g], Null[h],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],* <*Real[g], Char[h],*>
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WAS** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
Aqueous Dissolved-Air Relative Saturation[e],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[f], Units[g] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[f], Units[g] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
Water-Vapor Relative Humidity[h],
If: Salt Boundary Type Option = { Volumetric Conc. } { Inflow Volumetric }
Salt Volumetric Conc[i], Units[j] (kg/m^3),
Elseif: Salt Boundary Type Option = { Aqueous Conc. } { Inflow Aqueous }
Salt Aqueous-Phase Volumetric Conc[i], Units[j] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Relative Saturation }
Salt Relative Saturation[i], Null[j],
Else:
Null[i], Null[j],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[k], Units[l] (1/m^3),
Else:
Null[k], Null[l],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g], Real[h], Real[i], Char[j],*
< Real[k], Char[l], >
Endfor: Number of Boundary Times

Elseif: Operational Mode Option = { **STOMP-WASE** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Energy Boundary Type Option = { Convective } { Convective-Radiative }
If: Energy Boundary Type Option = { Convective }
Conv. Temperature[c], Units[d] (C), Conv. Heat Transfer Coeff.[e], Units[r] (kg/C s^3),
Elseif: Energy Boundary Type Option = { Convective-Radiative }
Conv. Temperature[c], Units[d] (C), Rad. Temperature[e], Units[r] (C),
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[g], Units[h] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[g], Units[h] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[g], Units[h] (Pa),
Else:
Null[g], Null[h],
Endif:
Aqueous Dissolved-Air Relative Saturation[i],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[j], Units[k] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[j], Units[k] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[j], Units[k] (Pa),
Else:
Null[j], Null[k],
Endif:
Water-Vapor Relative Humidity[l],
If: Salt Boundary Type Option = { Volumetric Conc. }
Salt Volumetric Conc[m], Units[n] (kg/m^3),
Elseif: Salt Boundary Type Option = { Aqueous Conc. }
Salt Aqueous-Phase Volumetric Conc[m], Units[n] (kg/m^3),
Elseif: Salt Boundary Type Option = { Inflow Volumetric }
Salt Volumetric Conc[m], Units[n] (kg/m^3),
Elseif: Salt Boundary Type Option = { Inflow Aqueous }
Salt Aqueous-Phase Volumetric Conc[m], Units[n] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Relative Saturation }
Salt Relative Saturation[m], Null[n],
Else:
Null[m], Null[n],
Endif:
For: Number of Solutes

If: Solute Transport Boundary Type Option = { Volumetric Conc. }

Solute Volumetric Conc[o], Units[p] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[o], Units[p] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[o], Units[p] (1/m^3),
Else:
Null[o], Null[p],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k], Real[l], Real[m], Char[n],* < *Real[o], Char[p],* >
Endif:

Else:
If: Energy Boundary Type Option = { Dirichlet }
Temperature[c], Units[d] (C),
Elseif: Energy Boundary Type Option = { Neumann }
Energy Flux[c], Units[d] (W/m^2),
Elseif: Energy Boundary Type Option = { Ground }
Air Temperature[c], Units[d] (C),
Else:
Null[c], Null[d],
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[e], Units[f] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[e], Units[f] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[e], Units[f] (Pa),
Else:
Null[e], Null[f],
Endif:
Aqueous Dissolved-Air Relative Saturation[g],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[h], Units[i] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[h], Units[i] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[h], Units[i] (Pa),
Else:
Null[h], Null[i],
Endif:
Water-Vapor Relative Humidity[j]
If: Salt Boundary Type Option = { Volumetric Conc. }
Salt Volumetric Conc[k], Units[l] (kg/m^3),
Elseif: Salt Boundary Type Option = { Aqueous Conc. }

Salt Aqueous-Phase Volumetric Conc[k], Units[l] (kg/m^3),
Elseif: Salt Boundary Type Option = { Inflow Volumetric }
Salt Volumetric Conc[k], Units[l] (kg/m^3),
Elseif: Salt Boundary Type Option = { Inflow Aqueous }
Salt Aqueous-Phase Volumetric Conc[k], Units[l] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Relative Saturation }
Salt Relative Saturation[k], Null[l],
Else:
Null[k], Null[l],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[m], Units[n] (1/m^3),
Else:
Null[m], Null[n],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i], Real[j], Real[k], Char[l],* < *Real[m], Char[n],* >
Endif:
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-Sc** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
Aqueous Dissolved- CO_2-Relative Saturation[e],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[f], Units[g] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[f], Units[g] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
Water-Vapor Relative Humidity[h],
If: Salt Boundary Type Option = { [Inflow] Volumetric Conc. }
Salt Volumetric Conc[i], Units[j] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Aqueous Conc. }
Salt Aqueous-Phase Volumetric Conc[i], Units[j] (kg/m^3),
Elseif: Salt Boundary Type Option =
{ Inflow Aqueous Saturation } { Inflow Relative Saturation }
Salt Relative Saturation[i], Null[j],
Elseif: Salt Boundary Type Option =
{ Inflow Mass Fraction } { Aqueous Mass Fraction }
Salt Mass Fraction[i], Null[j],
Else:
Null[i], Null[j],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[k], Units[l] (1/m^3),
Else:
Null[k], Null[l],

Boundary Conditions Card (cont'd)

Endif:
Endfor: Number of Solutes
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Real*[f], *Char*[g], *Real*[h], *Real*[i], *Char*[j], < *Real*[k], *Char*[l], >
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[c], Units[d] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[c], Units[d] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[c], Units[d] (Pa),
Else:
Null[c], Null[d],
Endif:
Aqueous Dissolved- CO_2-Relative Saturation[e],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[f], Units[g] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[f], Units[g] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[f], Units[g] (Pa),
Else:
Null[f], Null[g],
Endif:
Water-Vapor Relative Humidity[h],
If: Salt Boundary Type Option = { [Inflow] Volumetric Conc. }
Salt Volumetric Conc[i], Units[j] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Aqueous Conc. }
Salt Aqueous-Phase Volumetric Conc[i], Units[j] (kg/m^3),
Elseif: Salt Boundary Type Option =
{ Inflow Aqueous Saturation } { Inflow Relative Saturation }
Salt Relative Saturation[i], Null[j],
Elseif: Salt Boundary Type Option =
{ Inflow Mass Fraction } { Aqueous Mass Fraction }
Salt Mass Fraction[i], Null[j],
Else:
Null[i], Null[j],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[k], Units[l] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[k], Units[l] (1/m^3),
Else:
Null[k], Null[l],

Boundary Conditions Card (cont'd)

Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g], Real[h], Real[i], Char[j],* < *Real[k], Char[l],* >
For: Number of Species
If: Species Transport Boundary Type Option =
{ [Inflow] Aqueous Conc. }
Species Aqueous-Phase Conc[m], Units[n] (mol/m^3),
Else:
Null[m], Null[n],
Endif:
Endfor: Number of Species
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g], Real[h], Real[i], Char[j],* < *Real[k], Char[l],* >, < *Real[m], Char[n],* >

Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCSE** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),
If: Energy Boundary Type Option = { Convective } { Convective-Radiative }
If: Energy Boundary Type Option = { Convective }
Conv. Temperature[c], Units[d] (C), Conv. Heat Transfer Coeff.[e], Units[f] (kg/C s^3),
Elseif: Energy Boundary Type Option = { Convective-Radiative }
Conv. Temperature[c], Units[d] (C), Rad. Temperature[e], Units[r] (C),
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[g], Units[h] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[g], Units[h] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[g], Units[h] (Pa),
Else:
Null[g], Null[h],
Endif:
Aqueous Dissolved-CO_2 Relative Saturation[i],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[j], Units[k] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[j], Units[k] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[j], Units[k] (Pa),
Else:
Null[j], Null[k],
Endif:
Water-Vapor Relative Humidity[l],
If: Salt Boundary Type Option = { [Inflow] Volumetric Conc. }
Salt Volumetric Conc[m], Units[n] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Aqueous Conc. }
Salt Aqueous-Phase Volumetric Conc[m], Units[n] (kg/m^3),
Elseif: Salt Boundary Type Option =
{ Inflow Aqueous Saturation } { Inflow Relative Saturation }
Salt Relative Saturation[m], Null[n],
Elseif: Salt Boundary Type Option =
{ Inflow Mass Fraction } { Aqueous Mass Fraction }
Salt Mass Fraction[m], Null[n],
Else:
Null[m], Null[n],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[o], Units[p] (1/m^3),

Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[o], Units[p] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[o], Units[p] (1/m^3),
Else:
Null[o], Null[p],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k], Real[l], Real[m], Char[n],* < *Real[o], Char[p],* >
Endif:

Else:
If: Energy Boundary Type Option = { Dirichlet }
Temperature[c], Units[d] (C),
Elseif: Energy Boundary Type Option = { Neumann }
Energy Flux[c], Units[d] (W/m^2),
Elseif: Energy Boundary Type Option = { Ground }
Air Temperature[c], Units[d] (C),
Else:
Null[c], Null[d],
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[e], Units[f] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[e], Units[f] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[e], Units[f] (Pa),
Else:
Null[e], Null[f],
Endif:
Aqueous Dissolved-Air Relative Saturation[g],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[h], Units[i] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[h], Units[i] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[h], Units[i] (Pa),
Else:
Null[h], Null[i],
Endif:
Water-Vapor Relative Humidity[j]
If: Salt Boundary Type Option = { [Inflow] Volumetric Conc. }
Salt Volumetric Conc[k], Units[l] (kg/m^3),
Elseif: Salt Boundary Type Option = { [Inflow] Aqueous Conc. }

Salt Aqueous-Phase Volumetric Conc[k], Units[l] (kg/m^3),
Elseif: Salt Boundary Type Option =
{ Inflow Aqueous Saturation } { Inflow Relative Saturation }
Salt Relative Saturation[k], Null[l],
Elseif: Salt Boundary Type Option =
{ Inflow Mass Fraction } { Aqueous Mass Fraction }
Salt Mass Fraction[k], Null[l],
Else:
Null[k], Null[l],
Endif:
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[m], Units[n] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[m], Units[n] (1/m^3),
Else:
Null[m], Null[n],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i], Real[j], Real[k], Char[l],* < *Real[m], Char[n],* >
Endif:
Endfor: Number of Boundary Times

Boundary Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCMSE** }
For: Number of Boundary Times
Boundary Time[a], Units[b] (s),

If: Energy Boundary Type Option = { Convective } { Convective-Radiative }
If: Energy Boundary Type Option = { Convective }
Conv. Temperature[c], Units[d] (C), Conv. Heat Trans. Coeff.[e], Units[f] (kg/C s^3),
Elseif: Energy Boundary Type Option = { Convective-Radiative }
Conv. Temperature[c], Units[d] (C), Rad. Temperature[e], Units[f] (C),
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[g], Units[h] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[g], Units[h] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[g], Units[h] (Pa),
Else:
Null[g], Null[h],
Endif:
Aqueous Dissolved-CO_2 Relative Saturation[i],
Aqueous Dissolved-CH_4 Relative Saturation[j],
Aqueous Dissolved-Salt Relative Saturation[k],
CO_2 Slurry Aqueous Volume Fraction[l],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[m], Units[n] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[m], Units[n] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[m], Units[n] (Pa),
Else:
Null[m], Null[n],
Endif:
Water-Vapor Relative Humidity[o],
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[p], Units[q] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[p], Units[q] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[p], Units[q] (1/m^3),
Else:
Null[p], Null[q],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l], Real[m] ,Char[n], Real[o],* < *Real[p], Char[q],* >

Else:
If: Energy Boundary Type Option = { Dirichlet }
Temperature[c], Units[d] (C),
Elseif: Energy Boundary Type Option = { Neumann }
Energy Flux[c], Units[d] (W/m^2),
Elseif: Energy Boundary Type Option = { Ground }
Air Temperature[c], Units[d] (C),
Else:
Null[c], Null[d],
Endif:
If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Aqueous Pressure[e], Units[f] (Pa),
Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
Aqueous Volumetric Flux[e], Units[f] (m/s),
Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
Base Aqueous Pressure[e], Units[f] (Pa),
Else:
Null[e], Null[f],
Endif:
Aqueous Dissolved-CO_2 Relative Saturation[g],
Aqueous Dissolved-CH_4 Relative Saturation[h],
Aqueous Dissolved-Salt Relative Saturation[i],
CO_2 Slurry Aqueous Volume Fraction[j],
If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
Gas Pressure[k], Units[l] (Pa),
Elseif: Gas-Phase Boundary Type Option = { Neumann }
Gas Volumetric Flux[k], Units[l] (m/s),
Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
Base Gas Pressure[k], Units[l] (Pa),
Else:
Null[k], Null[l],
Endif:
Water-Vapor Relative Humidity[m]
For: Number of Solutes
If: Solute Transport Boundary Type Option = { Volumetric Conc. }
Solute Volumetric Conc[n], Units[o] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Aqueous Conc. }
Solute Aqueous-Phase Volumetric Conc[n], Units[o] (1/m^3),
Elseif: Solute Transport Boundary Type Option = { Gas Conc. }
Solute Gas Volumetric Conc[n], Units[o] (1/m^3),
Else:
Null[n], Null[o],
Endif:
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Real[i], Real[j], Real[k], Char[l],Real[m],* < *Real[n], Char[o],* >

Boundary Conditions Card (cont'd)

Endif:
Endfor: Number of Boundary Times
Endif:
Endfor: Number of Boundary Condition Domains

Endcard: Boundary Conditions Card

B.4.1 Boundary Conditions Card Examples

Extracted from a STOMP-W input file:

```
#-----------------------------------------------------
~Boundary Conditions Card
#-----------------------------------------------------
2,
East,Hydraulic Gradient,Aqueous Conc,
50,50,1,1,1,113,1,
0,day,183254,Pa,1.,1/ft^3,
West,Neumann,Outflow,
3,3,1,1,14,33,4,
0,min,-0.00021209,ft/sec,,,
37.4675,min,-0.00021209,ft/sec,,,
74.9232,min,-0.000200628,ft/sec,,,
112.379,min,-0.000201526,ft/sec,,,
```

Extracted from a STOMP-W input file:

```
#----------------------
~Boundary Condition Card
#----------------------
1,
file,eastbnd_rev4.inp,X-Y-Z Seepage Face,inflow-outflow, inflow-outflow,
2,
1991.00000,yr, 361987.35,Pa,0.00,1/m,5.000000E-01,1/m,-9.793519E+03,1/m,0.0,pCi/L,0.0,,
1991.08493,yr, 361987.35,Pa,0.00,1/m,5.000000E-01,1/m,-9.793519E+03,1/m,0.0,pCi/L,0.0,,
```

Extracted from a STOMP-W input file:

```
#13
~Boundary Conditions Card
6,
top,neumann,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,
  1,  4,  1,  1, 95, 95,6,
2000,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
12032,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
top,neumann,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,
  5,174,  1,  1, 94, 94,6,
2000,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
12032,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
top,neumann,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,
175,177,  1,  1, 95, 95,6,
2000,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
```

2032,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
12032,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
top,neumann,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,zero flux,
178,180, 1, 1, 96, 96,6,
2000,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-100,mm/yr,,,,,,,,,,,,,,,,,,
2032,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-0.5,mm/yr,,,,,,,,,,,,,,,,,,
2532,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
12032,yr,-1.0,mm/yr,,,,,,,,,,,,,,,,,,
west,hydraulic gradient,outflow,outflow,outflow,outflow,outflow,outflow,outflow,outflow,
1, 1, 1, 1, 2, 16,1,
0,day,233546.3,Pa,,,,,,,,,,,,,,,,,,,,
east,hydraulic gradient,outflow,outflow,outflow,outflow,outflow,outflow,outflow,outflow,
180,180, 1, 1, 1, 16,1,
0,day,243322.2,Pa,,,,,,,,,,,,,,,,,,,,

Extracted from a STOMP-W-R input file:

~Boundary Conditions Card
2,
West,Aqueous Neumann,Species Aqueous Conc.,
8,H+,HCO3-,NO3-,Na+,Ca++,Cl-,SO4--,UO2++,
1,1,1,1,1,1,1,
0,hr,1.44,m/day,
#River w/ Uranium
-1.93485E-03,mol/liter,9.18e-4,mol/liter,8.55e-6,mol/liter,
1.e-4,mol/liter,3.74e-4,mol/liter,3.1e-5,mol/liter,
7.08e-5,mol/liter,9.96e-4,mol/liter,
East,Aqueous Dirichlet,Species Outflow,
0,
100,100,1,1,1,1,1,
0,day,102064.81,Pa,

Extracted from a STOMP-W-R input file:

~Boundary Conditions Card
2,
West,Aqueous Neumann,Species Aqueous Conc.,
7,CO2,Na+,Nta---,H+,Co++,Cl-,O2,
1,1,1,1,1,1,4,
0,hr,0.4,m/hr,
4.9e-7,mol/liter,1.0e-3,mol/liter,
5.23e-6,mol/liter,1.277095e-6,mol/liter,
5.23e-6,mol/liter,1.0e-3,mol/liter,
3.125e-5,mol/liter,
20,hr,0.4,m/hr,
4.9e-7,mol/liter,1.0e-3,mol/liter,

5.23e-6,mol/liter,1.277095e-6,mol/liter,
5.23e-6,mol/liter,1.0e-3,mol/liter,
3.125e-5,mol/liter,
20,hr,0.4,m/hr,
4.9e-7,mol/liter,,,
,,0.83889e-6,mol/liter
,,1.0e-3,mol/liter,
3.125e-5,mol/liter,
75,hr,0.4,m/hr,
4.9e-7,mol/liter,,,
,,0.83889e-6,mol/liter,
,,1.0e-3,mol/liter,
3.125e-5,mol/liter,
East,Aqueous Dirichlet,Species Outflow,
0,
10,10,1,1,1,1,1,
0,day,102064.81,Pa,

Extracted from a STOMP-WA input file:
~Boundary Conditions Card
3,
Top,Zero Flux,Dirichlet,Zero Flux,
1,60,1,1,20,20,1,
0,s,,,,101325,Pa,1.0,,,
West,Hydraulic Gradient,Zero Flux,Outflow,
1,1,1,1,1,20,1,
0,s,106173.84,Pa,0.0,,,,,,
East,Hydraulic Gradient,Zero Flux,Outflow,
60,60,1,1,1,20,1,
0,s,105880.39,Pa,0.0,,,,,,

Extracted from a STOMP-WAE input file:
~Boundary Conditions Card
2,
West,Dirichlet Energy,Dirichlet Aqueous,Dirichlet Gas,
1,1,1,1,1,1,1,
0,day,70,C,101330,Pa,0,101330,Pa,1,
East,Neumann Energy,Zero Flux Aqueous,Zero Flux Gas,
50,50,1,1,1,1,1,
0,Day,-100,W/m^2,,,,,,

Extracted from a STOMP-WAE-B input file:
#---
~Boundary Conditions Card
#---
2,
Top,Bare Shuttleworth-Wallace,,,
1,1,1,1,113,113,1455,
file,idaho_precip.dat,

Boundary Conditions Card Examples (cont'd)

Bottom,dirichlet Energy,Seepage Face Aqueous,Dirichlet Gas,
1,1,1,1,1,1,2,
0,day,15.0,C,101325,pa,1.0,101365,pa,1.0,
803,day,15.0,C,101325,pa,1.0,101365,pa,1.0,

Extracted from a STOMP-WAE-B input file:

```
#-------------------------------------------------------
~Boundary Conditions Card
#-------------------------------------------------------
1,
Top,Bare Shuttleworth-Wallace,,,
1,1,1,1,113,113,6,
0.0,day,0,in/hr,,,,,,,
0.375,day,0,in/hr,,,,,,,
0.375,day,1.013188976,in/hr,,,,,,,
0.5833,day,1.013188976,in/hr,,,,,,,
0.5833,day,0,in/hr,,,,,,,
1.375,day,0,in/hr,,,,,,
```

Extracted from a STOMP-WAE-B input file:

```
#-------------------------------------------------------
~Boundary Conditions Card
#-------------------------------------------------------
2,
Top,Shuttleworth-Wallace,,,
1,1,1,1,145,145,922,
file,hms-bc1983.dat,
Bottom,Dirichlet Energy,Dirichlet Aqueous,Zero Flux Gas,
1,1,1,1,1,1,1,
0,day,20.0,C,101325,Pa,1.0,,,1.0,
```

Extracted from a STOMP-WAE-B input file:

```
#-------------------------------------------------------
~Boundary Conditions Card
#-------------------------------------------------------
1,
Top,Shuttleworth-Wallace,,,
1,1,1,1,145,145,5,
0.0,day,0,in/hr,2.55,0.20,2.55,0.30,,,,,,
1.333,day,0,in/hr,2.55,0.20,2.55,0.30,,,,,,
1.333,day,0.04,in/hr,2.55,0.20,2.55,0.30,,,,,,
1.375,day,0.04,in/hr,2.55,0.20,2.55,0.30,,,,,,
1.375,day,0.01,in/hr,2.55,0.20,2.55,0.30,,,,,,
```

Extracted from a STOMP-WO input file:

```
~Boundary Conditions Card
2,
Top,Zero Flux,Dirichlet,
```

1,1,1,1,91,91,2,
0,hr,-1.e9,Pa,1.,101758.43,Pa,
25,d,-1.e9,Pa,1.,101758.43,Pa,
Bottom,Dirichlet,Zero Flux,
1,1,1,1,1,1,2,
0,s,110629,Pa,1.0,,,
10,min,109595.22,Pa,1.0,,,

Extracted from a STOMP-WOD input file:

~Boundary Conditions Card
4,
Top,Aqueous Dirichlet,Aqueous Mass Fraction,NAPL Zero Flux,NAPL Mass Fraction,
1,1,1,1,20,20,2,
0,d,101325,Pa,,,-1.e9,Pa,,,
1,d,101325,Pa,,,-1.e9,Pa,,,
Bottom,Aqueous Zero Flux,Aqueous Mass Fraction,NAPL Neumann,NAPL Mass Fraction,
1,1,1,1,1,1,2,
0,d,-1.e9,Pa,,,0.01,cm/min,0.5,0.5,
1000,min,-1.e9,Pa,,,0.01,cm/min,0.5,0.5,
Top,Aqueous Dirichlet,Aqueous Mass Fraction,NAPL Zero Flux,NAPL Mass Fraction,
1,1,1,1,20,20,2,
1,d,101425,Pa,,,-1.e9,Pa,,,
4,d,101425,Pa,,,-1.e9,Pa,,,
Bottom,Aqueous Dirichlet,Aqueous Mass Fraction,NAPL Dirichlet,NAPL Mass Fraction,
1,1,1,1,1,1,2,
1,d,101325,Pa,,,-1.e9,Pa,0.5,0.5,
4,d,101325,Pa,,,-1.e9,Pa,0.5,0.5,

Extracted from a STOMP-WOA input file:

1,
Top,Neumann,Dirichlet,Neumann,
13,15,13,17,79,79,18,
1955.5,yr,-3.04,m/yr,,,101325.0,Pa,,,-0.062,m/yr,
1956,yr,-3.04,m/yr,,,101325.0,Pa,,,-0.062,m/yr,
1956,yr,-2.47,m/yr,,,101325.0,Pa,,,-0.277,m/yr,
1957,yr,-2.47,m/yr,,,101325.0,Pa,,,-0.277,m/yr,
1957,yr,-2.95,m/yr,,,101325.0,Pa,,,-0.277,m/yr,
1958,yr,-2.95,m/yr,,,101325.0,Pa,,,-0.277,m/yr,
1958,yr,-3.92,m/yr,,,101325.0,Pa,,,-0.264,m/yr,
1959,yr,-3.92,m/yr,,,101325.0,Pa,,,-0.264,m/yr,
1959,yr,-3.07,m/yr,,,101325.0,Pa,,,-0.283,m/yr,
1960,yr,-3.07,m/yr,,,101325.0,Pa,,,-0.283,m/yr,
1960,yr,-3.42,m/yr,,,101325.0,Pa,,,-0.283,m/yr,
1961,yr,-3.42,m/yr,,,101325.0,Pa,,,-0.286,m/yr,
1961,yr,-4.23,m/yr,,,101325.0,Pa,,,-0.379,m/yr,
1962,yr,-4.23,m/yr,,,101325.0,Pa,,,-0.379,m/yr,
1962,yr,-1.98,m/yr,,,101325.0,Pa,,,-0.178,m/yr,
1962.5,yr,-1.98,m/yr,,,101325.0,Pa,,,-0.178,m/yr,
1962.5,yr,-0.0,m/yr,,,101325.0,Pa,,,-0.0,m/yr,
2003,yr,-0.0,m/yr,,,101325.0,Pa,,,-0.0,m/yr,

Extracted from a STOMP-WOA input file:

```
~Boundary Conditions Card
10,
North,Fluctuating Water Table,Hydraulic Gradient,Dirichlet,
1,110,115,115,1,85,1,
0,yr,988890.61,Pa,,,103234.0,Pa,,,-1.e+09,Pa,
South,Fluctuating Water Table,Hydraulic Gradient,Dirichlet,
1,110,1,1,1,85,1,
0,yr,992326.8,Pa,,,103234.0,Pa,,,-1.e+09,Pa,
East,Zero Flux,Hydraulic Gradient,Dirichlet,
110,110,1,115,1,85,1,
0,yr,,,,,103234.0,Pa,1.0,,-1.e+09,Pa,
West,Zero Flux,Hydraulic Gradient,Dirichlet,
1,1,1,115,1,85,1,
0,yr,,,,,103234.0,Pa,1.0,,-1.e+09,Pa,
Top,Neumann,Dirichlet,Dirichlet,
1,110,1,45,85,85,2,
1954,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
2004,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
Top,Neumann,Dirichlet,Dirichlet,
1,110,66,115,85,85,2,
1954,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
2004,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
Top,Neumann,Dirichlet,Dirichlet,
1,50,46,65,85,85,2,
1954,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
2004,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
Top,Neumann,Dirichlet,Dirichlet,
61,110,46,65,85,85,2,
1954,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
2004,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
#z9 trench area
Top,Neumann,Zero Flux,Dirichlet,
51,60,46,65,79,79,2,
1954,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
1955.5,yr,-0.5,cm/yr,,,101325.0,Pa,0.54,,-1.e+09,Pa,
Top,Neumann,Zero Flux,Neumann,
51,60,46,65,79,79,18,
1955.5,yr,-3.04,m/yr,,,101325,Pa,0.54,,-0.062,m/yr,
1956,yr,-3.04,m/yr,,,101325,Pa,0.54,,-0.062,m/yr,
1956,yr,-2.47,m/yr,,,101325,Pa,0.54,,-0.277,m/yr,
1957,yr,-2.47,m/yr,,,101325,Pa,0.54,,-0.277,m/yr,
1957,yr,-2.95,m/yr,,,101325,Pa,0.54,,-0.277,m/yr,
1958,yr,-2.95,m/yr,,,101325,Pa,0.54,,-0.277,m/yr,
1958,yr,-3.92,m/yr,,,101325,Pa,0.54,,-0.264,m/yr,
1959,yr,-3.92,m/yr,,,101325,Pa,0.54,,-0.264,m/yr,
1959,yr,-3.07,m/yr,,,101325,Pa,0.54,,-0.283,m/yr,
1960,yr,-3.07,m/yr,,,101325,Pa,0.54,,-0.283,m/yr,
1960,yr,-3.42,m/yr,,,101325,Pa,0.54,,-0.283,m/yr,
1961,yr,-3.42,m/yr,,,101325,Pa,0.54,,-0.286,m/yr,
```

1961,yr,-4.23,m/yr,,,101325,Pa,0.54,,-0.379,m/yr,

Boundary Conditions Card Examples (cont'd)

1962,yr,-4.23,m/yr,,,101325,Pa,0.54,,-0.379,m/yr,
1962,yr,-1.98,m/yr,,,101325,Pa,0.54,,-0.178,m/yr,
1962.5,yr,-1.98,m/yr,,,101325,Pa,0.54,,-0.178,m/yr,
1962.5,yr,-0.0,m/yr,,,101325,Pa,0.54,,-0.0,m/yr,
2004,yr,-0.0,m/yr,,,101325,Pa,0.54,,-0.0,m/yr,

Extracted from a STOMP-WS input file:
~Boundary Conditions Card
1,
Bottom,Dirichlet,Outflow,
1,90,1,1,1,1,1,
0.0,min,101325.,Pa,,,

Extracted from a STOMP-WCSE input file:
~Boundary Conditions Card
1,
South,Energy Dirichlet,Aqu. Dirichlet,Gas Dirichlet,Aqu. Mass Frac.,
98,98,1,1,3,3,1,
0,s,45.0,C,138.0,bar,0.0,138.0,bar,1.0,0.0,,

Extracted from a STOMP-WCMSE input file:
~Boundary Conditions Card
1,
West,Energy Dirichlet,Aqueous Dirichlet,Gas Dirichlet,Liquid CO2 Zero Flux,
1,1,1,1,1,1,1,
0,day,45,C,8.0,MPa,0.0,0.0,0.0,0.0,8.0,MPa,0.0,1.0,8.0,MPa,

B.5 Conservation Equations Card

Card Title[a] { ~Conservation Equations [Card] }
Format: *Char*[a],

If: Operational Mode = { **STOMP-W-R** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }

Number of Conservation Equations[a],
Format: *Integer*[a],

For: Number of Conservation Equations
Component Species Name[a],
Number of Species in Conservation Equation[b],
For: Number of Species in Conservation Equation
Species Name[c], Species Stochiometric Coefficient[d],
Endfor:
Format: *Char*[a], *Integer*[b], < *Char*[c], *Real*[d], >
Endfor:

Endcard: Conservation Equations Card

B.5.1 Conservation Equations Card Examples

Extracted from a STOMP-W input file:

```
~Conservation Equations Card
10,
Total_CO2,3,CO2,0.100E+01,CO3--,0.100E+01,HCO3-,0.100E+01,
Total_Cl-,1,Cl-,0.100E+01,
Total_Co++,9,Co++,0.100E+01,Co(ads),0.100E+01,CoNTA(ads),0.100E+01,
 Co(OH)2,0.100E+01,Co(OH)3-,0.100E+01,CoNTA-,0.100E+01,
 CoNTA2----,0.100E+01,CoOH+,0.100E+01,CoOHNTA--,0.100E+01,
Total_H+,13,H+,0.100E+01,C5H7O2N,0.594E+01,CO3--,-0.200E+01,
 Co(OH)2,-0.200E+01,Co(OH)3-,-0.300E+01,CoOH+,-0.100E+01,
 CoOHNTA--,-0.100E+01,H2NTA-,0.200E+01,H3NTA,0.300E+01,
 HCO3-,-0.100E+01,HNTA--,0.100E+01,NH3,-0.100E+01,OH-,-0.100E+01,
Total_H2CO3,2,H2CO3,0.100E+01,C5H7O2N,-0.542E+01,
Total_H2O,9,H2O,0.100E+01,C5H7O2N,0.221E+01,CO3--,0.100E+01,
 Co(OH)2,0.200E+01,Co(OH)3-,0.300E+01,CoOH+,0.100E+01,
 CoOHNTA--,0.100E+01,HCO3-,0.100E+01,OH-,0.100E+01,
Total_NH4+,3,NH4+,0.100E+01,C5H7O2N,-0.736E+00,NH3,0.100E+01,
Total_NTA---,9,NTA---,0.100E+01,CoNTA(ads),0.100E+01,
 C5H7O2N,0.174E+01,CoNTA-,0.100E+01,CoNTA2----,0.200E+01,
 CoOHNTA--,0.100E+01,H2NTA-,0.100E+01,H3NTA,0.100E+01,
 HNTA--,0.100E+01,
Total_Na+,1,Na+,0.100E+01,
Total_O2,2,O2,0.100E+01,C5H7O2N,0.281E+01,
```

Extracted from a STOMP-W input file:

```
~Conservation Equations Card
8,
Total_CO2,4,CO2,1.00000E+00,C5H7O2N,-5.41667E+00,CO3--,1.00000E+00,HCO3-,1.00000E+00,
Total_Cl-,1,Cl-,1.00000E+00,
Total_Co++,9,Co++,1.00000E+00,Co(ads),1.00000E+00,CoNTA(ads),1.00000E+00,Co(OH)2,1.00000E+00,C
o(OH)3-,1.00000E+00,CoNTA-,1.00000E+00,CoNTA2----,1.00000E+00,CoOH+,1.00000E+00,CoOHNTA--
,1.00000E+00,
Total_H+,13,H+,1.00000E+00,C5H7O2N,5.93750E+00,CO3--,-2.00000E+00,Co(OH)2,-
2.00000E+00,Co(OH)3-,-3.00000E+00,CoOH+,-1.00000E+00,CoOHNTA--,-1.00000E+00,H2NTA-
,2.00000E+00,H3NTA,3.00000E+00,HCO3-,-1.00000E+00,HNTA--,1.00000E+00,NH3,-1.00000E+00,OH-,-
1.00000E+00,
Total_NH4+,3,NH4+,1.00000E+00,C5H7O2N,-7.36111E-01,NH3,1.00000E+00,
Total_NTA---,9,NTA---,1.00000E+00,CoNTA(ads),1.00000E+00,C5H7O2N,1.73611E+00,CoNTA-
,1.00000E+00,CoNTA2----,2.00000E+00,CoOHNTA--,1.00000E+00,H2NTA-
,1.00000E+00,H3NTA,1.00000E+00,HNTA--,1.00000E+00,
Total_Na+,1,Na+,1.00000E+00,
Total_O2,2,O2,1.00000E+00,C5H7O2N,2.81250E+00,
```

B.6 Directional Aqueous Relative Permeability Card

Card Title[a] { ~X-Aqueous Rel [ative Permeability Card] } and/or
Card Title[a] { ~Y-Aqueous Rel [ative Permeability Card] } and/or
Card Title[a] { ~Z-Aqueous Rel [ative Permeability Card] }

For each directional card:

Format: *Char*[a]

For each directional aqueous relative permeability card, IJK indexing can be used:

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. Example input cards are included in section B.6.1

Endif:

The format of the line input for each direction is identical to the Aqueous Relative Permeability Card.

B.6.1 Directional Aqueous Relative Permeability Card Examples

Extracted from STOMP-W input file:
~X-Direction Aqueous Relative Permeabiliy Card
IJK Indexing, Tabular, 12, file: tabh.dat, file: tabkrx.dat

Extracted from STOMP-W input file:
~Y-Direction Aqueous Relative Permeabiliy Card
IJK Indexing, Tabular, 12, file: tabh.dat, file: tabkry.dat

Extracted from STOMP-W input file:
~Z-Direction Aqueous Relative Permeabiliy Card
IJK Indexing, Tabular, 12, file: tabh.dat, file: tabkrz.dat

B.7 Dissolved-Oil Transport Card

Card Title[a] { ~Dissolved Oil Transport [Card] } {~Dissolved VOC Transport [Card] }
{ ~Dissolved Organic Transport [Card] }
Format: *Char[a]*

If: Operational Mode ={ **STOMP-WOD** }
Interphase Mass Transfer Function Option[a],

{ Welty } **Note:** $Sh = \dfrac{k^{o}_{n\ell} d_p}{D^{o}_{\ell}} = a + bRe^{m} Sc^{n}$

{ Parker } **Note:** $Sh = \dfrac{k^{o}_{n\ell} d_p}{D^{o}_{\ell}} = a + bRe^{m} \left(s_n n_D \right)^{n}$

Constant a[b], Constant b[c], Constant m[d], Constant n[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Endif:

For: Number of Rock/Soil Types
If: Operational Mode = { **STOMP-WO** } **{ STOMP-WOA } { STOMP-WOA-Sc }**
Rock/Soil Name[a],
Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m),
Dissolved-Oil Adsorption Function Option[f],

{ Linear } **Note:** $R_D = \left[1 + \dfrac{k}{s_\ell n_D} \right]$

{ Linear Kd } **Note:** $R_D = \left[1 + \dfrac{K_d \rho_b}{s_\ell n_D} \right]$

{ Freundlich } **Note:** $R_D = \left[1 + \dfrac{n k C_\ell^{n-1}}{s_\ell n_D} \right]$

{ Langmuir } **Note:** $R_D = \left[1 + \dfrac{a}{s_\ell n_D \left(1 + b C_\ell \right)^2} \right]$

If: Dissolved-Oil Adsorption Function Option = { Linear }
Constant k[g],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g],*
Elseif: Dissolved-Oil Adsorption Function Option = { Linear Kd }
Constant Kd[g], Units[h] (m^3/kg),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h],*

Elseif: Dissolved-Oil Adsorption Function Option = { Freundlich }
Constant k[g],
Constant n[h],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved-Oil Adsorption Function Option = { Langmuir }
Constant a[g], Constant b[h], Units[i] (m^3/kg),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h], Char[i],*
Endif:
Elseif: Operational Mode = { **STOMP-WOD** }
Rock/Soil Name[a],
Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m),
Dissolved-Oil Adsorption Function Option[f],

{ Linear } **Note:** $R_D = \left[1 + \frac{k}{s_\ell n_D}\right]$

{ Linear Kd } **Note:** $R_D = \left[1 + \frac{K_d \rho_b}{s_\ell n_D}\right]$

{ Freundlich } **Note:** $R_D = \left[1 + \frac{n k C_\ell^{n-1}}{s_\ell n_D}\right]$

{ Langmuir } **Note:** $R_D = \left[1 + \frac{a}{s_\ell n_D (1 + b C_\ell)^2}\right]$

If: Dissolved-Oil Adsorption Function Option = { Linear }
Constant k[g], Nominal Particle Diameter[h], Units[i] (m),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h], Char[i],*
Elseif: Dissolved-Oil Adsorption Function Option = { Linear Kd }
Constant Kd[g], Units[h] (m^3/kg), Nominal Particle Diameter[i], Units[j] (m),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Dissolved-Oil Adsorption Function Option = { Freundlich }
Constant k[g], Constant n[h], Nominal Particle Diameter[i], Units[j] (m),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h], Real[h], Char[i],*
Elseif: Dissolved-Oil Adsorption Function Option = { Langmuir }
Constant a[g], Constant b[h], Units[i] (m^3/kg), Nominal Particle Diameter[j], Units[k] (m),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h], Char[i], Real[j], Char[k],*
Endif:
Endif:

Endfor: Number of Rock/Soil Types

Endcard: Dissolved-Oil Transport Card

B.7.1 Dissolved-Oil Transport Card Examples

Extracted from a STOMP-WO input file:
~Dissolved Oil Transport Card
Sand,0.17,cm,,cm,linear kd,0.0,m^3/kg,

Extracted from a STOMP-WOD input file:
~Dissolved Oil Transport Card
Welty,0.55,0.25,1.5,1.5,
20/30 Ottawa Sand,0.154,cm,0.0154,cm,Linear Kd,0.,m^3/kg,0.71,mm,

B.8 Equilibrium Equations Card

Card Title[a] { ~Equilibrium Equations [Card] }
Format: *Char*[a],

If: Operational Mode = { **STOMP-W-R** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }

Number of Equilibrium Equations[a],
Format: *Integer*[a],

For: Number of Equilibrium Equations
Number of Species in Equilibrium Equation[a],
For: Number of Species in Equilibrium Equation
Species Name[b], Species Exponent[c],
Endfor:
Equilibrium Reaction Name[d],
Format: *Integer*[a], < *Char*[b], *Real*[c], [cr] > *Char*[d],
Endfor:

Endif:

Endcard: Equilibrium Equations Card

B.8.1 Equilibrium Equations Card Examples

Extracted from a STOMP-W input file:
~Equilibrium Equations Card
13,
3,H3NTA,H+,3.0,NTA---,1.0,EqRc-1,
3,H2NTA-,H+,2.0,NTA---,1.0,EqRc-2,
3,HNTA--,H+,1.0,NTA---,1.0,EqRc-3,
3,CoNTA-,Co++,1.0,NTA---,1.0,EqRc-4,
3,CoNTA2----,Co++,1.0,NTA---,2.0,EqRc-5,
5,CoOHNTA--,Co++,1.0,H+,-1.0,H2O,1.0,NTA---,1.0,EqRc-6,
4,CoOH+,Co++,1.0,H+,-1.0,H2O,1.0,EqRc-7,
4,Co(OH)2,Co++,1.0,H+,-2.0,H2O,2.0,EqRc-8,
4,Co(OH)3-,Co++,1.0,H+,-3.0,H2O,3.0,EqRc-9,
4,HCO3-,CO2,1.0,H+,-1.0,H2O,1.0,EqRc-10,
4,CO3--,CO2,1.0,H+,-2.0,H2O,1.0,EqRc-11,
3,NH3,H+,-1.0,NH4+,1.0,EqRc-12,
3,OH-,H+,-1.0,H2O,1.0,EqRc-13,

Extracted from a STOMP-W input file:
~Equilibrium Equations Card
25,
3,UO2OH+,H+,-1.00000e+00,UO2++,1.00000e+00,EqRc-1,1.0,
3,UO2(OH)2,H+,-2.00000e+00,UO2++,1.00000e+00,EqRc-2,1.0,
3,UO2(OH)3-,H+,-3.00000e+00,UO2++,1.00000e+00,EqRc-3,1.0,
3,UO2(OH)4--,H+,-4.00000e+00,UO2++,1.00000e+00,EqRc-4,1.0,
3,(UO2)2OH+++,H+,-1.00000e+00,UO2++,2.00000e+00,EqRc-5,1.0,
3,(UO2)2(OH)2++,H+,-2.00000e+00,UO2++,2.00000e+00,EqRc-6,1.0,
3,(UO2)3(OH)4++,H+,-4.00000e+00,UO2++,3.00000e+00,EqRc-7,1.0,
3,(UO2)3(OH)5+,H+,-5.00000e+00,UO2++,3.00000e+00,EqRc-8,1.0,
3,(UO2)3(OH)7-,H+,-7.00000e+00,UO2++,3.00000e+00,EqRc-9,1.0,
3,(UO2)4(OH)7+,H+,-7.00000e+00,UO2++,4.00000e+00,EqRc-10,1.0,
3,CO3--,H+,-1.00000e+00,HCO3-,1.00000e+00,EqRc-11,1.0,
4,UO2(CO3)2--,H+,-2.00000e+00,HCO3-,2.00000e+00,UO2++,1.00000e+00,EqRc-12,1.0,
4,UO2(CO3)3----,H+,-3.00000e+00,HCO3-,3.00000e+00,UO2++,1.00000e+00,EqRc-13,1.0,
4,(UO2)2CO3(OH)3-,H+,-4.00000e+00,HCO3-,1.00000e+00,UO2++,2.00000e+00,EqRc-14,1.0,
5,CaUO2(CO3)3--,Ca++,1.00000e+00,H+,-3.00000e+00,HCO3-,3.00000e+00,UO2++,1.00000e+00,EqRc-15,1.0,
5,Ca2UO2(CO3)3,Ca++,2.00000e+00,H+,-3.00000e+00,HCO3-,3.00000e+00,UO2++,1.00000e+00,EqRc-16,1.0,
3,UO2NO3+,NO3-,1.00000e+00,UO2++,1.00000e+00,EqRc-17,1.0,
3,UO2Cl+,Cl-,1.00000e+00,UO2++,1.00000e+00,EqRc-18,1.0,
3,UO2Cl2,Cl-,2.00000e+00,UO2++,1.00000e+00,EqRc-19,1.0,
3,UO2SO4,SO4--,1.00000e+00,UO2++,1.00000e+00,EqRc-20,1.0,
3,UO2(SO4)2--,SO4--,2.00000e+00,UO2++,1.00000e+00,EqRc-21,1.0,
4,UO2CO3,H+,-1.00000e+00,HCO3-,1.00000e+00,UO2++,1.00000e+00,EqRc-22,1.0,
3,H2CO3,H+,1.00000e+00,HCO3-,1.00000e+00,EqRc-23,1.0,
4,SOUO2OH,H+,-2.00000e+00,SOH,1.00000e+00,UO2++,1.00000e+00,EqRc-24,1.0,
5,SOUO2(HCO3)2-,H+,-1.00000e+00,HCO3-,2.00000e+00,SOH,1.00000e+00,UO2++,1.00000e+00,EqRc-25,1.0,

B.9 Equilibrium Reactions Card

Card Title[a] { ~Equilibrium Reactions [Card] }
Format: *Char*[a],

If: Operational Mode = { **STOMP-W-R** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }

Number of Equilibrium Reactions[a],
Format: *Integer*[a],
For: Number of Equilibrium Reactions
Equilibrium Reaction Name[a],
Equilibrium Constant Coefficient *a*[b],
Equilibrium Constant Coefficient *b*[c],
Equilibrium Constant Coefficient *c*[d],
Equilibrium Constant Coefficient *d*[e],
Equilibrium Constant Coefficient *e*[f],
Format: *Char*[a], *Real*[b], *Real*[c], *Real*[d], *Real*[e], *Real*[f],

Note: $\log\left(K_{eq}\right) = a\ln\left(T\right) + b + c\,T + \frac{d}{T} + \frac{e}{T^2}$

Endfor:

Endif:

Endcard: Equilibruim Reactions Card

B.9.1 Equilibrium Equations Card Examples

Extracted from a STOMP-W input file:

```
~Equilibrium Reactions Card
13,
EqRc-1,0.0,14.900,0.0,0.0,0.0,1/mol,
EqRc-2,0.0,13.300,0.0,0.0,0.0,1/mol,
EqRc-3,0.0,10.300,0.0,0.0,0.0,1/mol,
EqRc-4,0.0,11.700,0.0,0.0,0.0,1/mol,
EqRc-5,0.0,14.500,0.0,0.0,0.0,1/mol,
EqRc-6,0.0,0.500,0.0,0.0,0.0,1/mol,
EqRc-7,0.0,-9.700,0.0,0.0,0.0,1/mol,
EqRc-8,0.0,-22.900,0.0,0.0,0.0,1/mol,
EqRc-9,0.0,-31.500,0.0,0.0,0.0,1/mol,
EqRc-10,0.0,-6.350,0.0,0.0,0.0,1/mol,
EqRc-11,0.0,-16.680,0.0,0.0,0.0,1/mol,
EqRc-12,0.0,-9.300,0.0,0.0,0.0,1/mol,
EqRc-13,0.0,-14.000,0.0,0.0,0.0,1/mol,
```

Extracted from a STOMP-W input file:

```
~Equilibrium Reactions Card
25,
EqRc-1,0.0,-5.250,0.0,0.0,0.0,1/mol,
EqRc-2,0.0,-12.150,0.0,0.0,0.0,1/mol,
EqRc-3,0.0,-20.250,0.0,0.0,0.0,1/mol,
EqRc-4,0.0,-32.400,0.0,0.0,0.0,1/mol,
EqRc-5,0.0,-2.700,0.0,0.0,0.0,1/mol,
EqRc-6,0.0,-5.620,0.0,0.0,0.0,1/mol,
EqRc-7,0.0,-11.900,0.0,0.0,0.0,1/mol,
EqRc-8,0.0,-15.550,0.0,0.0,0.0,1/mol,
EqRc-9,0.0,-32.200,0.0,0.0,0.0,1/mol,
EqRc-10,0.0,-21.900,0.0,0.0,0.0,1/mol,
EqRc-11,0.0,-10.329,0.0,0.0,0.0,1/mol,
EqRc-12,0.0,-4.048,0.0,0.0,0.0,1/mol,
EqRc-13,0.0,-9.147,0.0,0.0,0.0,1/mol,
EqRc-14,0.0,-11.184,0.0,0.0,0.0,1/mol,
EqRc-15,0.0,-5.587,0.0,0.0,0.0,1/mol,
EqRc-16,0.0,-0.437,0.0,0.0,0.0,1/mol,
EqRc-17,0.0,0.300,0.0,0.0,0.0,1/mol,
EqRc-18,0.0,0.170,0.0,0.0,0.0,1/mol,
EqRc-19,0.0,-1.100,0.0,0.0,0.0,1/mol,
EqRc-20,0.0,3.150,0.0,0.0,0.0,1/mol,
EqRc-21,0.0,4.140,0.0,0.0,0.0,1/mol,
EqRc-22,0.0,-0.389,0.0,0.0,0.0,1/mol,
EqRc-23,0.0,6.354,0.0,0.0,0.0,1/mol,
EqRc-24,0.0,-4.548,0.0,0.0,0.0,1/mol,
EqRc-25,0.0,8.661,0.0,0.0,0.0,1/mol,
```

B.10 Gas Relative Permeability Card

Card Title[a] { ~Gas Rel [ative Permeability Card] }
Format: *Char*[a]

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section B.10.1

Else:

For: Number of Rock/Soil Types

If: Operational Mode ={ **STOMP-WA** } { **STOMP-WAE** } { **STOMP-WAE-B** } { **STOMP-WAS** } { **STOMP-WASE** }
Rock/Soil Name[a],
Permeability Function Option[b],
{ Constant | Mualem | Burdine | Fatt and Klikoff | Corey | Tabular [Linear | Spline] [Water Content | Saturation] }
If: Permeability Function Option = { Constant }
If: Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix Gas Relative Permeability[c],
Fracture Gas Relative Permeability[d],
Format: *Char*[a], *Char*[b], *Real*[c], *Real*[d],
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char*[a], *Char*[b], *Real*[c], *Real*[d], *Char*[e], *Real*[f], *Real*[g], *Char*[h],
Endif:
Else:
Gas Relative Permeability[c]
Format: *Char*[a], *Char*[b], *Real*[c],
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
If: Saturation Function Option = {{ van Genuchten }}
and Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix van Genuchten m parameter[c],
Fracture van Genuchten m parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Elseif: Saturation Function Option = {{ Brooks and Corey }}
and Rock/Soil Type Name = {{ Fractured }} {{ DP }}
Matrix Brooks and Corey λ parameter[c],
Fracture Brooks and Corey λ parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Elseif: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],
Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Endif:
Elseif: Permeability Function Option = { Corey }
Irreducible Gas Saturation[c], Irreducible Aqueous Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }

Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Elseif: Permeability Function Option = { Fatt and Klikoff }
Format: *Char[a], Char[b],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[c], { Klinkenberg }
Scaling Parameter (C1)[d], Exponential Parameter[e], Pressure Units[f],
Format: *Char[a], Char[b], Char[c], Real[d], Real[e], Char[f],*
Endif:
Elseif: Permeability Function Option = { Tabular Water Content [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a],Char[b],Integer[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
For: Number of Table Entries
Water Content[a], Gas Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor:
Elseif: Permeability Function Option = { Tabular [Saturation] [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a],Char[b],Integer[c],*
For: Number of Table Entries
Saturation[a], Gas Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor:
Endif:

Elseif: Operational Mode = **{ STOMP-WOA } { STOMP-WOA-Sc }{ STOMP-WOAE }**
Rock/Soil Name[a],
Permeability Function Option[b]
{ Constant | Mualem | Burdine }
If: Permeabililty Function Option = { Constant }
Aqueous Relative Permeability[c],
Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Else:
If: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],

Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Endif:
Endif:

Elseif: Operational Mode = { **STOMP-WCS** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** } { **STOMP-WCMSE** }
Rock/Soil Name[a],
Permeability Function Option[b],
{ Constant | Mualem | Burdine | Fatt and Klikoff | Corey | Free Corey |
| Tabular [Linear | Spline] [Water Content | Saturation] }
If: Permeability Function Option = { Constant }
If: Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix Gas Relative Permeability[c],
Fracture Gas Relative Permeability[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Else:
Gas Relative Permeability[c]
Format: *Char[a], Char[b], Real[c],*
Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
If: Saturation Function Option = {{ van Genuchten }}
and Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix van Genuchten m parameter[c],
Fracture van Genuchten m parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],

Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Elseif: Saturation Function Option = {{ Brooks and Corey }}
and Rock/Soil Name = {{ Fractured }} {{ DP }}
Matrix Brooks and Corey λ parameter[c],
Fracture Brooks and Corey λ parameter[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Elseif: Saturation Function Option = {{ van Genuchten }}
van Genuchten m parameter[c],
Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Elseif: Saturation Function Option = {{ Brooks and Corey }}
Brooks and Corey λ parameter[c],
Format: *Char[a], Char[b], Real[c],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[d], { Klinkenberg }
Scaling Parameter (C1)[e], Exponential Parameter[f], Pressure Units[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Char[g],*
Endif:
Endif:
Elseif: Permeability Function Option = { Corey }
Irreducible Gas Saturation[c], Irreducible Aqueous Saturation[d],
Format: *Char[a], Char[b], Real[c], Real[d],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[e], { Klinkenberg }
Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],*
Endif:
Elseif: Permeability Function Option = { Free Corey }
Endpoint Gas Permeability[c], Exponent Gas Relative Permeability[d],
Residual Aqueous Saturation[e], Residual Gas Saturation[f],
Format: *Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[g], { Klinkenberg }
Scaling Parameter (C1)[h], Exponential Parameter[i], Pressure Units[h],
Format: *Char[a], Char[b], Real[c], Real[d], Real[e], Real[f], Char[g], Real[h], Real[i], Char[j],*

Endif:

Gas Permeability Card (cont'd)

Elseif: Permeability Function Option = { Fatt and Klikoff }
Format: *Char[a], Char[b],*
If: Klinkenberg Gas Relative Permeability Extension is Considered
Permeability Extension[c], { Klinkenberg }
Scaling Parameter (C1)[d], Exponential Parameter[e], Pressure Units[f],
Format: *Char[a], Char[b], Char[c], Real[d], Real[e], Char[f],*
Endif:
Elseif: Permeability Function Option = { Tabular Water Content [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a],Char[b],Integer[c],*
For: Number of Table Entries
Water Content[a], Gas Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor:
Elseif: Permeability Function Option = { Tabular [Saturation] [Linear | Spline] }
Number of Table Entries[c],
Format: *Char[a],Char[b],Integer[c],*
For: Number of Table Entries
Saturation[a], Gas Relative Permeability[b],
Format: *Real[a], Real[b],*
Endfor:
Endif:

Endif:

Endfor: Number of Rock/Soil Types

Endif:

Endcard: Gas Relative Permeability Card

B.10.1 Gas Relative Permeability Examples

Extracted from a STOMP-WA input file:
~Gas Relative Permeability Card
Sand,Mualem,,

Extracted from a STOMP-WA input file:
~Gas Relative Permeability Card
IJK Indexing,Constant,file:rel_g_x.dat,

Extracted from a STOMP-WA input file:
~Gas Rel
20/30 Ottawa Sand,Mualem,0.56,

Extracted from a STOMP-WA input file:
~Gas Relative Permeability Card
Sand,Fatt and Klikoff,

Extracted from a STOMP-WA input file:
~Gas Relative Permeability
SM-ML1,Burdine,,
SW1,Burdine,,
Tabular Input
SP3,Tabular,6,
1.0, 1.0,
0.8, 0.64,
0.6, 0.36,
0.4, 0.16,
0.2, 0.04,
0.0, 0.0,
SM-SP1,Mualem,0.5,
SP2,Burdine,,
SP1,Burdine,,

Extracted from a STOMP-WOA input file:

~Gas Relative Permeability Card
----
ring_a,Burdine,,Klinkenberg,0.6933,0.0501,atm,
low_rmud,Burdine,,Klinkenberg,0.6933,0.0501,atm,
ring_e,Burdine,,Klinkenberg,0.6933,0.0501,atm,
up_ring,Burdine,,Klinkenberg,0.6933,0.0501,atm,
pplc,Burdine,,Klinkenberg,0.6933,0.0501,atm,
pplz,Burdine,,Klinkenberg,0.6933,0.0501,atm,
low_sand,Burdine,,Klinkenberg,0.6933,0.0501,atm,
low_grvl,Burdine,,Klinkenberg,0.6933,0.0501,atm,
h2,Burdine,,Klinkenberg,0.6933,0.0501,atm,
h1,Burdine,,Klinkenberg,0.6933,0.0501,atm,
h1a,Burdine,,Klinkenberg,0.6933,0.0501,atm,

Gas Relative Permeability Examples (cont'd)

Extracted from a STOMP-WCMSE input file:
~Gas Relative Permeability Card
Medium,Free Corey,1.0,3.0,0.12,0.02,

B.11 Gas Species Card

Card Title[a] { ~Gas Species [Card] }
Format: *Char*[a],

If: Operational Mode = { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc }**

Number of Gas Species[a],
Gas Molecular Diffusion Coefficient[b], Units[c] (m),
Format: *Integer*[a], *Real*[b], *Char*[c],

For: Number of Gas Species
Gas Species Name[a],
Associate Aqueous Species Name[b],
Gas-Aqueous Partition Constant Coefficient *a*[b],
Gas-Aqueous Partition Constant Coefficient *b*[c],
Gas-Aqueous Partition Constant Coefficient *c*[d],
Gas-Aqueous Partition Constant Coefficient *d*[e],
Gas-Aqueous Partition Constant Coefficient *e*[f],
Format: *Char*[a], *Real*[b], *Real*[c], *Real*[d], *Real*[e], *Real*[f],

Note: $\ln\left(K_{gl}\right) = a + \frac{b}{T} + c\ln\left(T\right) + d\,T + e\,T^2$

Endfor: Number of Gas Species

Endif:

Endcard: Gas Species Card

B.12 Grid Card

Card Title[a] { ~Grid [Card] }
Format: *Char[a]*

Coordinate System Option[a],
{ Tilted | [Uniform] [Reference [Point]] Cartesian |
[Uniform] [Reference [Point]] Cylindrical |
[Boundary] [Fitted] [Orthogonal] | Generalized }

If: Coordinate System Option = { Tilted }
X-Z Plane Tilt Angle[b], Units[c] (deg),
Y-Z Plane Tilt Angle[d], Units[e] (deg),
If: Coordinate System Option = [Reference]
X Reference Point[f], Units[g] (m), Direction Index[h],
Y Reference Point[i], Units[j] (m), Direction Index[k],
Z Reference Point[l], Units[m] (m), Direction Index[n],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Integer[h],*
Real[i], Char[j], Integer[k], Real[l], Char[m], Integer[n],
Elseif:
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Endif:
Elseif:
If: Coordinate System Option = [Reference]
X Reference Point[b], Units[c] (m), Direction Index[d],
Y Reference Point[e], Units[f] (m), Direction Index[g],
Z Reference Point[h], Units[i] (m), Direction Index[j],
Format: *Char[a], Real[b], Char[c], Integer[d], Real[e], Char[f], Integer[g], Real[h], Char[i], Integer[j],*
Elseif:
Format: *Char[a],*
Endif:
Endif:

Note: A negative direction index implies a descending direction.

If: Coordinate System Option = { Generalized }
External File Name[a], Dimensional Units[b] (m),
Format: *Char[a], Char[b],*
Elseif: Coordinate System Option = { [Boundary] [Fitted] [Orthogonal] }
If: Operation Mode Option: **{ STOMP-W-Sc } { STOMP-WOA-Sc }**
{ STOMP-WCS-R-Sc }
Number of X-Dir. Grid Points[a],
Number of Y-Dir. Grid Points[b],
Number of Z-Dir. Grid Points[c],
Number of X Processors[d],
Number of Y Processors[e],
Number of Z Processors[f],

Grid Card (cont'd)

Format: *Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f],*
Else:
Number of X-Dir. Grid Points[a],
Number of Y-Dir. Grid Points[b],
Number of Z-Dir. Grid Points[c],
Format: *Integer[a], Integer[b], Integer[c],*
Endif:
Else:
If: Operation Mode Option: **{ STOMP-W-Sc } { STOMP-WOA-Sc } { STOMP-WCS-R-Sc }**
Number of X-Dir. Nodes[a],
Number of Y-Dir. Nodes[b],
Number of Z-Dir. Nodes[c],
Number of X Processors[d],
Number of Y Processors[e],
Number of Z Processors[f],
Format: *Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f],*
Else:
Number of X-Dir. Nodes[a],
Number of Y-Dir. Nodes[b],
Number of Z-Dir. Nodes[c],
Format: *Integer[a], Integer[b], Integer[c],*
Endif:
Endif:

If: Coordinate System Option = { Cartesian }

For: Number of X-Dir. Nodes + 1
< Surface Position[a], Units[b] (m), > or < Count[a] @ Node Width[b], Units[c] (m), >
Format: < *Real[a], Char[b],* > or < *Integer[a]@Real[a], Char[b],* >
Endfor: Number of X-Dir. Nodes

For: Number of Y-Dir. Nodes + 1
< Surface Position[a], Units[b] (m), > or < Count[a] @ Node Width[b], Units[c] (m), >
Format: < *Real[a], Char[b],* > or < *Integer[a]@Real[a], Char[b],* >
Endfor: Number of Y-Dir. Nodes

For: Number of Z-Dir. Nodes + 1
< Surface Position[a], Units[b] (m), > or < Count[a] @ Node Width[b], Units[c] (m), >
Format: < *Real[a], Char[b],* > or < *Integer[a]@Real[a], Char[b],* >
Endfor: Number of Z-Dir. Nodes

Elseif: Coordinate System Option = { Cylindrical }

For: Number of Radial-Dir. Nodes + 1
< Surface Position[a], Units[b] (m), > or < Count[a] @ Node Width[b], Units[c] (m), >

Grid Card (cont'd)

Format: < *Real[a], Char[b],* > or < *Integer[a]@Real[a], Char[b],* >
Endfor: Number of Radial-Dir. Nodes

For: Number of Azimuthal-Dir. Nodes + 1
< Surface Position[a], Units[b] (deg), > or < Count[a] @ Node Width[b], Units[c] (deg), >
Format: < *Real[a], Char[b],* > or < *Integer[a]@Real[a], Char[b],* >
Endfor: Number of Azimuthal-Dir. Nodes

For: Number of Z-Dir. Nodes + 1
< Surface Position[a], Units[b] (m), > or < Count[a] @ Node Width[b], Units[c] (m), >
Format: < *Real[a], Char[b],* > or < *Integer[a]@Real[a], Char[b],* >
Endfor: Number of Z-Dir. Nodes

Elseif: Coordinate System Option = { Uniform Cartesian }

X-Dir. Node Dimension[a], Units[b] (m)
Format: *Real[a], Char[b],*

Y-Dir. Node Dimension[a], Units[b] (m)
Format: *Real[a], Char[b],*

Z-Dir. Node Dimension[a], Units[b] (m)
Format: *Real[a], Char[b],*

Elseif: Coordinate System Option = { Uniform Cylindrical }

Radial-Dir. Node Dimension[a], Units[b] (m)
Format: *Real[a], Char[b],*

Azimuthal-Dir. Node Dimension[a], Units[b] (deg)
Format: *Real[a], Char[b],*

Z-Dir. Node Dimension[a], Units[b] (m)
Format: *Real[a], Char[b],*

Elseif: Coordinate System Option = { [Boundary] [Fitted] [Orthogonal] }
External File Name[a], Dimensional Units[b] (m),
Format: *Char[a], Char[b],*

Endif:

Endcard: Grid Card

B.12.1 Grid Card Examples

Extracted from a STOMP-W input file:

```
#----------------------
~Grid Card
#----------------------
Uniform Cartesian,
20,1,10,
10,cm,
10,cm,
10,cm,
```

Extracted from a STOMP-W input file:

```
~Grid Card
Cylindrical,
50,1,113,
#Nonuniform grid spacing
0,in,3.125,in,5.125,in,8,in,12,in,18@6,in,10@12,in,10@24,in,8@48,in,
0,deg,45,deg,
0,in,113@6,in,
```

Extracted from a STOMP-WO input file:

```
~Grid Card
Orthogonal,
61,1,17,
s_shape_lr.grd,m,
```

Extracted from a STOMP-WOA-Sc input file:

```
# ------------------------------------------------------------------
~Grid Card
# ------------------------------------------------------------------
Cartesian,
110,115,85,6,6,4,
-220,m,40@5,m,10@1.545,m,10@0.91,m,10@1.545,m,40@5,m,
-220,m,40@5,m,5@2.17,m,20@0.915,m,5@2.17,m,40@5,m,5@20,m,
42,m,3@3,m,3@2,m,7@3,m,2@2,m,2@3,m,13@5,m,2@3,m,1@2,m,2@1,m,24@0.5,m,26@1,m,
```

Extracted from a STOMP-WCS input file:

```
~Grid Card
Cylindrical,
70,1,15,
0.25,ft,100,ft,29@100,ft,5@150,ft,5@200,ft,5@300,ft,
5@450,ft,5@700,ft,5@1000,ft,7@1500,ft,3@2000,ft,
0.0,deg,36.0,deg,
0.0,ft,15@100.0,ft,
```

B.13 Hydraulic Properties Card

Card Title[a] { ~Hydraulic [Properties Card] }
Format: *Char*[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: The parameter input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section B.12.1

Elseif:

For: Number of Rock/Soil or Scaling-Group Types

Rock/Soil or Scaling-Group Name[a],

If: Rock/Soil or Scaling-Group Name = {{ Fractured }} {{ DP }}

X-Dir. (Radial-Dir.) Matrix Intrinsic Permeability[b], Units[c] (m^2),
or X-Dir. (Radial-Dir.) Matrix Hydraulic Conductivity[b], Units[c] (hc m/s),

Y-Dir. (Azimuthal-Dir.) Matrix Intrinsic Permeability[d], Units[e] (m^2),
or Y-Dir. (Azimuthal-Dir.) Matrix Hydraulic Conductivity[d], Units[e] (hc m/s),

Z-Dir. Matrix Intrinsic Permeability[f], Units[g] (m^2),
or Z-Dir. Matrix Hydraulic Conductivity[f], Units[g] (hc m/s),

X-Dir. (Radial-Dir.) Fracture Intrinsic Permeability[h], Units[i] (m^2),
or X-Dir. (Radial-Dir.) Fracture Hydraulic Conductivity[h], Units[i] (hc m/s),

Y-Dir. (Azimuthal-Dir.) Fracture Intrinsic Permeability[j], Units[l] (m^2),
or Y-Dir. (Azimuthal-Dir.) Fracture Hydraulic Conductivity[j], Units[k] (hc m/s),

Z-Dir. Fracture Intrinsic Permeability[l], Units[m] (m^2),
or Z-Dir. Fracture Hydraulic Conductivity[l], Units[m] (hc m/s),

Hydraulic Properties Card (cont'd)

If: Operational Mode Option: { **STOMP-WCS** } { **STOMP-WCS-R** }
{ **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE }**
and NaCl precipitation is considered,
Pore-body Fractional Length[o], Fractional Critical Porosity[p],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Chars, Real[h], Char[i], Real[j], Char[k], Real[l], Char[m], Real[h], Real[i],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Chars, Real[h], Char[i], Real[j], Char[k], Real[l], Char[m],*
Endif:

Else:

X-Dir. (Radial-Dir.) Intrinsic Permeability[b], Units[c] (m^2),
or X-Dir. (Radial-Dir.) Hydraulic Conductivity[b], Units[c] (hc m/s),

Y-Dir. (Azimuthal-Dir.) Intrinsic Permeability[d], Units[e] (m^2),
or Y-Dir. (Azimuthal-Dir.) Hydraulic Conductivity[d], Units[e] (hc m/s),

Z-Dir. Intrinsic Permeability[f], Units[g] (m^2),
or Z-Dir. Hydraulic Conductivity[f], Units[g] (hc m/s),

If: Operational Mode Option: { **STOMP-WCS** } { **STOMP-WCS-R** }
{ **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE }**
and NaCl precipitation is considered,
Pore-body Fractional Length[h], Fractional Critical Porosity[i],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Chars, Real[h], Real[i],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Chars,*
Endif:

Endif:

Endfor: Number of Rock/Soil or Scaling-Group Types

Endcard: Hydraulic Properties Card

B.13.1 Hydraulic Properties Card Examples

Extracted from a STOMP-W input file:
~Hydraulic Properties Card
20/30 Ottawa Sand,1.6e-7,cm^2,1.6e-7,cm^2,1.6e-7,cm^2,

Extracted from a STOMP-W input file:
~Hydraulic Properties Card
Column,100.0,Darcy,100.0,Darcy,100.0,Darcy,

Extracted from a STOMP-W input file:
~Hydraulic Properties Card
IJK Indexing, file:ksx.dat,hc:cm/s, file:ksy.dat,hc:cm/s, file:ksz.dat,hc:cm/s,

Extracted from a STOMP-W input file:
~Hydraulic Properties Card
Geologic Media,1.020408e-9,m^2,,,1.020408e-9,m^2,

Extracted from a STOMP-W input file:
#R1 is a scaling group
~Hydraulic Properties Card
R1,477.09,hc:cm/day,477.09,hc:cm/day,477.09,hc:cm/day,

Extracted from a STOMP-W input file:
~Hydraulic Properties Card
SP1,4.1987e-12,m^2,,,4.1987e-12,m^2,
SP2,9.3436e-13,m^2,,,9.3436e-13,m^2,
SM-ML1,5.3223e-13,m^2,,,5.3223e-13,m^2,
SM-SP1,7.695e-12,m^2,,,5.13e-12,m^2,
SP3,5.505e-12,m^2,,,3.67e-12,m^2,
SW1,9.195e-12,m^2,,,6.13e-12,m^2,
US,1.0e+04,hc cm/hr,,,1.0e+06,hc cm/hr,

Extracted from a STOMP-W input file:
~Hydraulic Properties Card
Column,100.0,Darcy,100.0,Darcy,100.0,Darcy,

Extracted from a STOMP-W input file:
#7
~Hydraulic Properties Card
Backfill, 5.60e-4,hc:cm/s,,,5.60e-4,hc:cm/s,
H2 Sand, 9.88e-5,hc:cm/s,,,9.88e-5,hc:cm/s,
H1 Gravelly Sand,2.62e-4,hc:cm/s,,,2.62e-4,hc:cm/s,
H3 Gravelly Sand,5.15e-4,hc:cm/s,,,5.15e-4,hc:cm/s,
PPlgR, 5.60e-4,hc:cm/s,,,5.60e-4,hc:cm/s,
Aquifer,3000.0,hc:m/d,,,300.0,hc:m/d,

B.14 Inactive Nodes Card

Card Title[a] { ~Inactive [Nodes Card] }
Format: *Char[a]*

Inactive Domain Input Option[a],
{ [Rock | Soil] | Zonation File | File | Integer }

If: Inactive Domain Input Option = { [Rock | Soil] }

Number of Rock/Soil Type Lines[b],
Format: *Char[a], Integer[b],*

For: Number of Rock/Soil Type Lines
Inactive Rock/Soil Types[a],
Format: *Char[a],*
Endfor: Number of Rock/Soil Type Lines

Elseif: Inactive Domain Input Option = { Zonation File }

Zonation File Name[b],
Format: *Char[a], Char[b],*

Elseif: Inactive Domain Input Option = { File }

Inactive Node File Name[b],
Format: *Char[a], Char[b],*

Elseif: Inactive Domain Input Option = { Integer }

Number of Inactive Node Domains[a],
Format: *Integer[a],*

For: Number of Inactive Node Domains
I-Start Index[a], I-End Index[b],
J-Start Index[c], J-End Index[d],
K-Start Index[e], K-End Index[f],
Format: *Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f],*
Endfor: Number of Inactive Node Domains

Endif:

Endcard: Inactive Nodes Card

B.14.1 Inactive Nodes Card Examples

Extracted from a STOMP-W input file:
~Inactive Nodes Card
4,
1,1,1,1,14,113,
2,2,1,1,14,33,
2,2,1,1,101,113,
2,2,1,1,70,70,

Extracted from a STOMP-W input file:
#----------
~Inactive
#----------
1,
2,2,1,1,14,20

Extracted from a STOMP-W input file:
#
~Inactive Nodes Card
#
file,inactive.dat,

Extracted from a STOMP-W input file:
~Inactive Nodes Card
zonation file, zonation_4,

Extracted from a STOMP-W input file:
~Inactive Nodes Card
Rock/Soil Types,2,
fill material,
engineered structure,

B.15 Initial Conditions Card

Card Title[a] { ~Initial [Conditions Card] }
Format: *Char*[a]

If: Operational Mode Option: { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
{ **STOMP-WA** } { **STOMP-WAE** } { **STOMP-WAE-B** } { **STOMP-WAE-Sc** }
{ **STOMP-WS** } { **STOMP-WS-Sc** } { **STOMP-WAS** } { **STOMP-WASE** }
{ **STOMP-WCS** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** }
{ **STOMP-WCSE** }
Initial Saturation Option[a], Initial Saturation Option[b],
{ Gas Pressure, Aqueous Pressure |
Gas Pressure, Aqueous Saturation |
Aqueous Pressure, Aqueous Saturation }
Format: *Char*[a], *Char*[b],
Endif:

Number of Initial Conditions Domains[a]
Format: *Integer*[a],

For: Number of Initial Conditions Domains

Note: The [Overwrite] option is used in conjunction with Restart simulations.

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-Sc** }
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] | Matrix Pressure [Overwrite] |
Moisture Content [Overwrite] |
Aqueous Saturation | Trapped Gas Saturation [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite] , Solute Name }

Elseif: Operational Mode Option = { **STOMP-W-R** }
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] | Matrix Pressure [Overwrite] |
Aqueous Saturation | Trapped Gas Saturation [Overwrite] |
Moisture Content [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite] , Solute Name |
Species [Volumetric Conc.] [Overwrite] |
Species Aqueous [Molal Conc.] [Overwrite] |
Species Aqueous [Volumetric Conc.] [Overwrite] }

Initial Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WA }**
{ Water-Air-Energy }
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite]
Gas Pressure [Overwrite] |
Aqueous Saturation | Trapped Gas Saturation [Overwrite] |
Aqueous Dissolved Air Mole frac[tion] [Overwrite] |
Aqueous Dissolved Air Mass Fraction [Overwrite] |
Aqueous Dissolved Air Relative Saturation [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WO } { STOMP-WOM }**
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] | NAPL Pressure [Overwrite] |
Trapped NAPL Saturation [Overwrite] |
Residual NAPL Saturation [Overwrite] |
Trapped Gas Saturation [Overwrite] |
Aqueous Dissolved Oil Mole frac[tion] [Overwrite] |
Aqueous Dissolved Oil Mass Fraction [Overwrite] |
Aqueous Dissolved Oil Conc. [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute NAPL [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WOA } { STOMP-WOA-Sc }**
{ **STOMP-WOAE }**
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] | NAPL Pressure [Overwrite] |
Trapped NAPL Saturation [Overwrite] | Trapped Gas Saturation [Overwrite] |
Aqueous Dissolved Air Mole frac[tion] [Overwrite] |
Aqueous Dissolved Air Mass Fraction [Overwrite] |
Aqueous Dissolved Air Conc. [Overwrite] |
Aqueous Dissolved Oil Mole frac[tion] [Overwrite] |
Aqueous Dissolved Oil Mass Fraction [Overwrite] |
Aqueous Dissolved Oil Conc. [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name |
Solute NAPL [Volumetric Conc.] [Overwrite], Solute Name }

Initial Conditions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WOD }**
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] | NAPL Pressure [Overwrite] |
Trapped NAPL Saturation [Overwrite] | Trapped Gas Saturation [Overwrite] |
Aqueous Dissolved Oil Mass Fraction [Overwrite] |
Aqueous Dissolved Oil Mole frac[tion] [Overwrite] |
Aqueous Dissolved Oil Conc[Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute NAPL [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WS }** { **STOMP-WS-Sc }**
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation | Trapped Gas Saturation [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] | Salt Volumetric [Conc.] [Overwrite] |
Salt Aqueous [Volumetric Conc.] [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WAS }** { **STOMP-WASE }**
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation | Trapped Gas Saturation [Overwrite] |
Aqueous Dissolved Air Relative Saturation [Overwrite] |
Aqueous Dissolved Air Mass Fraction [Overwrite] |
Aqueous Dissolved Air Conc. [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] | Salt Volumetric [Conc.] [Overwrite] |
Salt Aqueous [Volumetric Conc.] [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** }
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation | [Relative] Trapped Gas Saturation [Overwrite] |
CO_2 Partial Pressure [Overwrite] |
Dissolved CO_2 Relative Saturation [Overwrite] |
Dissolved CO_2 Mass Fraction [Overwrite] |
Dissolved CO_2 Conc. [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] |
Salt Aqueous [Volumetric Conc.] [Overwrite] |
Salt Volumetric [Conc.] [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** }
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation | [Relative] Trapped Gas Saturation [Overwrite] |
CO_2 Partial Pressure [Overwrite] |
Dissolved CO_2 Relative Saturation [Overwrite] |
Dissolved CO_2 Mass Fraction [Overwrite] |
Dissolved CO_2 Conc. [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] |
Salt Aqueous [Volumetric Conc.] [Overwrite] |
Salt Volumetric [Conc.] [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name }
Species [Volumetric Conc.] [Overwrite] |
Species Aqueous [Molal Conc.] [Overwrite] |
Species Aqueous [Volumetric Conc.] [Overwrite] }

Elseif: Operational Mode Option = { **STOMP-WCSE** }
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation | [Relative] Trapped Gas Saturation [Overwrite] |

Initial Conditions Card (cont'd)

CO_2 Partial Pressure [Overwrite] |
Dissolved CO_2 Relative Saturation [Overwrite] |
Dissolved CO_2 Mass Fraction [Overwrite] |
Dissolved CO_2 Conc. [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] |
Salt Aqueous [Volumetric Conc.] [Overwrite] |
Salt Volumetric [Conc.] [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name }

Elseif: Operational Mode Option = { **STOMP-WCMSE }**
Variable Name Option[a],
{ File [Binary] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation | [Relative] Trapped Gas Saturation [Overwrite] |
Hydrate Saturation | CO_2 Partial Pressure [Overwrite] |
Dissolved CO_2 Relative Saturation [Overwrite] |
Dissolved CO_2 Mass Fraction [Overwrite] |
Dissolved CO_2 Conc. [Overwrite] |
CO_2 Mass Fraction of Hydrate Formers [Overwrite] |
CO_2 Mole Fraction of Hydrate Formers [Overwrite] |
CO_2-Slurry Aqueous Volumetric Fraction [Overwrite] |
CH_4 Partial Pressure [Overwrite] |
Dissolved CH_4 Relative Saturation [Overwrite] |
Dissolved CH_4 Mass Fraction [Overwrite] |
Dissolved CH_4 Conc. [Overwrite] |
CH_4 Mass Fraction of Hydrate Formers [Overwrite] |
CH_4 Mole Fraction of Hydrate Formers [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] |
Salt Aqueous [Volumetric Conc.] [Overwrite] |
Salt Volumetric [Conc.] [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name }
Endif:

Initial Conditions Card (cont'd)

If: Variable Name Option = {{ Pressure }}
Pressure[b], Units[c] (Pa),
If: Variable Name Option = { File [Binary] }
File Name[d], File Units[e] (Pa),
Note: File contains pressure values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m),
Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],*
Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
Endif:

Elseif: Variable Name Option = {{ Temperature }}
Temperature[b], Units[c] (C),
If: Variable Name Option = { File [Binary] }
File Name[d], File Units[e] (C),
Note: File contains temperature values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m),
Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],*
Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
Endif:

Elseif: Variable Name Option = {{ Saturation }}
Saturation[b], Null[c],
If: Variable Name Option = { File [Binary] }
Note: File contains saturation values for every node.
Filename[d],
Format: *Char[a], Real[b], Char[c], Char[d],*

Initial Conditions Card (cont'd)

Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m),
Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Real[b],Null[c], Real[d], Char[e], Real[f], Char[g], Real[h],*
Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
Endif:

Elseif: Variable Name Option = {{ Fraction }}
Mass Fraction[b], Null[c],
If: Variable Name Option = { File [Binary] }
File Name[d], Null[e] ,
Note: File contains fraction values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m),
Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Real[b],Null[c], Real[d], Char[e], Real[f], Char[g], Real[h],*
Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
Endif:

Elseif: Variable Name Option = {{ Salt Aqueous }}
Aqueous Conc[b], Units[c] (kg/m^3[3]),
If: Variable Name Option = { File [Binary] }
File Name[d], Units[e] (kg/m^3),
Note: File contains salt Conc. values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m),

Y-Dir. Gradient[f], Units[g] (1/m),

Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Real[b], Null[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],*
Endif:

Elseif: Variable Name Option = {{ Salt Volumetric }}
Volumetric Conc[b], Units[c] (kg/m^3[3]),
If: Variable Name Option = { File [Binary] }
File Name[d], Units[e] (kg/m^3),
Note: File contains salt Conc. values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m),
Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Real[b], Null[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],*
Endif:

Elseif: Variable Name Option = {{ Solute }}
Solute Name[b], Volumetric Conc[c], Units[d] (1/m^3),
If: Variable Name Option = { File [Binary] }
File Name[d], Units[e] (1/m^3),
Note: File contains solute Conc. values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m), Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o], Integer[p],*
Endif:

Initial Conditions Card (cont'd)

Elseif: Variable Name Option = {{ Species }}
Solute Name[b], Volumetric Conc[c], Units[d] (mol/m^3),
If: Variable Name Option = { File [Binary] }
File Name[d], Units[e] (mol/m^3),
Note: File contains solute Conc. values for every node.
Format: *Char[a], Real[b], Char[c], Char[d], Char[e],*
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: *Char[a], Real[b], Char[c], Char[d],*
Else:
X-Dir. Gradient[d], Units[e] (1/m), Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],
J-Start Index[l], J-End Index[m],
K-Start Index[n], K-End Index[o],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h],*
Real[i], Char[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o], Integer[p],
Endif:

Endif:

Endfor: Number of Initial Conditions Domains

Endcard: Initial Conditions Card

B.15.1 Initial Conditions Card Examples

Extracted from a STOMP-W input file:
~Initial Conditions Card
Gas Pressure,Aaqueous Pressure,
2,
Gas Pressure,101325,Pa,,,,,,1,80,1,1,1,66,
Aqueous Pressure,91534.848,Pa,,,,,-9793.519,1/m,1,80,1,1,1,66,

Extracted from a STOMP-W input file:
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
7,
Aqueous Pressure Zonation,9.8614e+4,Pa,bf,
Aqueous Pressure Zonation,9.5260e+4,Pa,ss2,
Aqueous Pressure Zonation,9.7295e+4,Pa,ps2,
Aqueous Pressure Zonation,9.5272e+4,Pa,ss7,
Aqueous Pressure Zonation,9.2080e+4,Pa,ep3,
Gas Pressure,102130.86,Pa,,,,,-12.6549,1/m,1,50,1,50,1,50,
Temperature,16.979,C,,,,,-0.065625,1/m,1,50,1,50,1,50,

Extracted from a STOMP-W-R input file:
~Initial Conditions Card
Aqueous Pressure,Gas Pressure,
6,
Overwrite Species Aqueous Volumetric,H+,0.83889e-6,mol/liter,,,,,,1,10,1,1,1,1,
Overwrite Species Aqueous Volumetric,O2,3.125e-5,mol/liter,,,,,,1,10,1,1,1,1,
Overwrite Species Aqueous Volumetric,Na+,1.0e-3,mol/liter,,,,,,1,10,1,1,1,1,
Overwrite Species Aqueous Volumetric,Cl-,1.0e-3,mol/liter,,,,,,1,10,1,1,1,1,
Overwrite Species Aqueous Volumetric,Biomass,1.36e-4,mol/liter,,,,,,1,10,1,1,1,1,
Overwrite Species Aqueous Volumetric,CO2,4.9e-7,mol/liter,,,,,,1,10,1,1,1,1,

Extracted from a STOMP-WA input file:
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
4,
Temperature,20.0,C,,,,,,1,60,1,1,1,20,
Aqueous Pressure,106173.84,Pa,-1.7554,1/cm,,,-97.9352,1/cm,1,60,1,1,1,20,
Gas Pressure,101331.852,Pa,,,,,-0.11713,1/cm,1,60,1,1,1,20,
Overwrite Solute Aqueous Conc,TCE,200,1/l,,,,,,28,32,1,1,10,13,

Extracted from a STOMP-WAE input file:
#---
~Initial Conditions Card
#---
Aqueous Saturation,Gas Pressure,
2,
Gas Pressure Overwrite,101325,Pa,,,,,,1,1,1,1,1,145,
Temperature Overwrite,20.0,C,,,,,-0.5,1/m,1,1,1,1,1,145,

Initial Conditions Card Examples (Cont'd)

Extracted from a STOMP-WO input file:

```
~Initial Conditions Card
3,
Aqueous Pressure,116015.3,Pa,,,,,-9793.5192,1/m,1,1,1,1,1,2,
NAPL Pressure,116015.3,Pa,,,,,-9793.5192,1/m,1,1,1,1,1,2,
Trapped NAPL Saturation,0.15,,,,,,,,1,1,1,1,1,1,
```

Extracted from a STOMP-WOM input file:

```
~Initial Conditions Card
5,
Aqueous Pressure,98825,Pa,-5000,1/m,,,,,1,20,1,1,1,1,
NAPL Pressure,100500,Pa,,,,,,,1,20,1,1,1,1,
Oil NAPL Mass Fraction,lard-oil mixture,0.3,,0.03,1/m,,,,,1,20,1,1,1,1,
Oil NAPL Mass Fraction,lnapl,0.2,,0.02,1/m,,,,,1,20,1,1,1,1,
Oil NAPL Mass Fraction,brooklawn napl,0.5,,-0.05,1/m,,,,,1,20,1,1,1,1,
```

Extracted from a STOMP-WOA input file:

```
# ------------------------------------------------------------------------
~Initial Conditions Card
# ------------------------------------------------------------------------
10,
Aqueous Pressure, 1.02855E+06,Pa,,,,,,1, 27,1, 34,  1,  1,
Aqueous Pressure, 9.89357E+05,Pa,,,,,,1, 27,1, 34,  2,  2,
Aqueous Pressure, 9.50162E+05,Pa,,,,,,1, 27,1, 34,  3,  3,
Aqueous Pressure, 9.20767E+05,Pa,,,,,,1, 27,1, 34,  4,  4,
Aqueous Pressure, 9.01169E+05,Pa,,,,,,1, 27,1, 34,  5,  5,
Aqueous Pressure, 8.71772E+05,Pa,,,,,,1, 27,1, 34,  6,  6,
Aqueous Pressure, 8.32578E+05,Pa,,,,,,1, 27,1, 34,  7,  7,
Aqueous Pressure, 7.93383E+05,Pa,,,,,,1, 27,1, 34,  8,  8,
Gas Pressure, 1.03234E+05,Pa,,,,,-11.71,1/m,1, 27,1, 34,  1,  85,
NAPL Pressure,-1.e+09,Pa,,,,,,1, 27,1, 34,1, 85,
```

Extracted from a STOMP-WASE input file:

```
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
3,
Gas Pressure,138.0,Bar,,,,,,1,100,1,1,1,50,
Aqueous Pressure,138.0,Bar,,,,,,1,100,1,1,1,50,
Temperature,25.0,C,,,,,,1,100,1,1,1,50,
```

Extracted from a STOMP-WCMSE input file:

```
~Initial Conditions Card
4,
Aqueous Pressure,8.0,MPa,,,,,,1,30,1,1,1,1,
Temperature,2.0,C,,,,,,1,30,1,1,1,1,
Hydrate Saturation,0.5,,,,,,,,1,30,1,1,1,1,
CH4 Mass Fraction of Hydrate Formers,1.0,,,,,,,,1,30,1,1,1,1,
```

B.16 Kinetic Equations Card

Card Title[a] { ~Kinetic Equations [Card] }
Format: *Char*[a],

Number of Kinetic Equations[a],
Format: *Integer*[a],

For: Number of Kinetic Equations

Number of Species in Kinetic Equation[a],
For: Number of Species in Kinetic Equation
Species Name[b], Species Stochiometric Coefficient[c],
Endfor:
Number of Kinetic Reactions in Kinetic Equation[d],
For: Number of Kinetic Reactions in Kinetic Equation
Kinetic Reaction Name[e], Kinetic Reaction Coefficient[f],
Endfor:
Format: *Integer*[a], < *Char*[b], *Real*[c], > *Integer*[d], < *Char*[e], *Real*[f], >

Endfor:

Endcard: Kinetic Equations Card

B.16.1 Kinetic Equations Card Examples

Extracted from a STOMP-W-R input file:

```
~Kinetic Equations Card
4,
1,C5H7O2N,1.00000e+00,
1,KnRc-14,5.76000e-01,
1,Co(ads),1.00000e+00,
1,KnRc-15,1.00000e+00,
1,CoNTA(ads),1.00000e+00,
1,KnRc-16,1.00000e+00,
1,Biomass,1.00000e+00,
1,KnRc-17,1.00000e+00,
```

Extracted from a STOMP-W-R input file:

```
~Kinetic Equations Card
3,
1,C5H7O2N,1.0,
1,KnRc-14,0.576E+00,
1,Co(ads),1.0,
1,KnRc-15,0.100E+01,
1,CoNTA(ads),1.0,
1,KnRc-16,0.100E+01,
```

B.17 Kinetic Reactions Card

Card Title[a] { ~Kinetic Reactions [Card] }
Format: *Char*[a],

Number of Kinetic Reactions[a],
Format: *Integer*[a],

For: Number of Kinetic Reactions

Kinetic Reaction Name[a],
Kinetic Reaction Type Option[b],

{ [Steefel-Lasaga] Dissolution-Precipitation |

Note: $R_k = A_m\, k\left[1-\left(\frac{Q}{K_{eq}}\right)\right];\; k = k_{ref}\left[\frac{-E_a}{R}\left(\frac{1}{T}-\frac{1}{T_{ref}}\right)\right]$

[Smith-Atkins] Forward-Backward |

Note: $R_k = \left[k_f \prod_{i=1}^{N_{reactants}} C_i^{\,e_i} - k_b \prod_{j=1}^{N_{products}} C_j^{\,e_j}\right]$

Valocchi Monod |

Note: $R_k = q_m\, X_m \left(\frac{C_d}{K_d + C_d}\right)\left(\frac{C_a}{K_a + C_a}\right)$

Valocchi Biomass |

Note: $R_k = Y\, q_m\, X_m \left(\frac{C_d}{K_d + C_d}\right)\left(\frac{C_a}{K_a + C_a}\right) - b\, X_m$

Valocchi Sorption }

Note: $R_k = -k_m\left(C_{aq} - \frac{C_{sorb}}{K_d}\right)$

If: Kinetic Reaction Type Option = { [Steefel-Lasaga] Dissolution-Precipitation }
Precipitant Species Name (Mineral)[c],
Format: *Char*[a], *Char*[b], *Char*[c],
Reaction Rate at the Reference Temperature[a] (k_{ref}), Units[b] (mol/m^2 s),

Specific Reactive Surface Area[c] (A_m), Units[d] (m^3/kg aqueous),

Activation Energy[e] (E_a), Units[f] (J/mol),

Reference Temperature[g] (T_{ref}), Units[h] (C),

Format: *Char*[a], *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h],
Equilibrium Constant Coefficient *a*[b],
Equilibrium Constant Coefficient *b*[c],
Equilibrium Constant Coefficient *c*[d],
Equilibrium Constant Coefficient *d*[e],
Equilibrium Constant Coefficient *e*[f],
Format: *Real*[b], *Real*[c], *Real*[d], *Real*[e], *Real*[f],

Note: $\log\left(K_{eq}\right) = a\ln\left(T\right) + b + cT + \frac{d}{T} + \frac{e}{T^2}$

Elseif: Kinetic Reaction Type Option = { [Smith-Atkins] Forward-Backward }
Number of Reactants[c],
For: Number of Reactants
Reactant Species Name[d], Reactant Stochiometric Coefficient[e],
Endfor:
Number of Products[f],
For: Number of Products
Product Species Name[g], Product Stochiometric Coefficient[h],
Endfor:
Format: *Char*[a], *Char*[b], *Integer*[c], < *Char*[d], *Real*[e], [cr] >, *Integer*[f], < *Char*[g], *Real*[h], [cr] >,

Forward Reaction Rate[a], Units[b] (1/s),
Backward Reaction Rate[c], Units[d] (1/s),
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d],

Elseif: Kinetic Reaction Type Option = { Valocchi Biomass }
Number of Reactants (3)[c],
Product Species Name[d], Product Stochiometric Coefficient[e],
Donor Species Name[f], Donor Stochiometric Coefficient[g],
Acceptor Species Name[h], Acceptor Stochiometric Coefficient[i],
Format: *Char*[a], *Char*[b], *Integer*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h], *Real*[i],

Half-Saturation Constant for Donor[a], Units[b] (mol/kg),
Half-Saturation Constant for Acceptor[c], Units[d] (mol/kg),
Maximum Specific Rate of Substrate Utilization[e], Units[f] (1/s),
Microbial Yield Coefficient[g],
Microbial Decay Coefficient[h], Units[i] (1/s)
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Real*[h], *Char*[i],

Elseif: Kinetic Reaction Type Option = { Valocchi Monod }
Number of Reactants (3)[c],

Kinetic Reactions Card (cont'd)

Product Species Name[d], Product Stochiometric Coefficient[e],
Donor Species Name[f], Donor Stochiometric Coefficient[g],
Acceptor Species Name[h], Acceptor Stochiometric Coefficient[i],
Format: *Char*[a], *Char*[b], *Integer*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h], *Real*[i],

Half-Saturation Constant for Donor[a], Units[b] (mol/kg),
Half-Saturation Constant for Acceptor[c], Units[d] (mol/kg),
Maximum Specific Rate of Substrate Utilization[e], Units[f] (1/s),
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f],

Elseif: Kinetic Reaction Type Option = { Valocchi Sorption }
Number of Reactants (1)[c],
Aqueous Species Name[d],
Number of Products (1)[e],
Sorbed Species Name[f],
Format: *Char*[a], *Char*[b], *Integer*[c], *Char*[d], *Integer*[e], *Char*[f],
Mass Transfer Coefficient[a], Units[b] (1/s),
Distribution Coefficient[c], Units[d] (m^3/kg),
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d],
Endif:

Endfor:

Endcard: Kinetic Reactions Card

B.17.1 Kinetic Reactions Card Examples

Extracted from a STOMP-W-R input file:
4,
KnRc-14,Valocchi Monod,3,HNTA--,1.0,O2,1.0,Biomass,1.0,0,
7.64e-7,mol/kg,6.25e-6,mol/kg,1.407e-3,1/hr,
KnRc-15,Valocchi Sorption,1,Co++,1.0,1,Co(ads),1.0,
1.0,1/hr,5.07e-3,L/gm,
KnRc-16,Valocchi Sorption,1,CoNta-,1.0,1,CoNta(ads),1.0,
1.0,1/hr,5.33e-4,L/gm,
KnRc-17,Valocchi Biomass,3,HNta--,1.0,O2,1.0,Biomass,1.0,0,
7.64e-7,mol/kg,6.25e-6,mol/kg,1.407e-3,1/hr,65.14,0.00208,1/hr,

Extracted from a STOMP-W-R input file:
~Kinetic Reactions Card
3,
KnRc-14,Valocchi Monod,HNTA--,O2,C5H7O2N,
7.64e-7,mol/L,6.25e-6,mol/L,0.160398,1/hr,0.57587,0.00208,1/hr,
KnRc-15,Valocchi Sorption,Co++,Co(ads),
1.0,1/hr,5.07e-3,L/gm,
KnRc-16,Valocchi Sorption,CoNTA-,CoNTA(ads),
1.0,1/hr,5.33e-4,L/gm,

Extracted from a STOMP-WCS-R input file:
~Kinetic Reactions Card
1,
R1,Forward-Backward,2,Ca++,1.0,CO3--,1.0,1,CaCO3(s),1.0,
3.3,1/hr,-5.0,1/hr,

B.18 Lithology Card

Card Title[a] { ~Lithology [Card] }
Format: *Char*[a],

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: filename

or the following formattings for binary files:

binary file: filename

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil Types or Scaling Groups
Rock/Soil or Scaling-Group Name[a],
Primary Mineral Specific Area[b], Units[c] (m^2/m^3),
Secondary Mineral Specific Area[d], Units[e] (m^2/m^3),
Format: *Char*[a], *Real*[b], *Char*[c], *Real*[d], *Char*[e],
For: Number of Solid Species
Solid Species Name[a],
Solid Species Volume Fraction of Rock/Soil[b],
Format: *Char*[a], *Real*[b],
Endfor: Number of Solid Species
Endfor: Number of Rock/Soil Types or Scaling Groups

Endif:

EndCard: Lithology Card

B.18.1 Lithology Card

Extracted from a STOMP-WCS-R input file:

```
~Lithology Card
Glauconite Sandstone,1.e5,1/m,250.,1/m,
CaCO3(s),0.0,
```

B.19 Mechanical Properties Card

Card Title[a] { ~Mechanical [Properties Card] }
Format: *Char*[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: Parameter input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. See the example input card in section B.19.1.

Elseif:

For: Number of Rock/Soil or Scaling-Group Types

If: Operational Mode Option = { **STOMP-W w/o Transport** }
{ **STOMP-W-R w/o Transport** } { **STOMP-W-Sc w/o Transport** }
Rock/Soil or Scaling-Group Name[a], Particle Density[b] (2650.0), Units[c] (kg/m^3),
If: Rock/Soil or Scaling-Group Name = {{ Fractured }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Specific Storativity[h], Units[i] (1/m),
Format: *Char*[a], *Real*[b], *Char*[c], *Real*[d], *Real*[e], *Real*[f], *Real*[g], *Real*[h], *Char*[i],
Elseif: Rock/Soil or Scaling-Group Name = {{ DP }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Matrix Specific Storativity[h], Units[i] (1/m),
Fracture Specific Storativity[j], Units[k] (1/m),
Characteristic Length[l], Units[m] (m),
Format: *Char*[a], *Real*[b], *Char*[c], *Real*[d], *Real*[e], *Real*[f], *Real*[g], *Real*[h], *Char*[i],
Real[j], *Char*[k], *Real*[l], *Char*[m],
Else:
Total Porosity[d], Diffusive Porosity[e], Specific Storativity[f], Units[g] (1/m),
Format: *Char*[a], *Real*[b], *Char*[c], *Real*[d], *Real*[e], *Real*[f], *Char*[g],
Endif:

Mechanical Properties Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-W w/ Transport** }
{ **STOMP-W-R w/ Transport** } { **STOMP-W-Sc w/ Transport** }
{ STOMP-WS } { STOMP-WS-Sc }
Rock/Soil or Scaling-Group Name[a], Particle Density[b] (2650.0), Units[c] (kg/m^3),
If: Rock/Soil or Scaling-Group Name = {{ Fractured }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Specific Storativity[h], Units[i] (1/m),
Tortuosity Function Option[j],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[k],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Real[h], Char[i], Char[j], Real[k],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Real[h], Char[i], Char[j],*
Endif:
Elseif: Rock/Soil or Scaling-Group Name = {{ DP }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Matrix Specific Storativity[h], Units[i] (1/m),
Fracture Specific Storativity[j], Units[k] (1/m),
Characteristic Length[l], Units[m] (m),
Tortuosity Function Option[n],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[o],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Real[h], Char[i], Real[j], Char[k], Real[l], Char[m], Char[n], Real[o],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Real[h], Char[i], Real[j], Char[k], Real[l], Char[m], Char[n],*
Endif:
Else:
Total Porosity[d], Diffusive Porosity[e], Specific Storativity[f], Units[g] (1/m),
Tortuosity Function Option[h],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[i],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h], Real[i],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h],*
Endif:
Endif:

Mechanical Properties Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WA** } { **STOMP-WAE** }
{ **STOMP-WAE-B** } { **STOMP-WAE-Sc** } { **STOMP-WAS** }
{ **STOMP-WASE** } { **STOMP-WCS** } { **STOMP-WCS-R** }
{ **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** } { **STOMP-WCMSE** }
Rock/Soil or Scaling-Group Name[a], Particle Density[b] (2650.0), Units[c] (kg/m^3),
If: Rock/Soil or Scaling-Group Name = {{ Fractured }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Specific Storativity[h], Units[i] (1/m),
Tortuosity Function Option[j],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[k], Gas-Phase Tortuosity[l],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],*
Real[g], Real[h], Char[i], Char[j], Real[k], Real[l],
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],*
Real[g], Real[h], Char[i], Char[j],
Endif:
Else:
Total Porosity[d], Diffusive Porosity[e], Specific Storativity[f], Units[g] (1/m),
Tortuosity Function Option[h],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[i], Gas-Phase Tortuosity[j],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h], Real[i], Real[j],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h],*
Endif:
Endif:

Elseif: Operational Mode Option = { **STOMP-WO** } { **STOMP-WOD** }
{ **STOMP-WOM** }
Rock/Soil or Scaling-Group Name[a], Particle Density[b] (2650.0), Units[c] (kg/m^3),
If: Rock/Soil or Scaling-Group Name = {{ Fractured }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Specific Storativity[h], Units[i] (1/m),
Tortuosity Function Option[j],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[k], NAPL-Phase Tortuosity[l],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],*
Real[g], Real[h], Char[i], Char[j], Real[k], Real[l],
Else:

Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Real[h], Char[i], Char[j],*
Endif:
Else:
Total Porosity[d], Diffusive Porosity[e], Specific Storativity[f], Units[g] (1/m),
Tortuosity Function Option[h],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[i], NAPL-Phase Tortuosity[j],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h], Real[i], Real[j],*
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h],*
Endif:
Endif:

Elseif: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc }**
{ STOMP-WOAE }
Rock/Soil or Scaling-Group Name[a], Particle Density[b] (2650.0), Units[c] (kg/m^3),
If: Rock/Soil or Scaling-Group Name = {{ Fractured }}
Matrix Total Porosity[d], Matrix Diffusive Porosity[e],
Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
Specific Storativity[h], Units[i] (1/m),
Tortuosity Function Option[j],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[k], Gas-Phase Tortuosity[l], NAPL-Phase Tortuosity[m],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],*
Real[g], Real[h], Char[i], Char[j], Real[k], Real[l], Real[m],
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Real[h], Char[i], Char[j],*
Endif:
Else:
Total Porosity[d], Diffusive Porosity[e], Specific Storativity[f], Units[g] (1/m),
Tortuosity Function Option[h],
{ Constant | Millington and Quirk }
If: Tortuosity Function Option = { Constant }
Aqueous-Phase Tortuosity[i], Gas-Phase Tortuosity[j], NAPL-Phase Tortuosity[k],
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],*
Char[g], Char[h], Real[i], Real[j], Real[k],
Else:
Format: *Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h],*
Endif:
Endif:
Endif:

Endfor: Number of Rock/Soil or Scaling Group Types

Mechanical Properties Card (cont'd)

Endif:

Endcard: Mechanical Properties Card

B.19.1 Mechanical Properties Card Examples

Extracted from a STOMP-W input file:
~Mechanical Properties Card
IJK Indexing,2690,kg/m^3,file:por.dat,file:por.dat,,1/m,Millington and Quirk,

Extracted from a STOMP-W input file:
~Mechanical Properties Card
SP1,2.63,g/cm^3,0.19,0.19,,,,
SP2,2.63,g/cm^3,0.24,0.24,,,,
SM-ML1,2.63,g/cm^3,0.35,0.35,,,,
SM-SP1,2.63,g/cm^3,0.37,0.37,,,,
SP3,2.63,g/cm^3,0.27,0.27,,,,
SW1,2.63,g/cm^3,0.28,0.28,,,,
US,2.63,g/cm^3,0.96,0.96,,,,

Extracted from a STOMP-W input file:
~Mechanical Properties Card
R1 is a scaling group
R1,,,0.422,0.422,,,Millington and Quirk,

Extracted from a STOMP-WAE input file:
~Mechanical Properties Card
Silt Loam-Gravel Admix,2720,kg/m^3,0.456,0.456,0,,Millington and Quirk,
#Asphaltic Concrete,2630,kg/m^3,0.04,0.04,0,,Millington and Quirk,
Gravel Drainage,2725,kg/m^3,0.419,0.419,0,,Millington and Quirk,
Gravel Filter,2725,kg/m^3,0.419,0.419,0,,Millington and Quirk,
#Backfill,2800,kg/m^3,0.307,0.307,0,,Millington and Quirk,
Riprap,2950,kg/m^3,0.2711,0.2711,0,,Millington and Quirk,
Sand Filter,2755,kg/m^3,0.445,0.445,0,,Millington and Quirk,
Compacted Silt Loam,2720,kg/m^3,0.411,0.411,0,,Millington and Quirk,
#Top Course,2800,kg/m^3,0.2585,0.2585,0,,Millington and Quirk,

Extracted from a STOMP-WCMSE input file:
~Mechanical Properties Card
Medium,2600,kg/m^3,0.3,0.3,Compressibility,5.0e-9,1/Pa,8.0,MPa,Millington and Quirk,,,

B.20 NAPL Relative Permeability Card

Card Title[a] { ~NAPL Rel [ative Permeability Card] }
Format: *Char*[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: Parameter input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. See an example input card in section B.20.1.

Elseif:

For: Number of Rock/Soil Types

Rock/Soil Name[a],
Permeability Function Option[b],
{ Constant | Mualem | Burdine }
If: Permeability Function Option = { Constant }
NAPL Relative Permeability[c],
Format: *Char*[a], *Char*[b], *Real*[c],
Elseif: Permeability Function Option = { Mualem } { Burdine }
If: Saturation Function Option = { van Genuchten }
van Genuchten m parameter[c],
Format: *Char*[a], *Char*[b], *Real*[c],
Elseif: Saturation Function Option = { Brooks and Corey }
Brooks and Corey λ parameter[c],
Format: *Char*[a], *Char*[b], *Real*[c],
Endif:
Endif:

Endfor: Number of Rock/Soil Types

Endif:

Endcard: NAPL Relative Permeability Card

B.20.1 NAPL Relative Permeability Card Examples

Extracted from a STOMP-WO input file:
~NAPL Relative Permeability Card
Sand,Mualem,,

Extracted from a STOMP-WO input file:
~NAPL Relative Permeability
20/30 Ottawa Sand,Constant,0.58,

Extracted from a STOMP-WO input file:
~NAPL Relative Permeability Input Card
Sand,Mualem,0.56,

Extracted from a STOMP-WO input file:
~Gas Relative Permeability Card
IJK Indexing,Constant,file:rel_napl.dat,

Extracted from a STOMP-WO input file:
~NAPL Rel
SM-ML1,Burdine,,
Constant NAPL relative permeability
SW1,Constant,1.0,
SP3,Constant,1.0,
SM-SP1,Burdine,,
SP2,Burdine,,
SP1,Burdine,,
US,Mualem,,

Extracted from a STOMP-WO input file:
~NAPL Relative Permeability Card
pcl,Burdine,,
sm,Burdine,,
sp,Burdine,,
sw,Burdine,,
ml,Burdine,,
acl,Burdine,,
fm,Burdine,,

B.21 Observed Data Card

Card Title[a] { ~Observed Data [Card] }
Format: *Char*[a]

If: Operational Mode Option Card = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
Number of Observed Data Types[a],
Format: *Integer*[a],
If: Execution Mode Option = { Normal w/ Inverse } { Restart w/ Inverse }
Observed Data Type[a], { Field } { Reference } { Surface Flux } { Surface Rate }
{ Surface Integral }

If: Observed Data Type = { Field }
Field Observation Variable[b],
{ Aqueous Pressure | Aqueous Saturation | Aqueous Moisture Content |
Aqueous Hydraulic Head | X Aqueous Volumetric Flux |
Y Aqueous Volumetric Flux | Z Aqueous Volumetric Flux |
Matric Potential | Solute Volumetric Conc. | Solute Aqueous Conc. |
Solute Aqueous Mole frac[tion] | X Solute Flux | Y Solute Flux | Z Solute Flux }
If: Field Observation Variable = { Solute Volumetric Conc. }
{ Solute Aqueous Conc. } { Solute Aqueous Mole frac[tion] }
{ X Solute Flux } { Y Solute Flux } { Z Solute Flux }
Solute Name[c], Field Observation Output Units[d],
Field Observation X-Dir Coordinate[e], Field Observation X-Dir Coordinate Units[f],
Field Observation Y-Dir Coordinate[g], Field Observation X-Dir Coordinate Units[h],
Field Observation Z-Dir Coordinate[i], Field Observation X-Dir Coordinate Units[j]
Observed Data Statistical Index[k], Observed Data Statistic[l],
Observed Data Time Weighting Factor[m], Observed Data Space Weighting Factor[n],
Format: *Char*[a], *Char*[b], *Char*[c],*Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h], *Real*[i], *Char*[j],
Integer[k], *Real*[l], *Real*[m], *Real*[n],
Else:
Field Observation Output Units[c],
Field Observation X-Dir Coordinate[d], Field Observation X-Dir Coordinate Units[e],
Field Observation Y-Dir Coordinate[f], Field Observation X-Dir Coordinate Units[g],
Field Observation Z-Dir Coordinate[h], Field Observation X-Dir Coordinate Units[i],
Observed Data Statistical Index[j], Observed Data Statistic[k],
Observed Data Time Weighting Factor[l], Observed Data Space Weighting Factor[m],
Format: *Char*[a], *Char*[b], *Char*[c], *Real*[d], *Char*[e], *Real*[f], *Char*[g], *Real*[h], *Char*[i], *Integer*[j],
Real[k], *Real*[l], *Real*[m],
Endif:

Elseif: Observed Data Type = { Reference }
Reference Observation Variable[b],
{ Aqueous Pressure | Aqueous Saturation | Aqueous Moisture Content |
Aqueous Hydraulic Head | X Aqueous Volumetric Flux |

Observed Data Card (cont'd)

Y Aqueous Volumetric Flux | Z Aqueous Volumetric Flux |
Matric Potential | Solute Volumetric Conc. | Solute Aqueous Conc. |
Solute Aqueous Mole frac[tion] | X Solute Flux | Y Solute Flux | Z Solute Flux }
If: Reference Observation Variable = { Solute Volumetric Conc. }
{ Solute Aqueous Conc. } { Solute Aqueous Mole frac[tion] }
{ X Solute Flux } { Y Solute Flux } { Z Solute Flux }
Solute Name[c], Reference Observation Output Units[d],
IJK Index 1[e], IJK Index 2[f], IJK Index 3[g],
Observed Data Statistical Index[h], Observed Data Statistic[i],
Observed Data Time Weighting Factor[j], Observed Data Space Weighting Factor[i],
Format: *Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Real[h], Real[i], Real[j],*
Else:
Reference Observation Output Units[c],
IJK Index 1[c], IJK Index 2[d], IJK Index 3[e],
Observed Data Statistical Index[f], Observed Data Statistic[g],
Observed Data Time Weighting Factor[h], Observed Data Space Weighting Factor[i],
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Real[g], Real[h], Real[i],*
Endif:

Elseif: Observed Data Type = { Surface Flux | Surface Rate }
Surface Rate Observation Variable[b],
{ Aqueous Volumetric Flux | Aqueous Mass Flux | Solute Flux }
If: Surface Rate Observation Variable = { Solute Flux }
Solute Name[c], Surface Rate Observation Output Units[d],
Surface Rate Observation Orientation[d],
I-Start Domain Index[f], I-End Domain Index[g],
J-Start Domain Index[h], J-End Domain Index[i],
K-Start Domain Index[j], K-End Domain Index[k],
Observed Data Statistical Index[l], Observed Data Statistic[m],
Observed Data Time Weighting Factor[n], Observed Data Space Weighting Factor[o],
Format: *Char[a], Char[b], Char[c], Char[d], Char[e], Integer[f], Integer[g], Integer[h],*
Integer[i], Integer[j], Integer[k], Integer[l], Real[m], Real[n], Real[o],
Else:
Surface Rate Observation Output Units[c],
Surface Rate Observation Orientation[d],
I-Start Domain Index[e], I-End Domain Index[f],
J-Start Domain Index[g], J-End Domain Index[h],
K-Start Domain Index[i], K-End Domain Index[j],
Observed Data Statistical Index[k], Observed Data Statistic[l],
Observed Data Time Weighting Factor[m], Observed Data Space Weighting Factor[n],
Format: *Char[a], Char[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Integer[i], Integer[j], Integer[k], Real[l], Real[m], Real[n],
Endif:

Else: Observed Data Type = { Surface Integral }
{ Aqueous Volumetric Flux Integral | Aqueous Mass Flux Integral |

Observed Data Card (cont'd)

Solute Flux Integral }
If: Surface Rate Observation Variable = { Solute Integral }
Solute Name[c], Surface Integral Observation Output Units[d],
Surface Integral Observation Orientation[d],
I-Start Domain Index[f], I-End Domain Index[g],
J-Start Domain Index[h], J-End Domain Index[i],
K-Start Domain Index[j], K-End Domain Index[k],
Observed Data Statistical Index[l], Observed Data Statistic[m],
Observed Data Time Weighting Factor[n], Observed Data Space Weighting Factor[o],
Format: *Char[a], Char[b], Char[c], Char[d], Char[e], Integer[f], Integer[g], Integer[h],*
Integer[i], Integer[j], Integer[k], Integer[l], Real[m], Real[n], Real[o],
Else:
Surface Integral Observation Output Units[c],
Surface Integral Observation Orientation[d],
I-Start Domain Index[e], I-End Domain Index[f],
J-Start Domain Index[g], J-End Domain Index[h],
K-Start Domain Index[i], K-End Domain Index[j],
Observed Data Statistical Index[k], Observed Data Statistic[l],
Observed Data Time Weighting Factor[m], Observed Data Space Weighting Factor[n],
Format: *Char[a], Char[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Integer[i], Integer[j], Integer[k], Real[l], Real[m], Real[n],
Endif:
Endif:

If: Observed Data Read in from External File
File[a] (File), Filename[b], External File Time Units[c], External File Variable Units[d],
Format: *Char[a], Char[b], Char[c], Char[d],*
Elseif: Observed Data Read in from Input file
Number of Observed Data Samples[a],
Format: Integer[a],
Time[a], Time Units[b], Value[c], Value Units[d],
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endif:

Endif:

Observed Data Card (cont'd)

If: Operational Mode Option Card = { **STOMP-WAE** } { **STOMP-WAE-B** } { **STOMP-WAE-Sc** }
Number of Observed Data Types[a],
Format: *Integer*[a],
If: Execution Mode Option = { Normal w/ Inverse } { Restart w/ Inverse }
Observed Data Type[a], { Field } { Reference } { Surface Flux } { Surface Rate }
{ Surface Integral }
If: Observed Data Type = { Field }
Field Observation Variable[b],
{ Aqueous Pressure | Aqueous Saturation | Aqueous Moisture Content |
Aqueous Hydraulic Head | X Aqueous Volumetric Flux |
Y Aqueous Volumetric Flux | Z Aqueous Volumetric Flux |
Matric Potential | Solute Volumetric Conc. | Solute Aqueous Conc. |
Solute Aqueous Mole frac[tion] | X Solute Flux | Y Solute Flux | Z Solute Flux |
Atmospheric Temperature | Atmospheric Relative Humidity |
Atmospheric Solar Radiation | Atmospheric Wind Speed |
Grond Surface Temperature | Ground Surface Water-Vapor Pressure |
Actual Evaporation Rate | Potential Evaporation Rate | Actual Transpiration Rate |
Potential Transpiration Rate | Gas Pressure | Gas Saturation |
Water Gas Mass Fraction | Air Gas Mass Fraction | Water Aqueous Mass Fraction |
Air Aqueous Mass Fraction | Gas Hydraulic Head | Aqueous Relative Permeability |
Gas Relative Permeability | Aqueous Density | Gas Density |
X-Dir. Effective Thermal Conductivity | Y-Dir. Effective Thermal Conductivity |
Z-Dir. Effective Thermal Conductivity | X-Dir. Aqueous Volumetric Flux |
Y-Dir. Aqueous Volumetric Flux | Z-Dir. Aqueous Volumetric Flux |
X-Dir. Gas Volumetric Flux | Y-Dir. Gas Volumetric Flux |
Z-Dir. Gas Volumetric Flux | X-Dir. Heat Flux | Y-Dir. Heat Flux |
Z-Dir. Heat Flux | Water Gas Concentration | Air Gas Concentration |
Water Aqueous Concentration | Air Aqueous Concentration |
Solute Gas Concentration | Solute Gas Mole Fraction }
If: Field Observation Variable = {{ Concentration | Fraction | Solute Flux }}
Solute Name[c], Field Observation Output Units[d],
Field Observation X-Dir Coordinate[e], Field Observation X-Dir Coordinate Units[f],
Field Observation Y-Dir Coordinate[g], Field Observation X-Dir Coordinate Units[h],
Field Observation Z-Dir Coordinate[i], Field Observation X-Dir Coordinate Units[j]
Observed Data Statistical Index[k], Observed Data Statistic[l],
Observed Data Time Weighting Factor[m], Observed Data Space Weighting Factor[n],
Format: *Char*[a], *Char*[b], *Char*[c],*Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h], *Real*[i], *Char*[j],
Integer[k], *Real*[l], *Real*[m], *Real*[n],
Else:
Field Observation Output Units[c],
Field Observation X-Dir Coordinate[d], Field Observation X-Dir Coordinate Units[e],
Field Observation Y-Dir Coordinate[f], Field Observation X-Dir Coordinate Units[g],
Field Observation Z-Dir Coordinate[h], Field Observation X-Dir Coordinate Units[i],
Observed Data Statistical Index[j], Observed Data Statistic[k],
Observed Data Time Weighting Factor[l], Observed Data Space Weighting Factor[m],
Format: *Char*[a], *Char*[b], *Char*[c], *Real*[d], *Char*[e], *Real*[f], *Char*[g], *Real*[h], *Char*[i], *Integer*[j],

Real[k], Real[l], Real[m],
Endif:

Elseif: Observed Data Type = { Reference }
Reference Observation Variable[b],
{ Aqueous Pressure | Aqueous Saturation | Aqueous Moisture Content |
Aqueous Hydraulic Head | X Aqueous Volumetric Flux |
Y Aqueous Volumetric Flux | Z Aqueous Volumetric Flux |
Matric Potential | Solute Volumetric Conc. | Solute Aqueous Conc. |
Solute Aqueous Mole frac[tion] | X Solute Flux | Y Solute Flux | Z Solute Flux |
Atmospheric Temperature | Atmospheric Relative Humidity |
Atmospheric Solar Radiation | Atmospheric Wind Speed |
Grond Surface Temperature | Ground Surface Water-Vapor Pressure |
Actual Evaporation Rate | Potential Evaporation Rate | Actual Transpiration Rate |
Potential Transpiration Rate | Gas Pressure | Gas Saturation |
Water Gas Mass Fraction | Air Gas Mass Fraction | Water Aqueous Mass Fraction |
Air Aqueous Mass Fraction | Gas Hydraulic Head | Aqueous Relative Permeability |
Gas Relative Permeability | Aqueous Density | Gas Density |
X-Dir. Effective Thermal Conductivity | Y-Dir. Effective Thermal Conductivity |
Z-Dir. Effective Thermal Conductivity | X-Dir. Aqueous Volumetric Flux |
Y-Dir. Aqueous Volumetric Flux | Z-Dir. Aqueous Volumetric Flux |
X-Dir. Gas Volumetric Flux | Y-Dir. Gas Volumetric Flux |
Z-Dir. Gas Volumetric Flux | X-Dir. Heat Flux | Y-Dir. Heat Flux |
Z-Dir. Heat Flux | Water Gas Concentration | Air Gas Concentration |
Water Aqueous Concentration | Air Aqueous Concentration |
Solute Gas Concentration | Solute Gas Mole Fraction }
If: Reference Observation Variable = {{ Concentration | Fraction | Solute Flux }}
Solute Name[c], Reference Observation Output Units[d],
IJK Index 1[e], IJK Index 2[f], IJK Index 3[g],
Observed Data Statistical Index[h], Observed Data Statistic[i],
Observed Data Time Weighting Factor[j], Observed Data Space Weighting Factor[i],
Format: *Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Real[h], Real[i], Real[j],*
Else:
Reference Observation Output Units[c],
IJK Index 1[c], IJK Index 2[d], IJK Index 3[e],
Observed Data Statistical Index[f], Observed Data Statistic[g],
Observed Data Time Weighting Factor[h], Observed Data Space Weighting Factor[i],
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Real[g], Real[h], Real[i],*
Endif:

Elseif: Observed Data Type = { Surface Flux | Surface Rate }
Surface Rate Observation Variable[b],
{ Aqueous Volumetric Flux | Aqueous Mass Flux | Solute Flux | Heat Flux }
If: Surface Rate Observation Variable = { Solute Flux }
Solute Name[c], Surface Rate Observation Output Units[d],
Surface Rate Observation Orientation[d],

Observed Data Card (cont'd)

I-Start Domain Index[f], I-End Domain Index[g],
J-Start Domain Index[h], J-End Domain Index[i],
K-Start Domain Index[j], K-End Domain Index[k],
Observed Data Statistical Index[l], Observed Data Statistic[m],
Observed Data Time Weighting Factor[n], Observed Data Space Weighting Factor[o],
Format: *Char[a], Char[b], Char[c], Char[d], Char[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j], Integer[k], Integer[l], Real[m], Real[n], Real[o],*

Else:

Surface Rate Observation Output Units[c],
Surface Rate Observation Orientation[d],
I-Start Domain Index[e], I-End Domain Index[f],
J-Start Domain Index[g], J-End Domain Index[h],
K-Start Domain Index[i], K-End Domain Index[j],
Observed Data Statistical Index[k], Observed Data Statistic[l],
Observed Data Time Weighting Factor[m], Observed Data Space Weighting Factor[n],
Format: *Char[a], Char[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j], Integer[k], Real[l], Real[m], Real[n],*

Endif:

Else: Observed Data Type = { Surface Integral }
{ Aqueous Volumetric Flux Integral | Aqueous Mass Flux Integral | Solute Flux Integral }

If: Surface Rate Observation Variable = { Solute Integral }

Solute Name[c], Surface Integral Observation Output Units[d],
Surface Integral Observation Orientation[d],
I-Start Domain Index[f], I-End Domain Index[g],
J-Start Domain Index[h], J-End Domain Index[i],
K-Start Domain Index[j], K-End Domain Index[k],
Observed Data Statistical Index[l], Observed Data Statistic[m],
Observed Data Time Weighting Factor[n], Observed Data Space Weighting Factor[o],
Format: *Char[a], Char[b], Char[c], Char[d], Char[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j], Integer[k], Integer[l], Real[m], Real[n], Real[o],*

Else:

Surface Integral Observation Output Units[c],
Surface Integral Observation Orientation[d],
I-Start Domain Index[e], I-End Domain Index[f],
J-Start Domain Index[g], J-End Domain Index[h],
K-Start Domain Index[i], K-End Domain Index[j],
Observed Data Statistical Index[k], Observed Data Statistic[l],
Observed Data Time Weighting Factor[m], Observed Data Space Weighting Factor[n],
Format: *Char[a], Char[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j], Integer[k], Real[l], Real[m], Real[n],*

Endif:

Endif:

Observed Data Card (cont'd)

If: Observed Data Read in from External File
File[a] (File), Filename[b], External File Time Units[c], External File Variable Units[d],
Format: *Char[a], Char[b], Char[c], Char[d],*
Elseif: Observed Data Read in from Input file
Number of Observed Data Samples[a],
Format: Integer[a],
Time[a], Time Units[b], Value[c], Value Units[d],
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endif:

Endif:

Endcard: Observed Data Card

B.21.1 Observed Data Card Examples

Extracted from a STOMP-WAE input file:

```
~Observed Data Card
2,
field,aqueous moisture content,,0.5,cm,0.5,cm,165.5,cm,1,0.01,0.8,0.95,
21,
0,s,0.218,,
484,s,0.213,,
1080,s,0.213,,
1680,s,0.204,,
2880,s,0.204,,
4080,s,0.205,,
5280,s,0.207,,
7080,s,0.204,,
8880,s,0.209,,
13100,s,0.201,,
16700,s,0.201,,
19700,s,0.205,,
68900,s,0.2,,
99300,s,0.196,,
187000,s,0.191,,
427000,s,0.174,,
618000,s,0.164,,
767000,s,0.159,,
1030000,s,0.151,,
1380000,s,0.143,,
1980000,s,0.136,,
field,matric potential,cm,0.5,cm,0.5,cm,60.5,cm,1,4.0,0.8,0.8,
21,
0,s,-1,cm,
484,s,-1,cm,
1080,s,-2,cm,
1680,s,-3,cm,
2880,s,-5,cm,
4080,s,-7,cm,
5280,s,-9,cm,
7080,s,-10,cm,
8880,s,-11,cm,
13100,s,-14,cm,
16700,s,-16,cm,
19700,s,-16,cm,
68900,s,-20,cm,
99300,s,-20,cm,
187000,s,-23,cm,
427000,s,-24,cm,
618000,s,-26,cm,
767000,s,-27,cm,
1030000,s,-28,cm,
1380000,s,-31,cm,
1980000,s,-33,cm,
```

B.22 Oil Properties Card

Card Title[a] { ~Volatile Organic Compound Prop [erties Card] }
{ ~Organic Compound Prop [erties Card] }
{ ~VOC Prop [erties Card] } { ~Oil Prop [erties Card] }
Format: *Char[a]*

If: Operational Mode Option = **{ STOMP-WO } { STOMP-WO-Sc } { STOMP-WOD } { STOMP-WOA } { STOMP-WOA-Sc }**

Oil Name[a],
Format: *Char[a],*

Molecular Weight[a], Units[b] (kgmol/kg),
Freezing Point Temperature[c], Units[d] (K),
Normal BOiling Point[e], Units[f] (K),
Critical Temperature[g], Units[h] (K),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*

Critical Pressure[a], Units[b] (bar),
Critical Molar Volume[c], Units[d] (cm^3/mole),
Critical Compressibility[e],
Pitzner Acentric Factor[f],
Dipole Moment[g], Units[h] (Debyes),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h],*

Note: $\bar{c}^{\,o}_{pn} = a + bT + cT^2 + d\,T^3, \ \frac{J}{mol\,K}$

Isobaric Molar Specific Heat Constants a[a],
Isobaric Molar Specific Heat Constants b[b],
Isobaric Molar Specific Heat Constants c[c],
Isobaric Molar Specific Heat Constants d[d],
Format: *Real[a], Real[b], Real[c], Real[d],*

Saturated Vapor Pressure Function Option[a]

{ Equation 1 } **Note:** $\ln\left(\frac{P_g^o}{P_c^o}\right) = \frac{\left(ax + bx^{1.5} + cx^3 + d\,x^6\right)}{(1-x)}$ *where,* $x = 1 - \frac{T}{T_c^o}$

{ Equation 2 } **Note:** $\ln\left(P_g^o\right) = a - \frac{b}{T} + c\ln(T) + d\left(\frac{P_g^o}{T^2}\right)$

{ Equation 3 } **Note:** $\ln\left(P_g^o\right) = a - \frac{b}{(c+T)}$

Oil Properties Card (cont'd)

{ Constant } **Note:** $P_g^o = \overline{P}_g^o$

If: Saturated Vapor Pressure Function Option = { Equation 1 }
Constant a[b], Constant b[c], Constant c[d], Constant d[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Elseif: Saturated Vapor Pressure Function Option = { Equation 2 }
Constant a[b], Constant b[c], Constant c[d], Constant d[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Elseif: Saturated Vapor Pressure Function Option = { Equation 3 }
Constant a[b], Constant b[c], Constant c[d],
Format: *Char[a], Real[b], Real[c], Real[d],*
Elseif: Saturated Vapor Pressure Function Option = { Constant }
Saturated Oil Vapor Pressure[b], Units[c] (Pa),
Format: *Char[a], Real[b], Char[c],*
Endif:

Liquid Density Function Option[a]

{ HBT Technique } **Note:** [Reid et al. 1987, pp. 55-66]

{ Modified Rackett w/Reference } **Note:** [Reid et al. 1987, pp. 67]

{ Modified Rackett } **Note:** [Reid et al. 1987, pp. 67]

{ Constant } **Note:** $\rho_n^o = \overline{\rho}_n^o$

If: Liquid Density Function Option = { HBT Technique }
Pure Component Characteristic Volume[b], Units[c] (L/mol),
HBT Acentric Factor[d],
Format: *Char[a], Real[b], Char[c], Real[d],*
Elseif: Liquid Density Function Option = { Modified Rackett w/ Reference }
Rackett Compressibility Factor[b],
HBT Acentric Factor[c],
Reference Liquid Density[d], Units[e] (kg/m^3),
Reference Temperature[f], Units[g] (C),
Format: *Char[a], Real[b], Real[c], Real[d], Char[e], Real[f], Char[g],*
Elseif: Liquid Density Function Option = { Modified Rackett }
Rackett Compressibility Factor[b],
HBT Acentric Factor[c],
Format: *Char[a],Real[b],Real[c],*
Elseif: Liquid Density Function Option = { Constant }
Reference Liquid Density[b], Units[c] (kg/m^3),
Format: *Char[a],Real[b],Char[c],*
Endif:

Oil Properties Card (cont'd)

Liquid Viscosity Function Option[a]

{ Reference } **Note:** $\mu_n^o = \left[\left(\overline{\mu}_n^o \right)^{-0.2661} + \frac{\left(T - \overline{T}_n^o \right)}{233.0} \right]^{-3.758}$ *in cP and K*

{ Equation 1 } **Note:** $\mu_n^o = aT^b$ *in cP and K*

{ Equation 2 } **Note:** $\mu_n^o = \exp\left(a + \frac{b}{T} \right)$ *in cP and K*

{ Equation 3 } **Note:** $\mu_n^o = \exp\left(a + \frac{b}{T} + cT + dT^2 \right)$ *in cP and K*

{ Constant } **Note:** $\mu_n^o = \overline{\mu}_n^o$

If: Liquid Viscosity Function Option = { Reference }
Reference Liquid Viscosity[b], Units[c] (Pa s),
Reference Temperature[d], Units[e] (C),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Elseif: Liquid Viscosity Function Option = { Equation 1 }
Constant a[b], Constant b[c],
Format: *Char[a], Real[b], Real[c],*
Elseif: Liquid Viscosity Function Option = { Equation 2 }
Constant a[b], Constant b[c],
Format: *Char[a], Real[b], Real[c],*
Elseif: Liquid Viscosity Function Option = { Equation 3 }
Constant a[b], Constant b[c],
Constant c[d], Constant d[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Elseif: Liquid Viscosity Function Option = { Constant }
Reference Liquid Viscosity[b], Units[c] (Pa s),
Format: *Char[a], Real[b], Char[c],*
Endif:

Henry's Constant for Aqueous Solubility[a] (1.e+20), Units[b] (Pa),
Format: *Real[a],Char[b],*

Oil Properties Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WOM }**

Number of NAPL Components[a],
Format: Integer[a],

For: Number of NAPL Components

Oil Name[a],
Format: *Char[a],*

Molecular Weight[a], Units[b] (kgmol/kg),
Freezing Point Temperature[c], Units[d] (K),
Normal BOiling Point[e], Units[f] (K),
Critical Temperature[g], Units[h] (K),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*

Critical Pressure[a], Units[b] (bar),
Critical Molar Volume[c], Units[d] (cm^3/mole),
Critical Compressibility[e],
Pitzner Acentric Factor[f],
Dipole Moment[g], Units[h] (Debyes),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h],*

Note: $\bar{c}^{o}_{pn} = a + bT + cT^2 + d\,T^3,\ \frac{J}{mol\,K}$

Isobaric Molar Specific Heat Constants a[a],
Isobaric Molar Specific Heat Constants b[b],
Isobaric Molar Specific Heat Constants c[c],
Isobaric Molar Specific Heat Constants d[d],
Format: *Real[a], Real[b], Real[c], Real[d],*

Saturated Vapor Pressure Function Option[a]

{ Equation 1 } **Note:** $\ln\left(\frac{P_g^o}{P_c^o}\right) = \frac{\left(ax + bx^{1.5} + cx^3 + dx^6\right)}{(1-x)}$ *where,* $x = 1 - \frac{T}{T_c^o}$

{ Equation 2 } **Note:** $\ln\left(P_g^o\right) = a - \frac{b}{T} + c\ln(T) + d\left(\frac{P_g^o}{T^2}\right)$

{ Equation 3 } **Note:** $\ln\left(P_g^o\right) = a - \frac{b}{(c+T)}$

{ Constant } **Note:** $P_g^o = \overline{P}_g^o$

Oil Properties Card (cont'd)

If: Saturated Vapor Pressure Function Option = { Equation 1 }
Constant a[b], Constant b[c], Constant c[d], Constant d[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Elseif: Saturated Vapor Pressure Function Option = { Equation 2 }
Constant a[b], Constant b[c], Constant c[d], Constant d[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Elseif: Saturated Vapor Pressure Function Option = { Equation 3 }
Constant a[b], Constant b[c], Constant c[d],
Format: *Char[a], Real[b], Real[c], Real[d],*
Elseif: Saturated Vapor Pressure Function Option = { Constant }
Saturated Oil Vapor Pressure[b], Units[c] (Pa),
Format: *Char[a], Real[b], Char[c],*
Endif:

Liquid Density Function Option[a] ,

{ HBT Technique } **Note:** [Reid et al. 1987, pp. 55-66]

{ Modified Rackett w/Reference } **Note:** [Reid et al. 1987, pp. 67]

{ Modified Rackett } **Note:** [Reid et al. 1987, pp. 67]

{ Constant } **Note:** $\rho_n^o = \overline{\rho}_n^o$

If: Liquid Density Function Option = { HBT Technique }
Pure Component Characteristic Volume[b], Units[c] (L/mol),
HBT Acentric Factor[d],
Format: *Char[a], Real[b], Char[c], Real[d],*
Elseif: Liquid Density Function Option = { Modified Rackett w/ Reference }
Rackett Compressibility Factor[b],
HBT Acentric Factor[c],
Reference Liquid Density[d], Units[e] (kg/m^3),
Reference Temperature[f], Units[g] (C),
Format: *Char[a], Real[b], Real[c], Real[d], Char[e], Real[f], Char[g],*
Elseif: Liquid Density Function Option = { Modified Rackett }
Rackett Compressibility Factor[b],
HBT Acentric Factor[c],
Format: *Char[a],Real[b],Real[c],*
Elseif: Liquid Density Function Option = { Constant }
Reference Liquid Density[b], Units[c] (kg/m^3),
Format: *Char[a],Real[b],Char[c],*
Endif:

Oil Properties Card (cont'd)

Liquid Viscosity Function Option[a]

{ Reference } **Note:** $\mu_n^o = \left[\left(\overline{\mu}_n^o \right)^{-0.2661} + \frac{\left(T - \overline{T}_n^o \right)}{233.0} \right]^{-3.758}$ *in cP and K*

{ Equation 1 } **Note:** $\mu_n^o = aT^b$ *in cP and K*

{ Equation 2 } **Note:** $\mu_n^o = \exp\left(a + \frac{b}{T} \right)$ *in cP and K*

{ Equation 3 } **Note:** $\mu_n^o = \exp\left(a + \frac{b}{T} + cT + dT^2 \right)$ *in cP and K*

{ Constant } **Note:** $\mu_n^o = \overline{\mu}_n^o$

If: Liquid Viscosity Function Option = { Reference }
Reference Liquid Viscosity[b], Units[c] (Pa s),
Reference Temperature[d], Units[e] (C),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Elseif: Liquid Viscosity Function Option = { Equation 1 }
Constant a[b], Constant b[c],
Format: *Char[a], Real[b], Real[c],*
Elseif: Liquid Viscosity Function Option = { Equation 2 }
Constant a[b], Constant b[c],
Format: *Char[a], Real[b], Real[c],*
Elseif: Liquid Viscosity Function Option = { Equation 3 }
Constant a[b], Constant b[c],
Constant c[d], Constant d[e],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e],*
Elseif: Liquid Viscosity Function Option = { Constant }
Reference Liquid Viscosity[b], Units[c] (Pa s),
Format: *Char[a], Real[b], Char[c],*
Endif:

Henry's Constant for Aqueous Solubility[a] (1.e+20), Units[b] (Pa),
Format: *Real[a],Char[b],*

Endfor:

Endif

Endcard: Oil Properties Card

B.22.1 Oil Properties Card Examples

Extracted from a STOMP-WO input file:
~Volatile Organic Compound Properties Card
Test NAPL,
170.34,g/mol,-9.55,C,225.35,C,385.05,C,
18.2,bar,713,cm^3/mol,0.24,0.0,0,debyes,
-9.328,1.149,-0.0006347,1.359e-07,
Equation 2,77.628,10012.5,-9.236,10030.0,
Constant,840,kg/m^3,
Constant,0.0047,Pa s,
1.0e10,Pa,

Extracted from a STOMP-WO input file:
~Oil Properties Card
brooklawn napl, (hexachloro-1,3-butadiene)
260.7602,g/mol,164.2,K,268.7,K,425.0,K,
43.3,bar,221.0,cm^3/mol,0.270,0.195,0.0,debyes,
-1.687e+0,3.419e-1,-2.340e-4,6.335e-8,
Equation 5,0.20892e+2,0.30247e+04,-0.64044e+02,(1,1,2-trichloroethane)
Constant,1.39,g/cm^3,
Constant,32.9,cP,
1.582e10,Pa,(gives a solubility of 2 gm/liter)

Extracted from a STOMP-WOM input file:
~Oil Component Properties Card
2,
lnapl,
165.834,g/mol,251.,K,394.4,K,620.2,K,
47.6,bar,289.6,cm^3/mol,0.2758,0.2515,0.0,debyes,
-1.431e+1,5.506e-1,-4.513e-4,1.429e-7,
Equation 1,-7.36067,1.82732,-3.47735,-1.00033,
Constant,0.800,g/cm^3,
Constant,2.0e-3,Pa s,
1.0e+8,Pa,
Carbontetrachloride,
165.834,g/mol,251.,K,394.4,K,620.2,K,
47.6,bar,289.6,cm^3/mol,0.2758,0.2515,0.0,debyes,
-1.431e+1,5.506e-1,-4.513e-4,1.429e-7,
Equation 1,-7.36067,1.82732,-3.47735,-1.00033,
Constant,1594,kg/m^3,
Constant,0.97e-3,Pa s,
1.3062e+8,Pa,

Extracted from a STOMP-WOM input file:
~Oil Component Properties Card
3,
lard-oil mixture,
260.7602,g/mol,164.2,K,268.7,K,425.0,K,
43.3,bar,221.0,cm^3/mol,0.270,0.195,0.0,debyes,
-1.687e+0,3.419e-1,-2.340e-4,6.335e-8,

Equation 5,0.20892e+2,0.30247e+04,-0.64044e+02,(1,1,2-trichloroethane)

Constant,1.2,g/cm^3,
Constant,1.0,cP,
1.0e9,Pa,
lnapl,
165.834,g/mol,251.,K,394.4,K,620.2,K,
47.6,bar,289.6,cm^3/mol,0.2758,0.2515,0.0,debyes,
-1.431e+1,5.506e-1,-4.513e-4,1.429e-7,
Equation 1,-7.36067,1.82732,-3.47735,-1.00033,
Constant,0.800,g/cm^3,
Constant,2.0e-3,Pa s,
1.0e+10,Pa,
brooklawn napl, (hexachloro-1,3-butadiene)
260.7602,g/mol,164.2,K,268.7,K,425.0,K,
43.3,bar,221.0,cm^3/mol,0.270,0.195,0.0,debyes,
-1.687e+0,3.419e-1,-2.340e-4,6.335e-8,
Equation 5,0.20892e+2,0.30247e+04,-0.64044e+02,(1,1,2-trichloroethane)
Constant,1.39,g/cm^3,0.8,g/cm^3,1.39,g/cm^3,
Constant,32.9,cP,
1.582e10,Pa,(gives a solubility of 2 gm/liter)

Extracted from a STOMP-WOA input file:
~Volatile Organic Compound Properties Card
Carbontetrachloride,
165.834,g/mol,251.,K,394.4,K,620.2,K,
47.6,bar,289.6,cm^3/mol,0.2758,0.2515,0.0,debyes,
-1.431e+1,5.506e-1,-4.513e-4,1.429e-7,
Equation 1,-7.36067,1.82732,-3.47735,-1.00033,
Constant,1594,kg/m^3,
Constant,0.97e-3,Pa s,
1.3062e+8,Pa,

Extracted from a STOMP-WOA input file:

~Volatile Organic Compound Properties Card

Hanford Site Z9 carbon tetrachloride mixture,
153.82,g/mol,250.,K,349.9,K,556.4,K,
45.6,bar,275.9,cm^3/mol,0.272,0.193,0.0,debyes,
4.072e+1,2.0496e-1,-2.27e-4,8.843e-8,
Constant,10830,Pa,
Constant,1426,kg/m^3,
Constant,1.11e-3,Pa s,
1.3062e8,Pa,

Extracted from a STOMP-WOA input file:
~Oil Properties Card
PCE,
165.834,g/mol,251.,K,394.4,K,620.2,K,

Oil Properties Card Examples (cont'd)

47.6,bar,289.6,cm^3/mol,0.2758,0.2515,0.0,debyes,
-1.431e+1,5.506e-1,-4.513e-4,1.429e-7,
Equation 1,-7.36067,1.82732,-3.47735,-1.00033,
Constant,1.623,g/cm^3,
Constant,0.89e-3,Pa s,
1.13009e+08,Pa,

B.23 Output Control Card

Card Title[a] { ~Output [Control Card] }
Format: *Char[a]*

Number of Reference Nodes[a],
Format: *Integer[a],*

For: Number of Reference Nodes
I Index[a], J Index[b], K Index[c],
Endfor: Number of Reference Nodes
Format: *Integer[a], Integer[b], Integer[c],*

Reference Node Screen Output Frequency[a],
Reference Node Output File Frequency[b],
Output Time Units[c] (s),
Output Length Units[d] (m),
Screen Significant Digits[e],
Output File Significant Digits[f],
Plot File Significant Digits[g]
Format: *Integer[a], Integer[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g],*

Number of Reference Node Variables[a],
Format: *Integer[a],*

For: Number of Reference Node Variables
Reference Node Variable Option[a], Reference Node Variable Units[b],
Format: *Real[a], Char[b],*
Endfor: Number of Reference Node Variables

Note: Refer to following pages for Reference Node Variable Options and Units.

Number of Plot File Times[a]
Format: *Integer[a],*

For: Number of Plot File Times
Plot File Output Time[a], Units[b] (s)
Format: *Real[a], Char[b],*
Endfor: Number of Plot File Times

Number of Plot File Variables[a]
Format: *Integer[a],*

For: Number of Plot File Variables
Plot File Variable Option[a], Plot File Variable Units[b],
Format: *Char[a], Char[b],*

Output Control Card (cont'd)

Endfor: Number of Plot File Variables

Note: Refer to the following pages for Plot File Variable Options and Units.

Endcard: Output Control Card

Reference Node Variable and Plot File Variable Options

If: Operational Mode Option = **{ STOMP-W } { STOMP-W-Sc }**
{ apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture saturation | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix saturation | aqueous moisture cont[ent] | aqueous pressure |
aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
aqueous well depth*P | diffusive porosity | effective trapped gas saturation | gas gauge pressure |
gas pressure | gas saturation | integrated water mass* | matric potential | matrix pressure |
phase condition | rock/soil type | scanning path |solute aqueous conc[entration] |
solute aqueous mole frac[tion] | solute integrated mass | solute integrated aqueous |
solute inventory | solute source int[egral] |solute volumetric conc[entration] |
temperature | total water mass | trapped gas saturation | well flow integral*P | well flow rate*P |
water aqueous mass fraction | water mass source int[egral] | water mass source rate |
x aqueous relative permeability |
x aqueous vol[umetric flux] | x solute flux | xnc aqueous vol[umetric flux (node centered)] |
y aqueous relative permeability |
y aqueous vol[umetric flux] | y solute flux | ync aqueous vol[umetric flux (node centered)] |
y aqueous relative permeability |
z aqueous vol[umetric flux] | z solute flux | znc aqueous vol[umetric flux (node centered)] }

* Reference Node Variable Only
P Proprietary

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

If: Operational Mode Option = **{ STOMP-W-R }**
{ apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture saturation | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix saturation | aqueous moisture cont[ent] | aqueous pressure |
aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
aqueous well depth*P | diffusive porosity | effective trapped gas saturation | gas gauge pressure
|
gas pressure | gas saturation | integrated water mass* | matric potential | matrix pressure |
phase condition | rock/soil type | scanning path |solute aqueous conc[entration] |
solute aqueous mole frac[tion] | solute integrated mass | solute integrated aqueous |
solute inventory | solute source int[egral] |solute volumetric conc[entration] |
species aqueous conc[entration] | species volumetric conc[entration] |
species source | species integrated mass |
temperature | total water mass | trapped gas saturation | well flow integral*P | well flow rate*P |
water aqueous mass fraction | water mass source int[egral] | water mass source rate |
x aqueous relative permeability |
x aqueous vol[umetric flux] | x solute flux | xnc aqueous vol[umetric flux (node centered)] |
y aqueous relative permeability |
y aqueous vol[umetric flux] | y solute flux | ync aqueous vol[umetric flux (node centered)] |
y aqueous relative permeability |
z aqueous vol[umetric flux] | z solute flux | znc aqueous vol[umetric flux (node centered)] }

* Reference Node Variable Only
P Proprietary

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WA** }
{ air aqueous conc[entration] | air aqueous mass frac[tion] | air gas conc[entration] |
air gas mass frac[tion] | air gas mole frac[tion] | air mass source int[egral] |
air mass source rate | air partial pressure |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture saturation | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix saturation | aqueous moisture cont[ent] | aqueous pressure |
aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
aqueous well depth*P | axial aquous flux | axial gas flux |
diffusive porosity | dissolved air saturation |
effective trapped gas saturation | gas courant [number] | gas density |
gas fracture saturation | gas gauge pressure | gas hydraulic head |
gas matrix saturation | gas pressure | gas relative perm[eability] |
gas saturation | integrated air mass* | integrated aqueous air [mass]* |
integrated aqueous water [mass]* | { integrated gas air [mass]* |
integrated gas water [mass]* | { integrated trapped gas air [mass]* |
integrated water mass* | phase condition | rock/soil type |
solute aqueous conc[entration] | solute aqueous conc[entration] |
solute aqueous mole frac[tion] |solute gas conc[entration] | solute gas mole frac[tion] |
solute gas conc[entration] | solute source int[egral] | solute volumetric conc[entration] |
temperature | total air mass | total water mass | trapped gas saturation |
vertical aqueous flux | vertical gas flux |
water aqueous conc[entration] | water aqueous mass frac[tion] | water gas conc[entration] |
water gas mass frac[tion] | water gas mole frac[tion] | water source int[egral] |
water mass source int[egral] | water mass source rate \ water vapor partial pres[sure] |
well flow integral*P | well flow rate*P |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x solute flux |
xnc aqueous vol[umetric flux (node centered)] | xnc gas vol[umetric flux (node centered)] |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y solute flux |
ync aqueous vol[umetric flux (node centered)] | ync gas vol[umetric flux (node centered)] |
z aqueous vol[umetric flux] | z gas vol[umetric flux] | z solute flux |
{ znc aqueous vol[umetric flux (node centered)] | znc gas vol[umetric flux (node centered)] }

* Reference Node Variable Only
P Proprietary

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WAE** } { **STOMP-WAE-Sc** }
{ air aqueous conc[entration] | air aqueous mass frac[tion] | air gas conc[entration] |
air gas mass frac[tion] | air gas mole frac[tion] | air mass source int[egral] |
air mass source rate | air partial pressure |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture saturation | aqueous gauge pressure | aqueous hydraulic head |
aqueous matric potential | aqueous moisture cont[ent] | aqueous pressure |
aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
axial aquous flux | axial gas flux |diffusive porosity |
dissolved air saturation | effective trapped air | energy source int[egral] |
energy source rate | gas courant [number] | gas density | gas fracture saturation |
gas gauge pressure | gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation | integrated air mass* |
integrated aqueous air [mass]* | integrated aqueous water [mass]* |
integrated gas air [mass]* | integrated gas water [mass]* | integrated trapped gas air* |
integrated water mass* | phase condition | plant temperature |
potential evaporation rate | potential transpiration rate | rainfall interception mass |
rock/soil type | solute aqueous conc[entration] | solute aqueous mole fra[ction] |
solute gas conc[entration] | solute gas mole fra[ction] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total air mass frac[tion] |
total water mass frac[tion] | vertical aqueous flux | vertical gas flux |
water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate | water vapor partial pressure |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x heat flux | x solute flux |
x thermal cond[uctivity] |xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc heat flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y heat flux |y solute flux |
y thermal cond[uctivity] | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync heat flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] | z heat flux |z solute flux |
z thermal cond[uctivity] | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc heat flux (node centered) }

* Reference Node Variable Only
P Proprietary

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WAE-B** }
{ actual evaporation rate* | actual transpiration rate* |
air aqueous conc[entration] | air aqueous mass frac[tion] | air gas conc[entration] |
air gas mass frac[tion] | air gas mole frac[tion] | air mass source int[egral] |
air mass source rate | air partial pressure |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture saturation | aqueous gauge pressure | aqueous hydraulic head |
aqueous matric potential | aqueous moisture cont[ent] | aqueous pressure |
aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
atmospheric pressure* | atmospheric relative humidity* | atmospheric solar radiation* |
atmospheric temperature* | atmospheric wind speed* |
axial aquous flux* | axial gas flux* | bare-soil aero[dynamic resistance]* |
bare-surface aero[dynamic resistance]* |diffusive porosity |
dissolved air saturation | effective trapped air | energy source int[egral] |
energy source rate | gas courant [number] | gas density | gas fracture saturation |
gas gauge pressure | gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation | integrated air mass* |
integrated aqueous air [mass]* | integrated aqueous water [mass]* |
integrated gas air [mass]* | integrated gas water [mass]* | integrated trapped gas air* |
integrated water mass* | phase condition | plant temperature |
potential evaporation rate* | potential transpiration rate* | rainfall interception mass* |
rock/soil type | solute aqueous conc[entration] | solute aqueous mole fra[ction] |
solute gas conc[entration] | solute gas mole fra[ction] | solute source int[egral] |
solute volumetric conc[entration] | stomatal resistance* | surface aqueous pressure* |
surface aqueous saturation* | surface gas pressure* | surface ground heat flux* |
surface latent heat flux* | surface mass precipitation* |
surface net long-wave rad[iation]* | surface net-short wave rad[iation]* |
surface net total rad[iation]* | surface sensible heat flux* | surface volumetric precipitation* |
surface temperature* | surface vapor pressure* | surface water mass bal[ance]* |
temperature | total air mass frac[tion] | total water mass frac[tion] |
vertical aqueous flux | vertical gas flux |
water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate | water vapor partial pressure |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x heat flux | x solute flux |
x thermal cond[uctivity] |xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc heat flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y heat flux |y solute flux |
y thermal cond[uctivity] | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync heat flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] | z heat flux |z solute flux |
z thermal cond[uctivity] | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc heat flux (node centered) }

* Reference Node Variable Only
P Proprietary

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WO** } { **STOMP-WOD** }
{ apparent aqueous saturation | apparent total saturation |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous gauge pressure | aqueous hydraulic head | aqueous moisture cont[ent] |
aqueous pressure | aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
aqueous trapped gas saturation | diffusive porosity | entrapped napl-water interfacial area |
effective trapped gas saturation | effective trapped napl saturation | free napl saturation |
free napl-water interfacial area | gas-free napl interfacial area |
gas-water interfacial area | integrated aqueous water [mass]* |
integrated aqueous oil [mass]* | integrated oil mass* | integrated water mass* |
mobile napl saturation | napl courant [number] | napl density |
napl gauge pressure | napl hydraulic head | napl moisture cont[ent] |
napl pressure | napl relative perm[eability] | napl saturation |
napl trapped gas saturation | oil aqueous conc[entration] | oil aqueous mass frac[tion] |
oil gas conc[entration] | oil gas mass frac[tion] | oil gas mole frac[tion] |
oil mass source int[egral] | oil mass source rate |
phase condition | residual napl saturation | rock/soil type |
solute aqueous conc[entration] | solute aqueous mole frac[tion] | solute napl conc[entration] |
solute napl mole frac[tion] | solute source int[egral] | solute volumetric conc[entration] |
temperature | total moisture cont[ent] | total oil mass |
total saturation | total water mass | total well flow rate* | trapped gas saturation |
trapped napl saturation | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate | well depth* |
x aqueous vol[umetric flux] | x napl vol[umetric flux] | x solute flux |
xnc aqueous vol[umetric flux (node centered)] | xnc napl vol[umetric flux (node centered)] |
y aqueous vol[umetric flux] | y napl vol[umetric flux] | y solute flux |
ync aqueous vol[umetric flux (node centered)] | ync napl vol[umetric flux (node centered)] |
z aqueous vol[umetric flux] | z napl vol[umetric flux] | z solute flux |
znc aqueous vol[umetric flux (node centered)] | znc napl vol[umetric flux (node centered)] }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WOM** }
{ apparent aqueous saturation | apparent total saturation |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous gauge pressure | aqueous hydraulic head | aqueous moisture cont[ent] |
aqueous pressure | aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
aqueous trapped gas saturation | diffusive porosity | effective trapped gas saturation |
effective trapped napl saturation | free napl saturation | integrated aqueous water [mass]* |
integrated aqueous oil [mass]* | integrated oil mass* | integrated water mass* |
mobile napl saturation | napl courant [number] | napl density |
napl gauge pressure | napl hydraulic head | napl moisture cont[ent] |
napl pressure | napl relative perm[eability] | napl saturation | napl viscosity |
napl trapped gas saturation | oil aqueous conc[entration] | oil aqueous mass frac[tion] |
oil gas conc[entration] | oil gas mass frac[tion] | oil gas mole frac[tion] |
oil mass source int[egral] | oil mass source rate | oil napl mass fraction |
phase condition | residual napl saturation | rock/soil type |
solute aqueous conc[entration] | solute aqueous mole frac[tion] | solute napl conc[entration] |
solute napl mole frac[tion] | solute source int[egral] | solute volumetric conc[entration] |
temperature | total moisture cont[ent] | total oil mass |
total saturation | total water mass | total well flow rate* | trapped gas saturation |
trapped napl saturation | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate | well depth* |
x aqueous vol[umetric flux] | x napl vol[umetric flux] | x solute flux |
xnc aqueous vol[umetric flux (node centered)] | xnc napl vol[umetric flux (node centered)] |
y aqueous vol[umetric flux] | y napl vol[umetric flux] | y solute flux |
ync aqueous vol[umetric flux (node centered)] | ync napl vol[umetric flux (node centered)] |
z aqueous vol[umetric flux] | z napl vol[umetric flux] | z solute flux |
znc aqueous vol[umetric flux (node centered)] | znc napl vol[umetric flux (node centered)] }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc }**

{ air aqueous conc[entration] | air aqueous mass frac[tion] | air gas conc[entration] |
air gas mass frac[tion] | air gas mole frac[tion] | air mass source int[egral] |
air mass source rate | air partial pressure | apparent total saturation |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous gauge pressure | aqueous hydraulic head | aqueous moisture cont[ent] |
aqueous pressure | aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
diffusive porosity | effective trapped gas saturation | effective trapped napl saturation |
free napl saturation | gas courant [number] | gas density | gas gauge pressure |
gas hydraulic head | gas pressure | gas relative perm[eability] |
integrated air mass* | integrated aqueous air [mass]* | integrated aqueous oil [mass]* |
integrated aqueous water [mass]* | integrated gas air [mass]* |
integrated gas oil [mass]* | integrated gas water [mass]* | integrated oil mass* |
integrated water mass* | gas saturation | mobile napl saturation |
napl courant [number] | napl density |
napl gauge pressure | napl hydraulic head | napl moisture cont[ent] |
napl pressure | napl relative perm[eability] | napl saturation | napl vapor partial pressure |
oil aqueous conc[entration] | oil aqueous mass frac[tion] | oil gas conc[entration] |
oil gas mass frac[tion] | oil gas mole frac[tion] | oil mass source int[egral] |
oil mass source rate | phase condition | residual napl saturation |
rock/soil type | solute aqueous conc[entration] |
solute aqueous mole frac[tion] | solute gas conc[entration] | solute gas mole frac[tion] |
solute napl conc[entration] | solute napl mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total air mass |
total moisture cont[ent] | total oil mass | total saturation |
total water mass | trapped gas saturation | trapped napl saturation |
water aqueous conc[entration] | water aqueous mass frac[tion] | water gas conc[entration] |
water gas mass frac[tion] |water gas mole frac[tion] |water mass source int[egral] |
water mass source rate | water vapor partial pressure |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x napl vol[umetric flux] |
x solute flux | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc napl vol[umetric flux (node centered)] |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y napl vol[umetric flux] |
y solute flux | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync napl vol[umetric flux (node centered)] |
z aqueous vol[umetric flux] | z gas vol[umetric flux] | z napl vol[umetric flux] |
z solute flux | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc napl vol[umetric flux (node centered)] }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = **{ STOMP-WOAE }**

{ air aqueous conc[entration] | aer aqueous mass frac[tion] | air conc[entration] |
air gas mass frac[tion] |air gas mole frac[tion] |
air mass source int[egral] | air mass source rate | apparent total saturation |
apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous gauge pressure | aqueous hydraulic head | aqueous moisture cont[ent] |
aqueous pressure | aqueous relative perm[eability] | aqueous saturation | aqueous viscosity |
diffusive porosity | energy source int[egral] | energy source rate |
effective trapped gas saturation | effective trapped napl saturation |
gas courant [number] | gas density | gas gauge pressure |
gas hydraulic head | gas pressure | gas relative perm[eability] |
gas saturation | integrated air mass* | integrated aqueous air [mass]* |
integrated aqueous oil [mass]* | integrated aqueous water [mass]* | integrated gas air [mass]* |
integrated gas oil [mass]* | integrated gas water [mass]* | integrated oil mass* |
integrated water mass* | napl courant [number] | napl density |
napl gauge pressure | napl hydraulic head | napl moisture cont[ent] |
napl pressure | napl relative perm[eability] | napl saturation |
oil aqueous conc[entration] |oil aqueous mass frac[tion] | oil gas conc[entration] |
oil gas mass frac[tion] | oil gas mole frac[tion] | oil mass source int[egral] |
oil mass source rate | phase condition | rock/soil type | solute aqueous conc[entration] |
solute aqueous mol | solute gas conc[entration] | solute gas mole frac[tion] |
solute napl conc[entration] | solute napl mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total air mass |
total moisture cont[ent] | total oil mass | total saturation |
total water mass | trapped gas saturation | trapped napl saturation |
water aqueous conc[entration] | water aqueous mass frac[tion] | water gas conc[entration] |
water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x heat flux |
x napl vol[umetric flux] | x solute flux | x thermal cond[uctivity] |
xnc aqueous vol[umetric flux (node centered)] | xnc gas vol[umetric flux (node centered)] |
xnc heat flux [node centered] | xnc napl vol[umetric flux (node centered)] |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y heat flux |
y napl vol[umetric flux] | y solute flux | y thermal cond[uctivity] |
ync aqueous vol[umetric flux (node centered)] | ync gas vol[umetric flux (node centered)] |
ync heat flux [node centered] } | ync napl vol[umetric flux (node centered)] |
z aqueous vol[umetric flux] | z gas vol[umetric flux] |z heat flux |
z napl vol[umetric flux] | z solute flux | z thermal cond[uctivity] |
znc aqueous vol[umetric flux (node centered)] | znc gas vol[umetric flux (node centered)] |
znc ync heat flux [node centered] | znc napl vol[umetric flux (node centered)] }

Endif:

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WS** } { **STOMP-WS-Sc** }

{ apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture saturation | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix saturation | aqueous moisture cont[ent] | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity |
aqueous trapped gas saturation | aqueous well depth*P | diffusive porosity |
effective trapped gas saturation | gas-aqueous scaling | gas gauge pressure | gas pressure
gas saturation | integrated water [mass]* | integrated aqueous water [mass]* |
phase condition | rock/soil type | osmotic eff | osmotic pressure |
salt aqueous conc[entration] | salt mass source int[egral] | salt mass source rate |
salt volumetric conc[entration] | solute aqueous conc[entration] |
solute gas conc[entration] | solute source | solute volumetric conc[entration] |
temperature | total water [mass] | total salt [mass] | trapped gas saturation |
water mass source int[egral] | water mass source rate |
x aqueous vol[umetric flux] | x salt flux |x solute flux |
xnc aqueous vol[umetric flux (node centered)] | xnc salt flux }
y aqueous vol[umetric flux] | y salt flux | y solute flux |
ync aqueous vol[umetric flux (node centered) | ync salt flux |
z aqueous vol[umetric flux] | z salt flux | z solute flux |
znc aqueous vol[umetric flux (node centered)] | znc salt flux }

* Reference Node Variable Only
P Proprietary Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WAS** }

{ air aqueous conc[entration] | air aqueous mass frac[tion] | air gas conc[entration] |
air gas mass frac[tion] | air gas mole frac[tion] | air mass source int[egral] |
air mass source rate | apparent aqueous saturation |
aqueous courant [number] | aqueous density |
aqueous fracture | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity |
diffusive porosity | effective trapped gas saturation | gas courant [number] |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation |
integrated air mass* | integrated aqueous air [mass]*
integrated aqueous water [mass]* | integrated gas air [mass]* |
integrated gas water [mass]* { integrated water mass* |
osmotic eff | osmotic pressure | phase condition | rock/soil type | salt aqueous conc[entration] |
salt conc[entration] | salt mass source int[egral] | salt mass source rate |
solute aqueous conc[entration] | solute aqueous mole frac[tion] |
solute gas conc[entration] | solute gas mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total air mass | total salt mas |
total water mass | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x salt flux | x solute flux |
xnc aqueous vol[umetric flux (node centered)] | xnc gas vol[umetric flux (node centered)] |
xnc heat flux (node centered) | xnc salt flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y salt flux | y solute flux |
ync aqueous vol[umetric flux (node centered)] | ync gas vol[umetric flux (node centered)] |
ync heat flux (node centered) |ync salt flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] |z salt flux | z solute flux |
znc aqueous vol[umetric flux (node centered)] | znc gas vol[umetric flux (node centered)] |
znc heat flux (node centered) | znc salt flux (node centered) }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WASE** }

{ air aqueous conc[entration] | air aqueous mass frac[tion] | air gas conc[entration] |
air gas mass frac[tion] | air gas mole frac[tion] | air mass source int[egral] |
air mass source rate | apparent aqueous saturation |
aqueous courant [number] | aqueous density |
aqueous fracture | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity |
diffusive porosity | effective trapped gas saturation | energy source int[egral]
energy source rate| gas courant [number] |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation |
integrated air mass* | integrated aqueous air [mass]*
integrated aqueous water [mass]* | integrated gas air [mass]* |
integrated gas water [mass]* { integrated water mass* |
osmotic eff | osmotic pressure | phase condition | rock/soil type | salt aqueous conc[entration] |
salt conc[entration] | salt mass source int[egral] | salt mass source rate |
solute aqueous conc[entration] | solute aqueous mole frac[tion] |
solute gas conc[entration] | solute gas mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total air mass | total salt mas |
total water mass | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral] | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x heat flux | x salt flux | x solute flux |
x thermal cond[uctivity] | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] |xnc heat flux (node centered) |
xnc salt flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] | y heat flux | y salt flux | y solute flux |
y thermal cond[uctivity] | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] |ync heat flux (node centered) |
ync salt flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] | z heat flux |z salt flux | z solute flux |
z thermal cond[uctivity] |znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc heat flux (node centered) |
znc salt flux (node centered) }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-Sc** }

{ apparent aqueous saturation | aqueous courant [number] |
aqueous density | aqueous fracture |
aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity |
CO_2 aqueous conc[entration] | CO_2 aqueous mass frac[tion] |
CO_2 aqueous mole frac[tion] | CO_2 gas conc[entration] |
CO_2 gas mass frac[tion] | CO_2 gas mole frac[tion] | CO_2 mass source int[egral] |
CO_2 mass source rate | apparent aqueous saturation | aqueous courant |
diffusive porosity | effective trapped gas saturation |
gas aqueous scaling | gas courant [number] |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation |
integrated CO_2 mass* | integrated aqueous CO_2 [mass]* |
integrated aqueous water [mass]* | integrated gas CO_2 [mass]* |
integrated gas water [mass]* | integrated water mass* |
osmotic eff | osmotic pressure | phase condition |
rock/soil type | salt aqueous conc[entration] | salt aqueous mass fraction |
salt aqueous mole frac[tion] |salt conc[entration] | salt mass source int[egral] |
salt mass source rate | salt saturation |
solute aqueous conc[entration] | solute aqueous mole frac[tion] |
solute gas conc[entration] | solute gas mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total CO_2 mass | total salt mass |
total water mass | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] |water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] |
x salt flux | x solute flux | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc salt flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] |
y salt flux | y solute flux | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync salt flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] |
z salt flux |z solute flux | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc salt flux (node centered) }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = **{ STOMP-WCS-R }** { **STOMP-WCS-R-Sc** }

{ apparent aqueous saturation | aqueous courant [number] |
aqueous density | aqueous fracture |
aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity |
CO_2 aqueous conc[entration] | CO_2 aqueous mass frac[tion] |
CO_2 aqueous mole frac[tion] | CO_2 gas conc[entration] |
CO_2 gas mass frac[tion] | CO_2 gas mole frac[tion] | CO_2 mass source int[egral] |
CO_2 mass source rate | apparent aqueous saturation | aqueous courant |
diffusive porosity | effective trapped gas saturation |
gas aqueous scaling | gas courant [number] |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation |
integrated CO_2 mass* | integrated aqueous CO_2 [mass]* |
integrated aqueous water [mass]* | integrated gas CO_2 [mass]* |
integrated gas water [mass]* | integrated water mass* |
osmotic eff | osmotic pressure | phase condition |
rock/soil type | salt aqueous conc[entration] | salt aqueous mass fraction |
salt aqueous mole frac[tion] |salt conc[entration] | salt mass source int[egral] |
salt mass source rate | salt saturation |
solute aqueous conc[entration] | solute aqueous mole frac[tion] |
solute gas conc[entration] | solute gas mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total CO_2 mass | total salt mass |
species aqueous conc[entration] | species volumetric conc[entration] |
species source | species integrated mass |
total water mass | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] |water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] |
x salt flux | x solute flux | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc salt flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] |
y salt flux | y solute flux | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync salt flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] |
z salt flux |z solute flux | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc salt flux (node centered) }

* Reference Node Variable Only

Output Control Card (cont'd)

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WCSE** }
{ apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity |
CO_2 aqueous conc[entration] | CO_2 aqueous mass frac[tion] |
CO_2 aqueous mole frac[tion] | CO_2 gas conc[entration] |
CO_2 gas mass frac[tion] | CO_2 gas mole frac[tion] | CO_2 mass source int[egral] |
CO_2 mass source rate | diffusive porosity | effective trapped gas saturation |
energy source int[egral] | energy source rate |
gas courant [number] | gas aqueous scaling |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation |
integrated CO_2 mass* | integrated aqueous CO_2 [mass]* |
integrated aqueous water [mass]* | integrated gas CO_2 [mass]* |
integrated gas water [mass]* | integrated water mass* |
osmotic eff | osmotic pressure | phase condition | saturated CO_2 aqueous mass fraction |
rock/soil type | salt aqueous conc[entration] | salt aqueous mass fraction |
salt aqueous mole frac[tion] |salt conc[entration] | salt mass source int[egral] |
salt mass source rate | salt saturation |
solute aqueous conc[entration] | solute aqueous mole frac[tion] |
solute gas conc[entration] | solute gas mole frac[tion] | solute source int[egral] |
solute volumetric conc[entration] | temperature | total CO_2 mass | total salt mass |
total water mass | water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas mass frac[tion] | water gas mole frac[tion] |
water mass source int[egral | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] |
x heat flux | x salt flux | x solute flux |
x thermal cond[uctivity] | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc heat flux (node centered) |
xnc salt flux (node centered) |
y aqueous vol[umetric flux] | y gas vol[umetric flux] |
y heat flux | y salt flux | y solute flux |
y thermal cond[uctivity] | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync heat flux (node centered) |
ync salt flux (node centered) |
z aqueous vol[umetric flux] | z gas vol[umetric flux] |
z heat flux | z salt flux |z solute flux |
z thermal cond[uctivity] |znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc heat flux (node centered) |
znc salt flux (node centered) }

* Reference Node Variable Only

Reference Node Variable and Plot File Variable Options

Elseif: Operational Mode Option = { **STOMP-WCMSE** }
{ apparent aqueous saturation | aqueous courant [number] | aqueous density |
aqueous fracture | aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt mass frac[tion] |
aqueous saturation | aqueous viscosity | CH_4 aqueous conc[entration] |
CH_4 aqueous mass frac[tion] | CH_4 aqueous mole frac[tion] | CH_4 gas conc[entration] |
CH_4 gas mass frac[tion] | CH_4 gas mole frac[tion] | CH_4 hydrate mass frac[tion] |
CH_4 hydrate mole frac[tion] | CH_4 mass source int[egral] |
CO_2 aqueous conc[entration] | CO_2 aqueous mass frac[tion] | CO_2 aqueous mole frac[tion] |
CO_2 gas conc[entration] | CO_2 gas mass frac[tion] | | CO_2 gas mole frac[tion] |
CO_2 hydrate mass frac[tion] | CO_2 hydrate mole frac[tion] CO_2 mass source int[egral] |
CO_2 mass source rate | diffusive porosity | effective trapped gas saturation |
energy source int[egral] | energy source rate | gas courant [number] | gas aqueous scaling |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure | gas relative perm[eability] |
gas saturation | hydrate density | hydrate pressure | hydrate saturation |
integrated CO_2 mass* | integrated aqueous CO_2 [mass]* | integrated CH_4 mass* | integrated
aqueous CH_4 [mass]* | integrated aqueous water [mass]* | integrated CO_2 gas [mass]* |
integrated CH_4 gas [mass]* |
integrated gas water [mass]* | integrated water mass* | liquid CO_2 saturation |
osmotic eff | osmotic pressure | phase condition | saturated CO_2 aqueous mass fraction |
rock/soil type | salt aqueous conc[entration] | salt aqueous mass fraction |
salt aqueous mole frac[tion] | salt conc[entration] | salt mass source int[egral] |
salt mass source rate | salt saturation | solute aqueous conc[entration] |
solute aqueous mole frac[tion] | solute gas conc[entration] | solute gas mole frac[tion] |
solute source int[egral] | solute volumetric conc[entration] | temperature | total CO_2 mass |
total salt mass | total water mass | total CH_4 mass | water aqueous conc[entration] |
water aqueous mass frac[tion] | water gas conc[entration] |
water gas mass frac[tion] | water gas mole frac[tion] | water hydrate mass frac[tion] |
water hydrate mole frac[tion] | water mass source int[egral | water mass source rate |
x aqueous vol[umetric flux] | x gas vol[umetric flux] | x heat flux | x salt flux | x solute flux |
x thermal cond[uctivity] | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc heat flux (node centered) |
xnc salt flux (node centered) | y aqueous vol[umetric flux] | y gas vol[umetric flux] |
y heat flux | y salt flux | y solute flux |
y thermal cond[uctivity] | ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync heat flux (node centered) |
ync salt flux (node centered) | z aqueous vol[umetric flux] | z gas vol[umetric flux] |
z heat flux | z salt flux | z solute flux |
z thermal cond[uctivity] | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc heat flux (node centered) |
znc salt flux (node centered) }

Endif:

* Reference Node Variable Only

Output Control Card (cont'd)

Default Units for the Reference Node and Plot File Variables

Air Conc. (kg/m^3)
Aerodynamic Resistance (s/m)
CO_2 Mass Source Integral (kg)
Aqueous Saturation (null)
Density (kg/m^3)
Evaporation (kg/s)
Gas Saturation (null)
Gauge Pressure (Pa)
Heat Flux (W/ m^2)
Mass (kg)
Mass Precipitation (kg/day)
Mole Frac[tion] (null)
NAPL Saturation (null)
Phase Condition (null)
Solar Radiation (W/m^2)
Saturation (null)
Salt Aqueous Conc. (kg/m^3)
Solute Conc. ($1/m^3$)
Solute Flux ($1/m^2 s$)
Species Mass Source Integral (mol)
Thermal Conductivity (W/m K)
Oil Conc. (kg/m^3)
Volumetric Precipitation (m^3/s)
Water Mass Source Integral (kg)
Air Mass Source Integral (kg)
CO_2 Conc. (kg/m^3)
Aqueous Relative Permeability (null)
Aqueous Volumetric Flux (m/s)
Energy Source Integral (J)
Gas Relative Permeability (null)
Gas Volumetric Flux (m/s)
Head (m),
Humidity (null)
Mass Fraction (null)
Moisture Content (null)
NAPL Relative Permeability (null)
NAPL Volumetric Flux (m/s)
Pressure (Pa)
Relative Permeability (null)
Salt Conc. (kg/m^3)
Salt Flux ($kg/ m^3 s$)
Solute Mole frac[tion] (null)
Species Conc. (mol/m^3)
Temperature (C)
Transpiration (kg/s)
Oil Mass Source Integral (kg)
Water Conc. (kg/m^3)
Well Depth (m)

B.23.1 Output Control Card Examples

Extracted from STOMP-W input file:

```
~Output Options Card
5,
57,1,82,
57,1,60,
57,1,40,
57,1,18,
57,1,9,
2,1,yr,m,6,6,6,
27,
aqueous saturation,,
aqueous pressure,Pa,
aqueous relative perm,,
aqueous moisture content,,
x aqueous vol,mm/yr,
z aqueous vol,mm/yr,
solute integrated mass,U:0.01,,
solute aqueous conc,   U:0.01, 1/L,
solute volumetric conc,U:0.01, 1/L,
solute integrated mass,U:0.03,,
solute aqueous conc,   U:0.03, 1/L,
solute volumetric conc,U:0.03, 1/L,
solute integrated mass,U:0.10,,
solute aqueous conc,   U:0.10, 1/L,
solute volumetric conc,U:0.10, 1/L,
solute integrated mass,U:0.30,,
solute aqueous conc,   U:0.30, 1/L,
solute volumetric conc,U:0.30, 1/L,
solute integrated mass,U:0.60,,
solute aqueous conc,   U:0.60, 1/L,
solute volumetric conc,U:0.60, 1/L,
solute integrated mass,U:1.00,,
solute aqueous conc,   U:1.00, 1/L,
solute volumetric conc,U:1.00, 1/L,
solute integrated mass,Tc,,
solute aqueous conc,   Tc,    1/L,
solute volumetric conc,Tc,    1/L,
5,
2000,yr,
2000.0383299,yr,
2001,yr,
2010,yr,
2050,yr,
21,
aqueous saturation,,
aqueous pressure,Pa,
aqueous relative perm,,
aqueous moisture content,,
rock/Soil type,,
```

Output Card Examples (cont'd)

xnc aqueous vol,mm/yr,
znc aqueous vol,mm/yr,
solute aqueous conc, U:0.01, 1/L,
solute volumetric conc,U:0.01, 1/L,
solute aqueous conc, U:0.03, 1/L,
solute volumetric conc,U:0.03, 1/L,
solute aqueous conc, U:0.10, 1/L,
solute volumetric conc,U:0.10, 1/L,
solute aqueous conc, U:0.30, 1/L,
solute volumetric conc,U:0.30, 1/L,
solute aqueous conc, U:0.60, 1/L,
solute volumetric conc,U:0.60, 1/L,
solute aqueous conc, U:1.00, 1/L,
solute volumetric conc,U:1.00, 1/L,
solute aqueous conc, Tc, 1/L,
solute volumetric conc,Tc, 1/L,

Extracted from STOMP-W-R input file:
~Output Options Card
10,
1,1,1,
2,1,1,
3,1,1,
4,1,1,
5,1,1,
6,1,1,
7,1,1,
8,1,1,
9,1,1,
10,1,1,
1,1,hr,m,6,6,6,
11,
Species Aqueous Concentration,H+,mol/L,
Species Aqueous Concentration,Biomass,mol/L,
Species Aqueous Concentration,HNta--,mol/L,
Species Aqueous Concentration,Co(ads),mol/L,
Species Aqueous Concentration,CoNta(ads),mol/L,
Species Aqueous Concentration,Nta---,mol/L,
Species Aqueous Concentration,CO2,mol/L,
Species Aqueous Concentration,Co++,mol/L,
Species Aqueous Concentration,CoNta-,mol/L,
Species Aqueous Concentration,CoNTA2----,mol/L,
Species Aqueous Concentration,C5H7O2N,mol/L,
0,
11,
Species Aqueous Concentration,H+,mol/L,
Species Aqueous Concentration,Biomass,mol/L,
Species Aqueous Concentration,HNta--,mol/L,
Species Aqueous Concentration,Co(ads),mol/L,
Species Aqueous Concentration,CoNta(ads),mol/L,

Output Card Examples (cont'd)

Species Aqueous Concentration,Nta---,mol/L,
Species Aqueous Concentration,CO2,mol/L,
Species Aqueous Concentration,Co++,mol/L,
Species Aqueous Concentration,CoNta-,mol/L,
Species Aqueous Concentration,CoNTA2----,mol/L,
Species Aqueous Concentration,C5H7O2N,mol/L,

Extracted from STOMP-WAE-B input file:

~Output Options Card
1,
1,1,10,
#1,1,9,
#1,1,8,
#1,1,7,
#1,1,6,
#1,1,1,
1,1,day,m,6,6,6,
12,
Aqueous Saturation,,
Temperature,C,
Surface Temperature,C,
Surface Vapor Pressure,Pa,
Atmospheric Temperature,C,
Atmospheric Solar Radiation,W/m^2,
Atmospheric Wind Speed,m/s,
Atmospheric Relative Humidity,,
Potential Evaporation,gm/s,
Actual Evaporation,gm/s,
Potential Transpiration,gm/s,
Actual Transpiration,gm/s,
0,
6,
Aqueous Saturation,,
Aqueous Pressure,Pa,
Gas Pressure,Pa,
Temperature,C,
Surface Temperature,C,
Surface Vapor Pressure,Pa,

Extracted from STOMP-WAE-B input file:

#---
~Output Options Card
#---
3,
1,1,113,# 0.1 cm
1,1,50, # 1.01 m deep
1,1,1,
1,1,day,m,3,6,6,
15,
Atmospheric Solar Radiation,W/m^2,

Output Card Examples (cont'd)

Surface Net Long-Wave Radiation,W/m^2,
Surface Net Short-Wave Radiation,W/m^2,
Surface Sensible Heat Flux,W/m^2,
Surface Latent Heat Flux,W/m^2,
Surface Ground Heat Flux,W/m^2,
Atmospheric Temperature,C,
Surface Temperature,C,
Surface Volumetric Precipitation,cm^3/day,
Potential Evaporation,kg/day,
Actual Evaporation,kg/day,
Potential Transpiration,kg/day,
Actual Transpiration,kg/day,
Aqueous matric potential,cm,
Aqueous moisture content,,
10,
0.0,day,
72.0,day,
164.0,day,
254.0,day,
345.0,day,
437.0,day,
529.0,day,
619.0,day,
710.0,day,
803.0,day,
5,
Temperature,C,
Aqueous pressure,pa,
Aqueous matric potential,cm,
Aqueous moisture content,,
Aqueous relative permeability,,

Extracted from STOMP-WOM input file:
~Output Options Card
3,
1,1,1,
10,1,1,
20,1,1,
1,1,d,cm,6,6,6,
8,
aqueous saturation,,
napl saturation,,
oil aqueous concentration,lard-oil mixture,gm/liter,
oil aqueous concentration,lnapl,gm/liter,
oil aqueous concentration,brooklawn napl,gm/liter,
oil NAPL mass fraction,lard-oil mixture,,
oil NAPL mass fraction,lnapl,,
oil NAPL mass fraction,brooklawn napl,,
0,
8,

aqueous saturation,,
napl saturation,,
oil aqueous concentration,lard-oil mixture,gm/liter,
oil aqueous concentration,lnapl,gm/liter,
oil aqueous concentration,brooklawn napl,gm/liter,
oil NAPL mass fraction,lard-oil mixture,,
oil NAPL mass fraction,lnapl,,
oil NAPL mass fraction,brooklawn napl,,

Extracted from input file:
~Output Options Card
3,
1,1,42,
1,1,62,
1,1,92,
10,10,hr,cm,6,6,6,
4,
aqueous saturation,,
apparent Water saturation,,
NAPL saturation,,
oil aqueous conc,mg/L,
3,
0.025,hr,
0.075,hr,
0.125,hr,
6,
aqueous saturation,,
apparent Water saturation,,
NAPL saturation,,
oil aqueous conc,mg/L,
trapped oil saturation,,
total trapping number,,

Extracted from STOMP-WS input file:
~Output Options Card
6,
30,1,120,
30,1,119,
30,1,118,
60,1,120,
60,1,119,
60,1,118,
1,1,min,m,6,6,6,
6,
Aqueous Saturation,,
Aqueous Moisture Content,,
Aqueous Pressure,,
Z Aqueous Vol,cm/hr,
Salt Aqueous Concentration,kg/m^3,
Aqueous Density,kg/m^3,

Output Card Examples (cont'd)

5,
0.,min,
5.,min,
10.,min,
20.,min,
30.,min,
0,

Extracted from STOMP-WCSE input file:
~Output Options Card
4,
3,1,3,
3,1,4,
4,1,3,
98,1,3,
1,1,hr,cm,6,6,6,
5,
Gas Saturation,,
Temperature,C,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Bar,
7,
0.5,min,
1.0,min,
2.0,min,
5.0,min,
10.0,min,
20.0,min,
50.0,min,
6,
Gas Saturation,,
Temperature,C,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Bar,
Aqueous Density,kg/m^3,

Extracted from STOMP-WCMSE input file:
~Output Options Card
8,
1,1,1,
2,1,1,
3,1,1,
4,1,1,
5,1,1,
10,1,1,
20,1,1,

30,1,1,

Output Card Examples (cont'd)

```
1,1,hr,m,6,6,6,
12,
Temperature,C,
Aqueous Pressure,MPa,
Gas Pressure,MPa,
Gas Saturation,,
Aqueous Saturation,,
Hydrate Saturation,,
Aqueous Relative Permeability,,
Integrated CH4 Mass,kg,
Integrated CH4 Aqueous Mass,kg,
Integrated CH4 Hydrate Mass,kg,
CH4 Aqueous Mass Fraction,,
CH4 Gas Mass Fraction,,
7,
1,hr,
3,hr,
6,hr,
12,hr,
1,day,
2,day,
3,day,
9,
Temperature,C,
Aqueous Pressure,MPa,
Gas Pressure,MPa,
Gas Saturation,,
Aqueous Saturation,,
Hydrate Saturation,,
Aqueous Relative Permeability,,
CH4 Aqueous Mass Fraction,,
CH4 Gas Mass Fraction,,
```

B.24 Plant Card

Card Title[a] { ~Plant [Card] }
Format: *Char[a]*

If: Operational Mode Option Card = { **STOMP-WAE-B** }

Number of Plant Species[a], [Plant Temperature Option[b]], [Rainfall Interception Option[c]],
Format: *Integer[a]*, *Char[a]*, *Char[b]*,

For: Number of Plant Species
Plant Name[a], [Plant Stress Option[b]], [Stomatal Resistance[c]], { Stress }
Format: *Char[a]*, *Char[b]*, *Char[b]*,
Max. Root Depth[a], Units[b] (m),
Null Root Depth[a], Units[b] (m),
Root Depth Fit Parameter[a],
Format: *Real[a]*, *Char[b]*, *Real[c]*, Char[d], *Real[e]*,
If: Plant Albedo = { Temporal[a] }
Plant Albedo Initial Stage Start[b], Plant Albedo Crop Development Start[c],
Plant Albedo Mid-Season Start[d], Plant Albedo Late-Season Start[e],
Plant Albedo Late-Season Stop[f], Plant Canopy Height[g], Unitsh (m)[h],
Maximum Condensate Depth[i], Units[j] (m),
Format: *Char[a]*, *Real[b]*, *Real[c]*, *Real[d]*, *Real[e]*, *Real[f]*, *Real[g]*, *Char[h]*, *Real[i]*, *Char[j]*,
Else:
Plant Albedo[a],Plant Canopy Height[b], Units (m)[b],
Format: *Real[a]*,*Real[b]*, *Char[c]*,
Endif:
If: Plant Stress Option = { Stress }
Water Stress Point 1[a], Units[b] (m), Water Stress Point 2[c], Units[d] (m),
Water Stress Point 3[e], Units[f] (m), Water Stress Point 4[g], Units[h] (m),
Format: *Real[a]*, *Char[b]*, *Real[c]*, *Char[d]*, *Real[e]*, *Char[f]*, *Real[g]*, *Char[h]*,
Else:
Root Uptake Reduced 50%[a], Units[b] (m),
Format: *Real[a]*, *Char[b]*,
Endif:
Crop Coefficient Start[a], Crop Coefficient Day of Year (Start)[b], Units (s)[c],
Crop Coefficient (Mature Stage 1)[d], Crop Coefficient Day of Year (Mature Stage 1)[e],
Units (s)[f], Crop Coefficient (Mature Stage 2)[g],
Crop Coefficient Day of Year (Mature Stage 2)[h], Units (s)[i],
Crop Coefficient (Die-off)[j], Crop Coefficient Day of Year (Die-off)[k], Units (s)[l],
Format: *Real[a]*, *Real[b]*, *Char[c]*, *Real[d]*, *Real[e]*, *Char[f]*, *Real[g]*, *Real[h]*, *Char[i]*, *Real[j]*, *Real[k]*, *Char[l]*,
Endfor: Number of Plant Species

Endif:

Endcard: Plant Card

B.24.1 Plant Card Examples

Extracted from STOMP-WAE-B input file:

```
#----------------------------------
~Plant Card
#----------------------------------
2,Single Plant Temperature Rainfall Interception,
Cheatgrass,Vrugt Root Stress Hicks,
0.50,m,0.10,m,4.875,
Temporal Albedo,0.05,0.06,0.15,0.19,0.05,0.30,m,1.984e-3,m,
0.1,m,1.0,m,10.0,m,150.,m,
0.0,0.0,day,0.0,55,day,1.0,112,day,1.0,148,day,0.0,159,day,
50,s/m,20,W/m^2,5,C,45,C,25,C,
Sandberg Bluegrass,Vrugt Root Stress Hicks,
0.35,m,0.20,m,2.62,
Temporal Albedo,0.05,0.06,0.15,0.19,0.05,0.10,m,1.984e-3,m,
0.1,m,1.0,m,10.0,m,150.,m,
0.0,0.0,day,0.0,55,day,1.0,104,day,1.0,147,day,0.0,160,day,
50,s/m,20,W/m^2,5,C,45,C,25,C,
```

B.25 Rock/Soil Zonation Card

Card Title[a] { ~Rock/Soil [Zonation Card] }
Format: *Char[a]*

Rock/Soil Zonation Input Option[a],
{ [Formatted | Unformatted] Zonation File | [IJK | JKI | KIJ] Indexing | Integer }

If: Rock/Soil Zonation Input Option = { [Formatted | Unformatted] Zonation File }
Rock/Soil Zonation File Name[b],
Format: *Char[a], Char[b],*
For: Number of Rock/Soil Zonation Domains (defined in external file)
Rock/Soil or Scaling Group Name[a],
Format: *Char[a],*
Endfor: Number of Rock/Soil Zonation Domains

Elseif: Rock/Soil Zonation Input Option = { Indexing }

Note: Each node is assigned a different Rock/Soil zonation index according to the indexing scheme chosen (i.e., IJK, JKI, or KIJ). This option is useful for stochastic realizations.

Elseif: Rock/Soil Zonation Input Option = { Integer }

For: Number of Rock/Soil Zonation Domains
Rock/Soil or Scaling Group Name[a],
I-Start Index[b], I-End Index[c],
J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g],
If: Operation Mode Option = { **STOMP-W** } { **STOMP-W-R**} { **STOMP-W-Sc** } and
Execution Mode Option = { **Normal w/ Scaling** | **Restart w/ Scaling** }
Scaling Group[h],
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Char[h],*
Else:
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g],*
Elseif:
Endfor: Number of Rock/Soil Zonation Domains

Endif:

Endcard: Rock/Soil Zonation Card

B.25.1 Rock/Soil Zonation Card Examples

Extracted from a STOMP-W input file:
~Rock/Soil Zonation Card
1,
20/30 Ottawa Sand,1,1,1,1,1,92,

Extracted from a STOMP-W input file:
~Rock/Soil Zonation Card
IJK Indexing,

Extracted from a STOMP-W input file:
~Rock/Soil Zonation Card
7,
SP1,1,50,1,1,86,113,
SP2,1,50,1,1,63,85,
SM-ML1,1,50,1,1,69,69,
SM-SP1,1,50,1,1,46,62,
SP3,1,50,1,1,16,45,
SW1,1,50,1,1,1,15,
US,2,2,1,1,71,100,

Extracted from a STOMP-W input file:
~Rock/Soil Zonation Card
formatted file, zonation_a,
backfill,
hanford sand,
plio-pleistocene,
upper ringold,
middle ringold,

Extracted from a STOMP-W input file:
~Rock/Soil Zonation Card
5,
L5,1,1,1,1,1,40,R1,
L4,1,1,1,1,41,80,R1,
L3,1,1,1,1,81,100,R1,
L2,1,1,1,1,101,105,R1,
L1,1,1,1,1,106,130,R1,

Extracted from a STOMP-W input file:
#4
~Rock/Soil Zonation Card
zonation file formatted,c_geology.dat,
Backfill,
H2 Sand,
H3 Gravelly Sand,
H1 Gravelly Sand,
PPlgR,
Aquifer,

B.26 Salt Transport Card

Card Title[a] { ~Salt Transport [Card] }
Format: *Char[a]*

If: Operational Mode Option Card = { **STOMP-WS** } { **STOMP-WS-Sc** } { **STOMP-WAS** } { **STOMP-WASE** }

Salt Name[a],
{ Sodium Chloride [Pitzer] | Sodium Nitrate [Pitzer] | Sodium Thiosulfate [Pitzer] | Sodium Chloride Henry [Pitzer] | Sodium Chloride Elder [Pitzer] }
Format: Char[a],
If: Salt Name = { Sodium Chloride Pitzer } { Sodium Nitrate Pitzer } { Sodium Thiosulfate Pitzer } { Sodium Chloride Henry Pitzer } { Sodium Chloride Elder Pitzer }
Number of Anions[a], Number of Cations[b],
Number of Positive Charges[c], Number of Negative Charges[d],
Pitzer Pairwise Ion-Interaction Parameter B0[e],
Pitzer Pairwise Ion-Interaction Parameter B1[f],
Pitzer Pairwise Ion-Interaction Parameter B2[h],
Pitzer Triplet Ion-Interaction Parameter[i],
Surface to Bulk Molality Ratio[j],
Format: *Integer[a], Integer[b], Integer[c], Integer[d], Real[e], Real[f], Real[g], Real[h], Real[i],*
Endif:

Effective Diffusion Option[a],

{ Constant } **Note:** $D_{\ell e}^{s} = D_{\ell}^{s}$

{ Conventional } **Note:** $D_{\ell e}^{s} = \tau_{\ell} s_{\ell} n_{D} D_{\ell}^{s}$

{ Empirical [Kemper and van Schaik] } **Note:** $D_{\ell e}^{s} = D_{\ell}^{s} a \exp(b n_{D} s_{\ell})$

If: Effective Diffusion Option = { Constant }
Aqueous-Phase Diffusion Coefficient @ 20 C[b], Units[c] (m^2/s),
Format: *Char[a], Real[b], Char[c],*
Elseif: Effective Diffusion Option = { Conventional }
Aqueous-Phase Diffusion Coefficient @ 20 C[b], Units[c] (m^2/s),
Format: *Char[a], Real[b], Char[c],*
Elseif: Effective Diffusion Option = { Empirical [Kemper and van Schaik] }
Format: *Char[a],*
Endif:

If: Effective Diffusion Option = { Constant } { Conventional }
Rock/Soil Name[a],

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input for the next input line can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil Types
Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Endfor: Number of Rock/Soil Types
Endif:

Elseif: Effective Diffusion Option = { Empirical }
Rock/Soil Name[a],

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input for the next input line can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Salt Transport Card (cont'd)

Elseif:
For: Number of Rock/Soil Types
Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m),
Aqueous Molecular Diffusion Coefficient[f], Units[g] (m^2/s),
Constant a[h],
Constant b[i],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Real[i],*
Endfor: Number of Rock/Soil Types
Endif:
Endif:
Endif:

If: Operational Mode Option Card = { **STOMP-WCS** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** } { **STOMP-WCMSE** }

Rock/Soil Name[a],

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:
For: Number of Rock/Soil Types
Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Endfor: Number of Rock/Soil Types
Endif:
Endif:

Endcard: Salt Transport Card

B.26.1 Salt Transport Card Examples

Extracted from a STOMP-WS input file:
~Salt Transport Card
Sodium Nitrate,
Constant Diffusion,3.565e-6,m^2/s,
30/40 sand,0.0,m,0.0,m,
Extracted from a STOMP-WCS input file
~Salt Transport Card
Eau Claire Carbonate,20.0,ft,5.0,ft,
Eau Claire Shale,20.0,ft,5.0,ft,
Lower Eau Claire,20.0,ft,5.0,ft,
Upper Mt. Simon,20.0,ft,5.0,ft,
Middle Mt. Simon,20.0,ft,5.0,ft,
Lower Mt. Simon,20.0,ft,5.0,ft,
Extracted from a STOMP-WCMSE input file
~Salt Transport Card
Medium,0.0,ft,0.0,ft,

B.27 Saturation Function Card

Card Title[a] { ~Saturation Function [Card] }
Format: *Char[a]*

If: Operational Mode Option = { **STOMP-WO** } { **STOMP-WOD** } { **STOMP-WOM** } { **STOMP-WOA** } { **STOMP-WOA-Sc** }{ **STOMP-WOAE** }

Fluid Pair Interfacial Tension

Note: Enter minimum of two fluid pair interfacial tension values. For nonspreading NAPLs, (spreading coefficient < 0) enter all 3 values when Operational Mode Option = { **STOMP-WO** }

{ Gas-Aqueous Interfacial Tension[a], Units[b] (N/m),
NAPL-Aqueous Interfacial Tension[c], Units[d] (N/m),
Null[e], Null[f], }

{ Null[a], Null[b],
NAPL-Aqueous Interfacial Tension[c], Units[d] (N/m),
Gas-NAPL Interfacial Tension[e], Units[f] (N/m), }

{ Gas-Aqueous Interfacial Tension[a], Units[b] (N/m),
Null[c], Null[d],
Gas-NAPL Interfacial Tension[e], Units[f] (N/m), }

Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*

Endif:

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing

Note: The parameter input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section B.26.1

Saturation Function Card (cont'd)

Elseif:

For: Number of Rock/Soil or Scaling-Group Types

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** } { **STOMP-WS** } { **STOMP-WS-Sc** } { **STOMP-WAS** } { **STOMP-WASE** }

Rock/Soil or Scaling Group Name[a],
Saturation Function[b],
{ [Entrapment | Extended | Fractured | Triple] van Genuchten |
[Entrapment | Extended | Fractured | Triple] Brooks and Corey |
Haverkamp | Tabular [Linear | Spline [Log]] [Water Content | Saturation] }

If: Saturation Function Option = { [Extended] van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g],*

Elseif: Saturation Function Option = { Entrapment van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g], Effective Gas Residual Saturation[h],
Critical Trapping Number[i],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h], Real[i],*

Elseif: Saturation Function Option = { Fractured van Genuchten }
Matrix α Parameter[c], Units[d] (1/m), Matrix n Parameter[e],
Matrix Minimum Saturation[f],
Fracture α Parameter[g], Units[h] (1/m), Fracture n Parameter[i],
Fracture Minimum Saturation[j],
Matrix m Parameter[k], Fracture m Parameter[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],*

Elseif: Saturation Function Option = { Triple van Genuchten }
Main Drainage α Parameter[c], Units[d] (1/m),
Main Drainage n Parameter[e], Main Drainage Minimum Saturation[f],
Main Drainage m Parameter[g],
Boundary Wetting α Parameter[h], Units[i] (1/m),
Main Wetting α Parameter[j], Units[k] (1/m),
Main Wetting n Parameter[l], Main Wetting Minimum Saturation[m],
Main Wetting m Parameter[n],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h], Real[i], Real[j], Char[k], Real[l], Real[m], Real[n],*

Elseif: Saturation Function Option = { [Extended] Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f],*

Elseif: Saturation Function Option = { Entrapment Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Effective Gas Residual Saturation[g],
Critical Trapping Number[h],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h],*

Elseif: Saturation Function Option = { Fractured Brooks and Corey }
Matrix Entry Head[c], Units[d] (m), Matrix λ Parameter[e],
Matrix Minimum Saturation[f],
Fracture Entry Head[g], Units[h] (m), Fracture λ Parameter[i],
Fracture Minimum Saturation[j],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h],Real[i], Real[j],*

Elseif: Saturation Function Option = { Triple Brooks and Corey }
Main Entry Head[c], Units[d] (m),
Main Drainage λ Parameter[e], Main Drainage Minimum Saturation[f],
Boundary Wetting Entry Head[g], Units[h] (m),
Main Wetting Entry Head[i], Units[j] (m),
Main Wetting λ Parameter[k], Main Wetting Minimum Saturation[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Char[j], Real[k], Real[l],*

Elseif: Saturation Function Option = { Haverkamp }
Entry Head Parameter[c], Units[d] (m), α Parameter[e], β Parameter[f], Minimum Saturation[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*

Elseif: Saturation Function Option = { Tabular [Saturation]
[Linear | Spline [Log]] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Aqueous Saturation[c],
Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Elseif: Saturation Function Option = { Tabular Water Content
[Linear | Spline [Log]] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Water Content[c],
Format: *Real[a], Char[b], Real[c],*

Saturation Function Card (cont'd)

Endfor: Number of Table Entries

Endif:

Elseif: Operational Mode Option: { **STOMP-WA** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCS** }

Rock/Soil Name[a],
Saturation Function[b],
{ [Entrapment | Extended | Fractured] van Genuchten |
[Entrapment | Extended | Fractured] Brooks and Corey |
Haverkamp | Tabular [Linear | Spline [Log]] [Water Content | Saturation] }

If: Saturation Function Option = { [Extended] van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g],*

Elseif: Saturation Function Option = { Entrapment van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g], Effective Gas Residual Saturation[h],
Critical Trapping Number[i],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h], Real[i],*

Elseif: Saturation Function Option = { Fractured van Genuchten }
Matrix α Parameter[c], Units[d] (1/m), Matrix n Parameter[e],
Matrix Minimum Saturation[f],
Fracture α Parameter[g], Units[h] (1/m), Fracture n Parameter[i],
Fracture Minimum Saturation[j],
Matrix m Parameter[k], Fracture m Parameter[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],*

Elseif: Saturation Function Option = { [Extended] Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f],*

Elseif: Saturation Function Option = { Entrapment Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Effective Gas Residual Saturation[g],
Critical Trapping Number[h],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h],*

Elseif: Saturation Function Option = { Fractured Brooks and Corey }
Matrix Entry Head[c], Units[d] (m), Matrix λ Parameter[e],
Matrix Minimum Saturation[f],

Fracture Entry Head[g], Units[h] (m), Fracture λ Parameter[i],
Fracture Minimum Saturation[j],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j],*

Elseif: Saturation Function Option = { Haverkamp }
Entry Head Parameter[c], Units[d] (m), α Parameter[e], β Parameter[f],
Minimum Saturation[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*

Elseif: Saturation Function Option = { Tabular [Saturation]
[Linear | Spline [Log]] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Aqueous Saturation[c],
Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Elseif: Saturation Function Option = { Tabular Water Content
[Linear | Spline [Log]] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Water Content[c],
Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Endif:

Saturation Function Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WAE** } { **STOMP-WAE-B** } { **STOMP-WAE-Sc** }

Rock/Soil Name[a],
Saturation Function[b],
{ [Extended | Fractured] van Genuchten |
[Extended | Fractured] Brooks and Corey |
Tabular [Linear | Spline] [Water Content | Saturation] }

If: Saturation Function Option = { [Extended] van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g],*

Elseif: Saturation Function Option = { Fractured van Genuchten }
Matrix α Parameter[c], Units[d] (1/m), Matrix n Parameter[e],
Matrix Minimum Saturation[f],
Fracture α Parameter[g], Units[h] (1/m), Fracture n Parameter[i],
Fracture Minimum Saturation[j],
Matrix m Parameter[k], Fracture m Parameter[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],*

Elseif: Saturation Function Option = { [Extended] Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f],*

Elseif: Saturation Function Option = { Fractured Brooks and Corey }
Matrix Entry Head[c], Units[d] (m), Matrix λ Parameter[e],
Matrix Minimum Saturation[f],
Fracture Entry Head[g], Units[h] (m), Fracture λ Parameter[i],
Fracture Minimum Saturation[j],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j],*

Elseif: Saturation Function Option = { Tabular [Saturation] [Linear | Spline] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Aqueous Saturation[c],
Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Elseif: Saturation Function Option = { Tabular Water Content [Linear | Spline] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Water Content[c],

Saturation Function Card (cont'd)

Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Endif:

Saturation Function Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WO** } { **STOMP-WO-We** } **{ STOMP-WOA } { STOMP-WOA-Sc }**

Rock/Soil Name[a],
Saturation Function[b]
{ [Residual [Geel [Linear [Mod]] | Geel [Land [Mod]] |
Lenhard [Mod] | [Constant]]] [Entrapment] van Genuchten |
[Residual [Geel [Linear [Mod]] | Geel [Land [Mod]] |
Lenhard [Mod] | [Constant]]] [Entrapment] Brooks and Corey }

If: Saturation Function Option = { van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i],*

Elseif: Saturation Function Option = { Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Format: *Char[a], Char[b], Real[e], Char[d], Real[e], Real[f], Real[g], Char[h],*

Elseif: Saturation Function Option = { Entrapment van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i], Effective NAPL Maximum Entrapped Saturation[j],
Critical NAPL Trapping Number[k],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k],*

Elseif: Saturation Function Option = { Entrapment Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Effective NAPL Maximum Entrapped Saturation[i],
Critical NAPL Trapping Number[j],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j],*

Elseif: Saturation Function Option = { Residual [Geel [Linear [Mod]] |
Geel [Land [Mod]] | Lenhard [Mod] | [Constant]] van Genuchten }
α Parameter[c], Units[d] (1/m),
n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i], Effective Maximum Residual NAPL Saturation[j],

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j],*

Elseif: Saturation Function Option = { Residual [Geel [Linear [Mod]] | Geel [Land [Mod]] | Lenhard [Mod] | [Constant]] Brooks and Corey }
Entry Head[c], Units[d] (m),
λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Effective Maximum Residual NAPL Saturation[i],
Format: *Char[a], Char[b], Real[e], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i],*

Elseif: Saturation Function Option = { Residual and Entrapment van Genuchten }
α Parameter[c], Units[d] (1/m),
n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i], Effective Maximum Entrapped NAPL Saturation[j],
Effective Maximum Residual NAPL Saturation[k],
Critical NAPL Trapping Number[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],*

Elseif: Saturation Function Option = { Residual and Entrapment Brooks and Corey }
Entry Head[c], Units[d] (m),
λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Effective Maximum Entrapped NAPL Saturation[i],
Effective Maximum Residual NAPL Saturation[j],
Critical NAPL Trapping Number[k],
Format: *Char[a], Char[b], Real[e], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k],*

Endif:

Saturation Function Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WOAE }**

Rock/Soil Name[a],
Saturation Function[b]
{ Nonhysteretic van Genuchten | Nonhysteretic Brooks and Corey }

If: Saturation Function Option = { Nonhysteretic van Genuchten }
α Parameter[c], Units[d] (1/m),
n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i],*

Elseif: Saturation Function Option = { Nonhysteretic Brooks and Corey }
Entry Head[c], Units[d] (m),
λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Format: *Char[a], Char[b], Real[e], Char[d], Real[e], Real[f], Real[g], Char[h],*

Endif:

Saturation Function Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WOD** } { **STOMP-WOM** }

Rock/Soil Name[a],
Saturation Function[b]
{ [Entrapment] van Genuchten | [Entrapment] Brooks and Corey }

If: Saturation Function Option = { van Genuchten }
α Parameter[c], Units[d] (1/m),
n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i],*

Elseif: Saturation Function Option = { Brooks and Corey }
Entry Head[c], Units[d] (m),
λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Format: *Char[a], Char[b], Real[e], Char[d], Real[e], Real[f], Real[g], Char[h],*

Elseif: Saturation Function Option = { Entrapment van Genuchten }
α Parameter[c], Units[d] (1/m),
n Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
m Parameter[i], Effective NAPL Residual Saturation[j],
Effective Gas Residual Saturation[k], Critical NAPL Trapping Number[l],
Critical Gas Trapping Number[m],
Format: *Char[a], Char[b], Real[c], Char[d],*
Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l], Real[m],

Elseif: Saturation Function Option = { Entrapment Brooks and Corey }
Entry Head[c], Units[d] (m),
λ Parameter[e],
Minimum Saturation[f],
Reference Fluid-Pair Interfacial Tension[g], Units[h] (N/m),
Effective NAPL Residual Saturation[i],
Effective Gas Residual Saturation[j], Critical NAPL Trapping Number[k],
Critical Gas Trapping Number[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],*

Endif:

Saturation Function Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCSE** } { **STOMP-WCMSE** }

Rock/Soil Name[a],
Saturation Function[b],
{ [Entrapment | Extended | Fractured] van Genuchten |
[Entrapment | Extended | Fractured] Brooks and Corey |
Haverkamp | Tabular [Linear | Spline] [Water Content | Saturation] }

If: Saturation Function Option = { [Extended] van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g],*

Elseif: Saturation Function Option = { Entrapment van Genuchten }
α Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g], Effective Gas Residual Saturation[h],
Critical Trapping Number[i],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h], Real[i],*

Elseif: Saturation Function Option = { Fractured van Genuchten }
Matrix α Parameter[c], Units[d] (1/m), Matrix n Parameter[e],
Matrix Minimum Saturation[f],
Fracture α Parameter[g], Units[h] (1/m), Fracture n Parameter[i],
Fracture Minimum Saturation[j],
Matrix m Parameter[k], Fracture m Parameter[l],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],*

Elseif: Saturation Function Option = { [Extended] Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f],*

Elseif: Saturation Function Option = { Entrapment Brooks and Corey }
Entry Head[c], Units[d] (m), λ Parameter[e], Minimum Saturation[f],
Effective Gas Residual Saturation[g],
Critical Trapping Number[h],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h],*

Elseif: Saturation Function Option = { Fractured Brooks and Corey }
Matrix Entry Head[c], Units[d] (m), Matrix λ Parameter[e],
Matrix Minimum Saturation[f],
Fracture Entry Head[g], Units[h] (m), Fracture λ Parameter[i],
Fracture Minimum Saturation[j],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Char[h], Real[i], Real[j],*

Saturation Function Card (cont'd)

Elseif: Saturation Function Option = { Haverkamp }
Entry Head Parameter[c], Units[d] (m), α Parameter[e], β Parameter[f], Minimum Saturation[g],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*

Elseif: Saturation Function Option = { Tabular Saturation [Linear | Spline] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Aqueous Saturation[c],
Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Elseif: Saturation Function Option = { Tabular Water Content [Linear | Spline] }
Number of Table Entries[a],
Format: Char[a], Char[b], Integer[a]
For: Number of Table Entries
Air-Water Capillary Head[a], Units[b] (m), Water Content[c],
Format: *Real[a], Char[b], Real[c],*
Endfor: Number of Table Entries

Endif:

Endif:

Endfor: Number of Rock/Soil Types

Endif:

Endcard: Saturation Function Card

B.27.1 Saturation Function Card Examples

Extracted from a STOMP-W input file:
~Saturation Function Card
IJK Indexing, Tabular Log-linear,12,file:tabh.dat,cm,file:tabs.dat,saturation,

Extracted from a STOMP-W input file:
~Saturation Function Card
Sand,Van Genuchten,1.563,1/m,5.4,0.15,,

Extracted from a STOMP-W input file:
#R1 is a scaling group
~Saturation Function Card
R1,van Genuchten,0.036,1/cm,1.756,0.092417,,

Extracted from a STOMP-W input file:
~Saturation Function Card
backfill,van Genuchten,9.885,1/m,2.928,0.0774,,
hanford sand,van Genuchten,0.0092,1/cm,1.8848,0.08366,,
plio-pleistocene,van Genuchten,0.0067,1/cm,1.8378,0.25953,,
upper ringold,van Genuchten,0.0029,1/cm,1.6285,0.21295,,
middle ringold,van Genuchten,0.0062,1/cm,1.6452,0.0686,,

Extracted from a STOMP-W input file:
~Saturation Function Card
Backfill,Entrapment van Genuchten,0.0210,1/cm,1.3740,0.07246,,0.2,2e-5,
H2 Sand,Entrapment van Genuchten,0.0117,1/cm,1.6162,0.11600,,0.2,2e-5,
H1 Gravelly Sand,Entrapment van Genuchten,0.0141,1/cm,1.3730,0.01505,,0.3,2e-5,
H3 Gravelly Sand,Entrapment van Genuchten,0.0197,1/cm,1.4194,0.05618,,0.2,2e-5,

Extracted from a STOMP-WO input file:
~Saturation Function Card
72.0,dynes/cm,38.0,dynes/cm,,,
Sand,Brooks and Corey,10.12,cm,2.67,0.08,72.0,dynes/cm,

Extracted from a STOMP-WO input file:
~Saturation Function Card
72.0,dynes/cm,44.4,dynes/cm,,,
Sand,Entrapment Van Genuchten,2.5,1/m,2.0,.1,72.0,dynes/cm,,0.25,2e-5,

Extracted from a STOMP-WO input file:
~Saturation Function Card
72.0,dynes/cm,53.1,dynes/cm,24.7,dynes/cm,
Sand,Brooks and Corey w/ Residual,10.12,cm,2.67,0.08,72.0,dynes/cm,0.235,

Extracted from a STOMP-WO input file:
~Saturation Function Card
72.0,dynes/cm,53.1,dynes/cm,24.7,dynes/cm,
Sand,Residual Lenhard Modified Brooks and Corey,10.12,cm,2.67,0.08,72.0,dynes/cm,0.235,

Saturation Function Card Examples (Cont'd)

Extracted from a STOMP-WO input file:

```
~Saturation Function Card
72.0,dynes/cm,53/1,dynes/cm,24.7,dynes/cm,
Sand,Brooks and Corey w/ Entrapment w/ Residual,10.12,cm,2.67,0.08,72.0,dynes/cm,0.10,0.20,2e-5,
```

Extracted from a STOMP-WO input file:

```
# ------------------------------------------------------------------
~Saturation Function Card
# ------------------------------------------------------------------
72.2,dynes/cm,,,25.1,dynes/cm,
basalt,Brooks and Corey,71.31,cm,0.5193,0.1299,72.2,dynes/cm,,,,,
ring_a,Brooks and Corey,71.31,cm,0.5193,0.1299,72.2,dynes/cm,,,,,
low_rmud,Brooks and Corey,71.31,cm,0.5193,0.1299,72.2,dynes/cm,,,,,
ring_e,Brooks and Corey,71.31,cm,0.5193,0.1299,72.2,dynes/cm,,,,,
up_ring,Brooks and Corey,71.31,cm,0.5193,0.1299,72.2,dynes/cm,,,,,
pplc,Brooks and Corey,36.31,cm,0.614,0.2451,72.2,dynes/cm,,,,,
pplz,Brooks and Corey,120.04,cm,0.7884,0.0967,72.2,dynes/cm,,,,,
low_sand,Brooks and Corey,4.65,cm,0.7773,0.0747,72.2,dynes/cm,,,,,
low_grvl,Brooks and Corey,23.04,cm,0.7465,0.1471,72.2,dynes/cm,,,,,
h2,Brooks and Corey,14.13,cm,0.949,0.0846,72.2,dynes/cm,,,,,
h1,Brooks and Corey,7.67,cm,0.5445,0.1386,72.2,dynes/cm,,,,,
h1a,Brooks and Corey,58.13,cm,0.7052,0.1740,72.2,dynes/cm,,,,,
backfill,Brooks and Corey,22.02,cm,0.3646,0.1145,72.2,dynes/cm,,,,,
```

B.28 Scaling Factor Card

Card Title[a] { ~Scaling [Factor Card] }
Format: *Char[a]*

If: Operational Mode Option Card = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }

If: Execution Mode Option = { Normal w/ Scaling } { Restart w/ Scaling }
Saturated Hydraulic Conductivity Scaling Function[a], { Log | Linear }
Diffusive Porosity Scaling Function[b], { Log | Linear }
Van Genuchten "alpha" or Brooks-Corey "psi" Scaling Function[c], { Log | Linear }
Van Genuchten "n" or Brooks-Corey "lambda" Scaling Function[d], { Log | Linear }
Residual Saturation Scaling Function[e], { Log | Linear }
Format: *Char[a], Char[b], Char[c], Char[d], Char[e],*
For: Number of Rock/Soil Types
Rock/Soil Type[a],
If: Rock/Soil Type contains {{ Fractured }} {{ DP }} {{ Dual }}
Saturated Hydraulic Conductivity Scaling Factor[b],
Saturated Hydraulic Conductivity Scaling Factor[c],
Diffusive Porosity Scaling Factor[d],
Diffusive Porosity Scaling Factor[e],
Van Genuchten "alpha" or Brooks-Corey "psi" Scaling Factor[f],
Van Genuchten "alpha" or Brooks-Corey "psi" Scaling Factor[g],
Van Genuchten "n" or Brooks-Corey "lambda" Scaling Factor[h],
Van Genuchten "n" or Brooks-Corey "lambda" Scaling Factor[i],
Residual Saturation Scaling Function Factor[j],
Residual Saturation Scaling Function Factor[k],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e], Real[f], Real[g], Real[h], Real[i], Real[j], Real[k],*
If: Permeability Function Option = { Anisotropy Mualem }
Horizontal Pore-Scale Scaling Factor[l],
Horizontal Pore-Scale Scaling Factor[m],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e], Real[f], Real[g], Real[h], Real[i], Real[j], Real[k], Real[l], Real[m],*
Endif:
Else:
Saturated Hydraulic Conductivity Scaling Factor[b],
Diffusive Porosity Scaling Factor[c],
Van Genuchten "alpha" or Brooks-Corey "psi" Scaling Factor[d],
Van Genuchten "n" or Brooks-Corey "lambda" Scaling Factor[e],
Residual Saturation Scaling Function Factor[f],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e], Real[f],*
If: Permeability Function Option = { Anisotropy Mualem }
Horizontal Pore-Scale Scaling Factor[g],
Horizontal Pore-Scale Scaling Factor[h],
Format: *Char[a], Real[b], Real[c], Real[d], Real[e], Real[f], Real[g], Real[h],*
Endif:

Scaling Factor Card (cont'd)

Endfor:
Endfor:
Endif:

Endif:

Endcard: Scaling Factor Card

B.28.1 Scaling Factor Card Examples

Extracted from a STOMP-W input file:

```
~Scaling Card
Linear,Linear,Linear,Linear,Linear,
L1,1.0000,1.000,1.000,1.000,1.000,
L2,0.0429,0.754,0.933,1.003,0.681,
L3,0.0256,0.580,1.512,1.100,0.681,
L4,0.0105,0.553,1.561,1.667,1.301,
```

B.29 Simulation Title Card

Card Title[a] { ~Simulation [Title Card] }
Format: *Char[a]*

Version Number[a],
Format: *Integer[a],*

Simulation Title[a],
Format: *Char[a],*

User Name[a],
Format: *Char[a],*

Company Name[a],
Format: *Char[a],*

Input Creation Date[a],
Format: *Char[a],*

Input Creation Time[a],
Format: *Char[a],*

Number of Simulation Note Lines[a],
Format: *Integer[a],*

For: Number of Simulation Note Lines

Simulation Notes[a]
Format: *Char[a] (maximum of 132 characters per line)*

Endfor: Number of Simulation Note Lines

Endcard: Simulation Title Card

B.29.1 Simulation Title Card Examples

Extracted from a STOMP-W input file:
~Simulation Title Card
1,
Scaling Method,
Zhang,
PNNL,
22 June 2001,
13:40,
2,
Rockhold et al.(1988)
Drainage in the Grass Site

Extracted from a STOMP-W-R input file:
~Simulation Title Card
3.2,
PHREEQC Example 15,
M.D. White,
Pacific Northwest Laboratory,
13 September 2005,
16:06 AM PDT,
2,
PHREEQC Example 15
1D transport: kinetic biodegradation, cell growth, and sorption

Extracted from a STOMP-W input file:
~Simulation Title Card
1,
C-Tank Farm Simulation: Initial Condition,
ZF Zhang, VL Freedman, and MD White,
PNNL,
January 29 2003,
9:30,
3,
CASE 1, 1945-2050: This input file is used to simulate transport (U and TC)
using an initial concentration condition profile.
Cross section for Tank Row C-103, C-106, C-109, and C-112.

Extracted from a STOMP-W input file:
~Simulation Title Card
1,
Field Test #12,
MD White,
PNNL,
Monday April 5 1996,
12:34,
2,
Simulation of field test #12 at Edwards AFB.
Starting time 1/3/96 14:19, Ending time 1/8/96 9:19.

Simulation Title Card Examples (cont'd)

Extracted from a STOMP-WAE input file:
~Simulation Title Card
1,
Evaporation/Condensation Heat Pipe,
MD White,
PNNL,
June 18 1994,
10:04 AM PDT,
3,
This application problem follows the heat-pipe problem solved
semi-analytically by Udell and Fitch. The Soil moisture retention
function has been changed to a modified van Genuchten function.

Extracted from a STOMP-WO input file:
~Simulation Title Card
1,
DW2 experiment for Partioning Tracer Study,
Mart Oostrom,
PNNL,
Jan-00,
15:05,
3,
Partitioning tracer test for detection, estimation, and remediation performance
assessment of subsurface nonaqueous phase liquids. Jin et al.
Water Resources Research, Vol. 31, No. 5, Pages 1201-1211, May 1995.

Extracted from a STOMP-WO input file:
~Simulation Title Card
1,
Brooklawn Site: Disposal Scenario,
Mark White,
Tel: (509) 372-6070, E-mail: mark.white@pnl.gov,
Pacific Northwest National Laboratory,
03-Mar-00,
4,
Simulation to establish an initial aqueous flow field in a cross-section
extending southward along the E. 6150 rdinate line from N. 5000 to N. 4000,
extending vertically from ground surface to a depth of -170 ft-msl, using
the steady-state calibration results.

Extracted from a STOMP-WCS input file:
~Simulation Title Card
1,
BP_UTCOMP Comparison (Base Case),
M.D. White,
Pacific Northwest Laboratory,
30 October 2002,
04:06 PM PST,
23,
CC**********

CC

Simulation Title Card Examples (cont'd)

```
CC  BRIEF DESCRIPTION OF DATA SET: UTCOMP ( VERSION UTCOMP-3.5 )    *
CC                                                                  *
CC*********************************************************************
CC Radial geometry                                                  *
CC Mt. Simon Sst. Run BPIN1 2D Full scale,radial model              *
CC (phase 1=inactivewater,phase 2 = water/oil,phase 3=co2/gas       *
CC LENGTH(FT):               INJECTION FLUID: CO2                   *
CC HEIGHT(FT):               INJECTION RATE: cont. pre.             *
CC WIDTH(FT): variable       W/O REL. PERM:                         *
CC POROSITY: variable        G/O REL. PERM: lindeburg               *
CC ABS. PERM(MD): variable    3-PHASE REL. PERM: water endpt.=1.0   *
CC TEMP(F):    .0            WETTIBILITY:                           *
CC PRESSURE(PSI): .    psi/ft  W/O CAP. PRESSURE:                  *
CC SOR:                      G/O CAP. PRESSURE:                     *
CC SWC:                      DISPLACEMENT TYPE: HORIZONTAL          *
CC stop injection after 20 years run for 40 yrs,h2o k endpoint=1    *
CC ****NON-IDEAL MIXING, NO GRAVITY, WITH Pc, WITH X-FLOW ****      *
CC FILE NAME:                                                       *
CC CREATED BY Neeraj Gupta                                          *
CC MODIFIED BY Neeraj Gupta,                                        *
CC*********************************************************************
```

Extracted from a STOMP-WCMSE input file:

```
~Simulation Title Card
3.2,
STOMP Short Course Problem 13,
M.D. White,
Pacific Northwest Laboratory,
31 January 2006,
08:52 AM PST,
1,
Methane Hydrate Dissociation via Thermal Stimulation in a 1-Dimensional Open Domain
```

B.30 Solid Species Card

Card Title[a] { ~Solid Species [Card] }
Format: *Char*[a],

Number of Solid Species[a],
Format: *Integer*[a],

For: Number of Solid Species
Solid Species Name[a],
Species Mass Density[b], Units[c] (kg/m^3)
Species Molecular Weight[d], Units[e] (kg/m^3)
Format: *Char*[a], *Real*[b], *Char*[c], *Real*[d], *Char*[e],
Endfor: Number of Solid Species

Endcard: Solid Species

B.30.1 Solid Species Card

Extracted from a STOMP-W-R input file:
~Solid Species Card
4,
Co(ads),,,58.9332,kg/kmol,
CoNta(ads),,,247.0638,kg/kmol,
C5H7O2N,,,113.09,kg/kmol,
Biomass,,,1.0,kg/kmol,

Extracted from a STOMP-W-R input file:
~Solid Species Card
3,
Co(ads),,,58.9332,kg/kmol,
CoNTA(ads),,,247.0638,kg/kmol,
C5H7O2N,,,113.1158,kg/kmol,

Extracted from a STOMP-WCS-R input file:
~Solid Species Card
1,
CaCO3(s),2.709893,gm/cm^3,100.0872,kg/kmol,

B.31 Solute/Fluid Interactions Card

Card Title[a] { ~Solute/Fluid [Interactions Card] }
Format: *Char*[a]

Number of Solutes
Format: *Integer*[a],

For: Number of Solutes

If: Operational Mode Option Card = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
Solute Name[a],
Effective Diffusion Option[b],

{ Constant | **Note:** $D_{\ell}^{C}{}_{e} = \overline{D}_{\ell}^{C}{}_{e}$

Conventional | **Note:** $D_{\ell}^{C}{}_{e} = \tau_{\ell} s_{\ell} n_{D} D_{\ell}^{C}$

Empirical [Kemper and van Schaik] } **Note:** $D_{\ell}^{C}{}_{e} = D_{\ell}^{C} a \exp\left(b n_{D} s_{\ell}\right)$
Model parameters for the Empirical option are entered in the Solute/Porous Medium Interaction Card

If: Effective Diffusion Option = { Empirical }
Solute-Aqueous Partition Option[c],

{ Continuous | **Note:** $C_{\ell} = \dfrac{C}{n_{D} s_{\ell} + \left(1 - n_{T}\right) \rho_{s} K_{s\ell}}$

Noncontinuous } **Note:** $C_{\ell} = \dfrac{C}{n_{D} s_{\ell} + s_{\ell} \left(1 - n_{T}\right) \rho_{s} K_{s\ell}}$

IfDef: Radioactive
Half-Life[d], Units[e] (s),
Format: *Char*[a], *Char*[b], *Char*[c], *Real*[d], *Char*[e],
ElseifDef: Reactive
Number of Reactions[d],
For: Number of Reactions:
First-Order Decay Constant[e], Units[f] (s),
Endfor: Number of Reactions
Format: *Char*[a], *Char*[b], *Char*[c], *Integer*[d],<*Real*[e], *Char*[f],>
EndifDef:

Else:
Aqueous-Phase Molecular Diffusion Coefficient @ 20 C[c], Units[d] (m^2/s),
Solute Partition Option[e],

{ Continuous | **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + (1 - n_T)\rho_s K_{s\ell}}$

Noncontinuous } **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + s_\ell (1 - n_T)\rho_s K_{s\ell}}$

IfDef: Radioactive
Half-Life[f], Units[g] (s),
Format: *Char[a], Char[b], Real[c], Char[d], Char[e], Real[f], Char[g],*
If: Operational Mode Option ={ Water Courant Vadose Transport }
Aqueous-Phase Cut-off Conc[h], Units[i] (1/m^3),
Format: *Char[a], Char[b], Real[c], Char[d], Char[e], Real[f], Char[g], Real[h], Char[i],*
Endif:
ElseifDef: Reactive
Number of Reactions[f],
For: Number of Reactions:
First-Order Reaction Rate Constant[g], Units[h] (s),
Endfor: Number of Reactions
Format: *Char[a], Char[b], Real[c], Char[d], Char[e], Integer[f],* <*Real[g], Char[h],*>
EndifDef:
Endif:

Elseif: Operational Mode Option = { **STOMP-WA** } { **STOMP-WAE** }
{ **STOMP-WAE-B** } { **STOMP-WAE-Sc** } { **STOMP-WAS** } { **STOMP-WASE** }
Solute Name[a],
Aqueous-Phase Molecular Diffusion Coefficient @ 20 C[b], Units[c] (m^2/s),
Gas-Phase Molecular Diffusion Coefficient @ 20 C[d], Units[e] (m^2/s),
Gas-Aqueous Partition Function Option[f]

{ Constant | **Note:** $K_{g\ell} = \overline{K}_{gl}$

Temperature Dependent } **Note:** $\ln\left(K_{g\ell}\right) = a + \dfrac{b}{T} + c\ln(T) + dT + eT^2$

If: Gas-Aqueous Partition Function Option = { Constant }
Gas-Aqueous Partition Coefficient[g], Units[h] (m^3/m^3),
Solute-Aqueous Partition Option[i],

{ Continuous | **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + (1 - n_T)\rho_s K_{s\ell}}$

Noncontinuous } **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + s_\ell (1 - n_T)\rho_s K_{s\ell}}$

IfDef: Radioactive
Half-Life[j], Units[k] (s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Reals[g], Char[h], Char[i], Real[j], Char[k],*
ElseifDef: Reactive
Number of Parent Reactions[j],
For: Number of Reactions
First-Order Reaction Rate Constant[k], Units[l] (s),
Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Reals[g], Char[h], Char[i], Integer[j],*
<Real[k], Char[l],>
EndifDef:

Elseif: Gas-Aqueous Partition Function Option = { Temperature Dependent }
Constant a[g], Constant b[h], Constant c[i], Constant d[j], Constant e[k],
Solute-Aqueous Partition Option[l],

{ Continuous | **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + (1 - n_T)\rho_s K_{s\ell}}$

Noncontinuous } **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + s_\ell (1 - n_T)\rho_s K_{s\ell}}$

IfDef: Radioactive
Half-Life[m], Units[n] (s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Reals[g], Real[h], Real[i], Real[j],*
Real[k], Char[l], Real[m], Char[n],
ElseifDef: Reactive
Number of Parent Reactions[m],
For: Number of Reactions
First-Order Reaction Rate Constant[n], Units[o] (s),

Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Reals[g], Real[h], Real[i], Real[j], Real[k], Integer[l],*
Char[m], <Real[n], Char[o],>
EndifDef:
Endif:

Elseif: Operational Mode Option = { **STOMP-WO }**
Solute Name[a], Aqueous-Phase Molecular Diffusion Coefficient @ 20 C [b], Units[c] (m^2/s),
NAPL Molecular Diffusion Coefficient @ 20 C[d], Units[e] (m^2/s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*

Aqueous-NAPL Adsorpton Function[a], { Linear | Freundlich | Langmuir }
If: Aqueous-NAPL Adsorpton Function = { Linear }
Linear k[b],
Format: *Char[a], Real[b],*
Elseif: Aqueous-NAPL Adsorpton Function = { Freundlich }
k parameter[b], n parameter[c],
Format: *Char[a], Real[b], Real[c],*
Elseif: Aqueous-NAPL Adsorpton Function = { Langmuir }
a parameter[b], b parameter[c], Units[d] (m^3),
Format: *Char[a], Real[b], Real[c], Char[d],*
Endif:
IfDef: Radioactive
Half-Life[a], Units[b] (s),
Format: *Real[a], Char[b],*
ElseifDef: Reactive
Number of Parent Reactions[a]
For: Number of Reactions
First-Order Reaction Rate Constant[b], Units[c] (s),
Endfor: Number of Reactions
Format: *Integer[a], <Real[b], Char[c],>*
EndifDef:

Elseif: Operational Mode Option = { **STOMP-WOD** } { **STOMP-WOM** }
Solute Name[a], Aqueous-Phase Molecular Diffusion Coefficient[b], Units[c] (m^2/s),
NAPL Molecular Diffusion Coefficient[d] @ 20 C, Units[e] (m^2/s),
Aqueous-NAPL Partition Coefficient[f], Units[g] (m^3/m^3),
IfDef: Radioactive
Half-Life[h], Units[i] (s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i],*
ElseifDef: Reactive
Number of Parent Reactions[h],
For: Number of Reactions
First-Order Reaction Rate Constant[i], Units[j] (s),
Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g],Integer[h], <Real[i], Char[j],>*

EndifDef:

Elseif: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc } { STOMP-WOAE }**
Solute Name[a],
Aqueous-Phase Molecular Diffusion Coefficient[b] @ 20 C, Units[c] (m^2/s),
Gas-Phase Molecular Diffusion Coefficient[d] @ 20 C, Units[e] (m^2/s),
NAPL Molecular Diffusion Coefficient[f] @ 20 C, Units[g] (m^2/s),
Gas-Aqueous Partition Function Option[h],

Solute/Fluid Interactions Card (cont'd)

{ Constant | **Note:** $K_{g\ell} = \overline{K}_{gl}$

Temperature Dependent } **Note:** $\ln\left(K_{g\ell}\right) = a + \frac{b}{T} + c\ln\left(T\right) + dT + eT^2$

If: Gas-Aqueous Partition Function Option = { Constant }
Gas-Aqueous Partition Coefficient[i], Units[j] (m^3/m^3),
Aqueous-NAPL Partition Coefficient[k], Units[l] (m^3/m^3),
IfDef: Radioactive
Half-Life[m], Units[n] (s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], eal[m], Char[n],*
ElseifDef: Reactive
Number of Parent Reactions[m],
For: Number of Reactions
First-Order Reaction Rate Constant[n], Units[o] (s),
Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Real[i],Char[j], Real[k], Char[l],Integer[m],* <*Real[n], Char[o],*>
EndifDef:
Elseif: Gas-Aqueous Partition Function Option = { Temperature Dependent }
Constant a[i], Constant b[j], Constant c[k], Constant d[l], Constant e[m],
Aqueous-NAPL Partition Coefficient[n], Units[o] (m^3/m^3),
IfDef: Radioactive
Half-Life[p], Units[q] (s)
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l],Real[m], Real[n], Char[o], Real[p], Char[q],*
ElseifDef: Reactive
Number of Parent Reactions[p],
For: Number of Reactions
First-Order Reaction Rate Constant[q], Units[r] (s),
Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Real[i], Real[j], Real[k], Real[l], Real[m], Real[n], Char[o], Integer[p],* <*Real[q], Char[r],*>
EndifDef:
Endif:

Elseif: Operational Mode Option Card = { **STOMP-WS** } { **STOMP-WS-Sc** }
Solute Name[a],
Effective Diffusion Option[b],

{ Constant | **Note:** $D^C_{\ell\ e} = \overline{D}^C_{\ell\ e}$

Conventional | **Note:** $D^C_{\ell\ e} = \tau_\ell s_\ell n_D D^C_\ell$

Empirical [Kemper and van Schaik] } **Note:** $D_{\ell\ e}^{C} = D_{\ell}^{C}\, a\exp\left(bn_D s_\ell\right)$

Model parameters for the Empirical option are entered in the Solute/Porous Medium Interaction Card

If: Effective Diffusion Option = { Empirical }
Solute-Aqueous Partition Option[c],

{ Continuous | **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + (1-n_T)\rho_s K_{s\ell}}$

Noncontinuous } **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + s_\ell (1-n_T)\rho_s K_{s\ell}}$

IfDef: Radioactive
Half-Life[d], Units[e] (s),
Format: *Char[a], Char[b], Char[c], Real[d], Char[e],*
ElseifDef: Reactive
Number of Reactions[d],
For: Number of Reactions:
First-Order Reaction Rate Constant[e], Units[f] (s),
Endfor: Number of Reactions
Format: *Char[a], Char[b], Char[c], Integer[d],<Real[e], Char[f],>*
EndifDef:
Else:
Aqueous-Phase Molecular Diffusion Coefficient @ 20 C[c], Units[d] (m^2/s),
Solute Partition Option[e],

{ Continuous | **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + (1-n_T)\rho_s K_{s\ell}}$

Noncontinuous } **Note:** $C_\ell = \dfrac{C}{n_D s_\ell + s_\ell (1-n_T)\rho_s K_{s\ell}}$

IfDef: Radioactive
Half-Life[f], Units[g] (s),
Format: *Char[a], Char[b], Real[c], Char[d], Char[e], Real[f], Char[g],*
ElseifDef: Reactive
Number of Reactions[f],
For: Number of Reactions:
First-Order Decay Constant[g], Units[h] (s),
Endfor: Number of Reactions
Format: *Char[a], Char[b], Real[c], Char[d], Char[e], Integer[f], <Real[g], Char[h],>*
EndifDef:
Endif:

Solute/Fluid Interactions Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** } { **STOMP-WCMSE** }
Solute Name[a],
Aqueous-Phase Molecular Diffusion Coefficient @ 20 C[b], Units[c] (m^2/s),
Gas-Phase Molecular Diffusion Coefficient @ 20 C[d], Units[e] (m^2/s),
Gas-Aqueous Partition Function Option[f]

{ Constant | **Note:** $K_{g\ell} = \overline{K}_{gl}$

Temperature Dependent } **Note:** $\ln\left(K_{g\ell}\right) = a + \frac{b}{T} + c\ln\left(T\right) + dT + eT^2$

If: Gas-Aqueous Partition Function Option = { Constant }
Gas-Aqueous Partition Coefficient[g], Units[h] (m^3/m^3),
IfDef: Radioactive
Half-Life[i], Units[j] (s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
ElseifDef: Reactive
Number of Parent Reactions[j],
For: Number of Reactions
First-Order Reaction Rate Constant[k], Units[l] (s),
Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Char[h], Char[i], Integer[j], <Real[k], Char[l],>*
EndifDef:

Elseif: Gas-Aqueous Partition Function Option = { Temperature Dependent }
Constant a[g], Constant b[h], Constant c[i], Constant d[j], Constant e[k],
IfDef: Radioactive
Half-Life[l], Units[m] (s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h], Real[i], Real[j], Real[k], Real[l], Char[m],*
ElseifDef: Reactive
Number of Parent Reactions[l],
For: Number of Reactions
First-Order Reaction Rate Constant[m], Units[n] (s),
Endfor: Number of Reactions
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Char[f], Real[g], Real[h],Real[i], Real[j], Real[k], Integer[l], <Real[m], Char[n],>*
EndifDef:
Endif:

Endif:

Endfor: Number of Solutes

Solute/Fluid Interactions Card (cont'd)

IfDef: Radioactive
  Number of Chain Decay Lines[a],
  Format: *Integer[a],*
  For: Number of Chain Decay Lines
    Parent Solute Name[a], Progeny Solute Name[b], Chain Decay Fraction[c],
    Format: *Char[a], Char[b], Real[c],*
  Endfor: Number of Chain Decay Lines
ElseifDef: Reactive
  Number of Reaction Lines[a],
  Format: *Integer[a],*
  For: Number of Reaction Lines
    Parent Solute Name[a], Progeny Solute Name[b], Reaction Number[c],
    Reaction Stoichiometry Fraction[d],
  Format: *Char[a], Char[b], Real[c], Real[d],*
  Endfor: Number of Reaction Lines
EndifDef

Endcard: Solute/Fluid Interactions Card

B.31.1 Solute/Fluid Interaction Card Examples

Extracted from a STOMP-W input file:
~Solute/Fluid Interaction Card
1,
TCE,Conventional,9.6283e-6,cm^2/s,Continuous,1000,yr,
0,

Extracted from a STOMP-W input file:
~Solute/Fluid Interaction Card
4,
Tc-99,Empirical,Noncontinuous,1.e20,yr,
U-238,Empirical,Noncontinuous,1.e20,yr,
Np-237,Empirical,Noncontinuous,1.e20,yr,
Pu-239,Empirical,Noncontinuous,1.e20,yr,
0,

Extracted from a STOMP-W input file:
~Solute/Fluid Interaction Card
7,
Dithionite,Conventional,1.e-5,cm^2/s,Continuous,2,5,hr,18,hr,
Sulfite,Conventional,1.e-5,cm^2/s,Continuous,0,
Bisulfite,Conventional,1.e-5,cm^2/s,Continuous,0,
Thiosulfate,Conventional,1.e-5,cm^2/s,Continuous,0,
Fe(III),Conventional,1.e-5,cm^2/s,Continuous,0,
Fe(II),Conventional,1.e-5,cm^2/s,Continuous,0,
H+,Conventional,1.e-5,cm^2/s,Continuous,0,
6,
Dithionite,Fe(III),1,-2.0,
Dithionite,Fe(II),1,2.0,
Dithionite,Sulfite,1,2.0,
Dithionite,H+,1,4.0,
Dithionite,Bisulfite,2,1.0,
Dithionite,Thiosulfate,2,0.5,

Extracted from a STOMP-WO input file:
~Solute/Fluid Interactions Card
Note different format for Water-Oil mode!
3,
Tritium,1.0e-10,m^2/s,1.0e-10,m^2/s,
Linear Isotherm,0.0,
1.e+8,yr,
IPA,1.0e-10,m^2/s,1.0e-10,m^2/s,
Linear Isotherm,0.04,
1.e+8,yr,
DMB,1.0e-10,m^2/s,1.0e-10,m^2/s,
Linear Isotherm,2.76,
1.e+8,yr,
0,

B.32 Solute/Porous Media Interactions Card

Card Title[a] { ~Solute/Porous [Media Interactions Card] }
Format: *Char[a]*

Rock/Soil or Scaling Group Name[a],

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: *filename*

or the following formattings for binary files:

binary file: *filename*

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil Types

Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m)
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*

If: Operational Mode Option = **{ STOMP-W } { STOMP-W-R } { STOMP-W-Sc }**
For: Number of Solutes
Solute Name[a],
Solid-Aqueous Partition Coefficient[b], Units[c] (m^3/kg),
If: Effective Diffusion Option: Solute/Fluid Interactions Card = { Empirical }
Aqueous Molecular Diffusion Coefficient[d], Units[e] (m^2/s),
Solute/Fluid Interactions Card: Constant a[f],
Solute/Fluid Interactions Card: Constant b[g],
[Macrodispersity Enhancement Factor[h],]
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Real[g], [Real[h],]*
Else:
[Macrodispersivity Enhancement Factor[d],]
Format: *Char[a], Real[b], Char[c], [Real[d],]*
Endif:
Endfor: Number of Solutes

Solute/Porous Media Interactions Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WS }**
For: Number of Solutes
Solute Name[a],
Solid-Aqueous Partition Coefficient[b], Units[c] (m^3/kg),
If: Effective Diffusion Option: Solute/Fluid Interactions Card = { Empirical }
Aqueous Molecular Diffusion Coefficient[d], Units[e] (m^2/s),
Solute/Fluid Interactions Card: Constant a[f],
Solute/Fluid Interactions Card: Constant b[g],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Real[g],*
Else:
Format: *Char[a], Real[b], Char[c],*
Endif:
Endfor: Number of Solutes

Elseif: Operational Mode Option Card = **{ STOMP-WA } { STOMP-WAE } { STOMP-WO } { STOMP-WO-We } { STOMP-WOD } { STOMP-WOM } { STOMP-WOA } { STOMP-WOA-Sc }{ STOMP-WOAE }{ STOMP-WAS } { STOMP-WASE } { STOMP-WCS } { STOMP-WCS-R } { STOMP-WCS-R-Sc } { STOMP-WCS-Sc } { STOMP-WCSE } { STOMP-WCMSE }**
For: Number of Solutes
Solute Name[a],
Solid-Aqueous Partition Coefficient[b], Units[c] (m^3/kg),
Format: *Char[a], Real[b], Char[c],*
Endfor: Number of Solutes

Endfor: Number of Rock/Soil Types

Endif:

Endcard: Solute/Porous Media Interactions Card

B.32.1 Solute/Porous Media Interactions Card Examples

Extracted from a STOMP-W input file:

```
~Solute/Porous Media Interaction Card
Backfill Soil,,,,,
Tc-99,0.,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
U-238,0.67,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Np-237,3.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Pu-239,21.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Hanford Sand,,,,,
Tc-99,0.,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
U-238,0.67,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Np-237,3.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Pu-239,21.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Hanford Gravel,,,,,
Tc-99,0.,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
U-238,0.67,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Np-237,3.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Pu-239,21.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Ringold Formation,,,,,
Tc-99,0.,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
U-238,0.67,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Np-237,3.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Pu-239,21.0,cm^3/g,7.8894e-2,m^2/yr,0.005,10.,
Glass Waste,,,,,
Tc-99,0.,cm^3/g,3.1557e-7,m^2/yr,1.,0.,
U-238,0.,cm^3/g,3.1557e-7,m^2/yr,1.,0.,
Np-237,0.,cm^3/g,3.1557e-7,m^2/yr,1.,0.,
Pu-239,0.,cm^3/g,3.1557e-7,m^2/yr,1.,0.,
```

Extracted from a STOMP-W input file:

```
~Solute/Porous Media Interaction Card
Soil,1,m,0.1,m,
Dithionite,0.0,m^3/kg,
Sulfite,0.0,m^3/kg,
Bisulfite,0.0,m^3/kg,
Thiosulfate,0.0,m^3/kg,
Fe(III),0.0,m^3/kg,
Fe(II),0.0,m^3/kg,
H+,0.0,m^3/kg,
```

Extracted from a STOMP-W input file:

```
~Solute/Porous Media Interaction Card
Backfill,150,cm,15,cm,
U:0.01, 0.01,ml/g,1.067,
U:0.03, 0.03,ml/g,1.067,
U:0.10, 0.10,ml/g,1.067,
U:0.30, 0.30,ml/g,1.067,
U:0.60, 0.60,ml/g,1.067,
U:1.00, 1.00,ml/g,1.067,
Tc, 0.00,ml/g,1.0,
```

Solute/Porous Media Interaction Card (cont'd)

H2 Sand,150,cm,15,cm,
U:0.01, 0.01,ml/g,1.063,
U:0.03, 0.03,ml/g,1.063,
U:0.10, 0.10,ml/g,1.063,
U:0.30, 0.30,ml/g,1.063,
U:0.60, 0.60,ml/g,1.063,
U:1.00, 1.00,ml/g,1.063,
Tc, 0.00,ml/g,1.0,
H1 Gravelly Sand,100,cm,10,cm,
U:0.01, 0.01,ml/g,1.120,
U:0.03, 0.03,ml/g,1.120,
U:0.10, 0.10,ml/g,1.120,
U:0.30, 0.30,ml/g,1.120,
U:0.60, 0.60,ml/g,1.120,
U:1.00, 1.00,ml/g,1.120,
Tc, 0.00,ml/g,1.0,
H3 Gravelly Sand,100,cm,10,cm,
U:0.01, 0.01,ml/g,1.062,
U:0.03, 0.03,ml/g,1.062,
U:0.10, 0.10,ml/g,1.062,
U:0.30, 0.30,ml/g,1.062,
U:0.60, 0.60,ml/g,1.062,
U:1.00, 1.00,ml/g,1.062,
Tc, 0.00,ml/g,1.0,

B.33 Solution Control Card

Card Title[a] {~Solution [Control Card] }
Format: *Char[a]*

Execution Mode Option[a],
If: Operational Mode Option = **{ STOMP-W } { STOMP-W-R}**
Execution Mode Option[a],
{ Normal [No Flow] [Dynamic Domain] [Second Order] [Scaling] [Inverse] |
Restart [No Flow] [Dynamic Domain] [File] [Mode] [Second Order]
[Scaling] [Inverse] |
Initial Conditions [Second Order] [Scaling] [Inverse] }
Elseif: Operational Mode Option = **{ STOMP-W-Sc }**
Execution Mode Option[a],
{ Normal [No Flow] [Dynamic Domain] [Second Order] [Scaling] [Inverse] [PETSc] |
Restart [No Flow] [Dynamic Domain] [File] [Mode] [Second Order] [PETSc] |
[Scaling] [Inverse] | Initial Conditions }
Elseif : Operational Mode Option = **{ STOMP-WAE-Sc } { STOMP-WOA-Sc }**
{ STOMP-WS-Sc } { STOMP-WCS-R-Sc }
Execution Mode Option[a],
{ Normal [Second Order] [PETSc] |
Restart [File] [Second Order] [PETSc] |
Initial Conditions }
Else:
Execution Mode Option[a],
{ Normal [Second Order] |
Restart [File] [Second Order] |
Initial Conditions }
Endif:

If: Execution Mode Option = { Restart File }
Restart File Name[b],
Format: *Char[a], Char[b],*
Else:
Format: *Char[a],*
Endif:

Note: For Operation Mode Options, H2O can be used instead of Water, NaCL instead of Salt, CH4 for Hydrate, and E for Energy.

Operational Mode Option[a],
{ Water
[[Globally] | [Guess]] | EckeChem | [Electrolyte] | [No Vapor Pressure Lowering] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection]] Transport
Note: Mode indication for **STOMP-W**, **STOMP-W-R** and **STOMP-W-Sc** modes.

Solution Control Card (cont'd)

Water-Air [No Vapor Pressure Lowering] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport

Water-Air-Energy
[Time Lag] | [No Vapor Pressure Lowering] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport
Note: Mode indication for **STOMP-WAE**, **STOMP-WAE-B** and **STOMP-WAE-Sc** modes.

Water-Oil [No Vapor Pressure Lowering] |
[[Partitioning] | [Kinetic]] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No NAPL Advection]] Transport
Note: Mode indication for **STOMP-WO, STOMP-WOD,** and **STOMP-WOM** modes.

Water-Air-Oil [No Vapor Pressure Lowering] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection] |
[No NAPL Advection]] Transport
Note: Mode indication for **STOMP-WOA** and **STOMP-WOA-Sc** modes.

Water-Air-Oil-Energy [No Vapor Pressure Lowering] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection] |
[No NAPL Advection]] Transport

Water-Oil-Dissolved Oil [No Vapor Pressure Lowering] |
[[Dissolution] | [Dissolved Oil]] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No NAPL Advection]] Transport

Water-Salt [No Vapor Pressure Lowering] |
[LFL] | [Osmotic [Surface | Tension]] | [Surface | Tension] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection]] Transport
Note: Mode indication for **STOMP-WS** and **STOMP-WS-Sc** modes.

Water-Air-Salt [No Vapor Pressure Lowering] |
[LFL] | [Osmotic [Surface | Tension]] | [Surface | Tension] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport

Solution Control Card (cont'd)

Water-Air-Salt-Energy [No Vapor Pressure Lowering] |
[LFL] | [Osmotic [Surface | Tension]] | [Surface | Tension] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport

Water-CO2-NaCl [No Vapor Pressure Lowering] |
[LFL] | [Osmotic [Surface | Tension]] | [Surface | Tension] | [Poynting] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport |
[[Globally] | [Guess]] EckeChem |
Note: Mode indication for **STOMP-WCS, STOMP-WCS-R, STOMP-WCS-R-Sc** and **STOMP-WCS-Sc** modes.

Water-CO2-NaCl-Energy [No Vapor Pressure Lowering] | [LFL] [Osmotic]
[Surface [Tension]] |
[LFL] | [Osmotic [Surface | Tension]] | [Surface | Tension] | [Poynting] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport

Water-CO2-CH4-Salt-Energy [No Vapor Pressure Lowering] |
[LFL] | [Osmotic [Surface | Tension]] | [Surface | Tension] |
[Isobrine] | [Poynting] |
[[[Leonard] | [TVD]] | [[Roe] | [Superbee]] | [[First-Order] | [Upwind]] |
[[Vadose] Courant] | [No Aqueous Advection] | [No Gas Advection]] Transport }

If: Maximum Courant Number is specified (optional):
Maximum Courant Number[b],
Format: *Char[a], Real[b]*,
Else:
Format: *Char[a]*,
Endif:

If: Execution Mode Option = { Initial Conditions }
Endcard: Solution Control Card
Endif:

Number of Execution Time Periods[a],
Format: *Integer[a]*,

For: Number of Execution Time Periods
If: Execution Mode Option = { Normal }
Initial Time[a], Units[b] (s), Final Time[c], Units[d] (s),
Initial Time Step[e], Units[f] (s),
Maximum Time Step[g], Units[h] (s),
Time Step Acceleration Factor[i],
Maximum Number of Newton-Raphson Iterations[j], Convergence Criterion[k],

Solution Control Card (cont'd)

Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Integer[j], Real[k],*
ElseIf: Execution Mode Option = { Restart }
Initial Time[a], Units[b] (s), Final Time[c], Units[d] (s),
Initial Time Step[e], Units[f] (s),
Maximum Time Step[g], Units[h] (s),
Time Step Acceleration Factor[i],
Maximum Number of Newton-Raphson Iterations[j],
Convergence Criterion[k],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Integer[j], Real[k],*
Endif:
Endfor: Number of Execution Time Periods

Maximum Number of Time Steps[a]
Format: *Integer[a],*

If: Operational Mode Option = **{ STOMP-WA } { STOMP-WAE } { STOMP-WAE-B } { STOMP-WAE-Sc }**
Aqueous Diffusion Option[a], { Zero } { Constant } { Variable }
If: Aqueous Diffusion Option = { Constant }
Dissolved Air Diffusion Coefficient[b], Units[c] (m^2/s),
Format: *Char[a], Real[b], Char[c],*
Else:
Format: *Char[a],*
Endif:
Endif:

If: Operational Mode Option = **{ STOMP-WO } { STOMP-WOD } { STOMP-WOM }**
Aqueous Diffusion Option[a], { Zero } { Constant } { Variable }
If: Aqueous Diffusion Option = { Constant }
Dissolved Oil Diffusion Coefficient[b], Units[c] (m^2/s),
Format: *Char[a], Real[b], Char[c],*
Else:
Format: *Char[a],*
Endif:
Endif:

If: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc } { STOMP-WOAE }**
Aqueous Diffusion Option[a], { Zero } { Constant } { Variable }
If: Aqueous Diffusion Option = { Constant }
Dissolved Air Diffusion Coefficient[b], Units[c] (m^2/s),
Dissolved Oil Diffusion Coefficient[d], Units[e] (m^2/s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Else:
Format: *Char[a],*
Endif:
Endif:

Solution Control Card (cont'd)

If: Operational Mode Option = { **STOMP-WAS** } { **STOMP-WASE** }
  Aqueous Diffusion Option[a], { Zero } { Constant } { Variable }
  If: Aqueous Diffusion Option = { Constant }
    Dissolved Air Diffusion Coefficient[b], Units[c] (m^2/s),
    Dissolved Salt Diffusion Coefficient[d], Units[e] (m^2/s),
    Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
  Else:
    Format: *Char[a],*
  Endif:
Endif:

If: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-R** }
  { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** } { **STOMP-WCSE** } { **STOMP-WCMSE** }
  Aqueous Diffusion Option[a], { Zero } { Constant } { Variable }
  If: Aqueous Diffusion Option = { Constant }
    Dissolved CO_2 Diffusion Coefficient[b], Units[c] (m^2/s),
    Dissolved Salt Diffusion Coefficient[d], Units[e] (m^2/s),
    Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
  Else:
    Format: *Char[a],*
  Endif:
Endif:

If: Operational Mode Option = { **STOMP-WA** } { **STOMP-WAE** } { **STOMP-WAE** }
  { **STOMP-WAE** } { **STOMP-WAS** } { **STOMP-WASE** } { **STOMP-WCS** }
  { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** }
  { **STOMP-WCSE** }
    Gas Diffusion Option[a], { Zero } { Constant } { Variable } { Enhanced }
  If: Gas Diffusion Option = { Constant }
    Water Vapor Diffusion Coefficient[b], Units[c] (m^2/s),
    Format: *Char[a], Real[b], Char[c],*
  Elseif: Gas Diffusion Option = { Enhanced }
    Clay Mass Fraction[b],
    Format: *Char[a], Real[b],*
  Endif:
Endif:

If: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc } { STOMP-WOAE }**
    Gas Diffusion Option[a], { Zero } { Constant } { Variable } { Enhanced }
  If: Gas Diffusion Option = { Constant }
    Water Vapor Diffusion Coefficient[b], Units[c] (m^2/s),
    Oil Vapor Diffusion Coefficient[d], Units[e] (m^2/s),
    Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
  Elseif: Gas Diffusion Option = { Enhanced }
    Clay Mass Fraction[b],
    Format: *Char[a], Real[b],*

Solution Control Card (cont'd)

Endif:
Endif:

If: Operational Mode Option = { **STOMP-WCMSE** }
Gas Diffusion Option[a], { Zero } { Constant } { Variable } { Enhanced }
If: Gas Diffusion Option = { Constant }
CO_2 Vapor Diffusion Coefficient[b], Units[c] (m^2/s),
CH_4 Vapor Diffusion Coefficient[d], Units[e] (m^2/s),
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Elseif: Gas Diffusion Option = { Enhanced }
Clay Mass Fraction[b],
Format: *Char[a], Real[b],*
Endif:

Hydrate Dissociation-Molecular Exchange Option[a] = { Equilibrium } { Kinetic }
If: Exchange Option = { Kinetic }
Hydrate Formation-Dissociation Rate Constant[b], Units[c],
Hydrate Molecular Exchange Rate Constant[c], Units[e],
Format: *Char[a], Real[b], Char[c], Real[d], Char[e],*
Else:
Format: *Char[a],*
Endif:
Endif:

If: Operational Mode Option = { **STOMP-W w/ Electrolyte Solute Transport** }
{ **STOMP-W-R w/ Electrolyte Solute Transport** }
{ **STOMP-W-Sc w/ Electrolyte Solute Transport** }
Electrolyte Name[a],
Format: *Char[a],*
Electrolyte Density Function Option[a] = { Leijnse | Fourth }
If: Electrolyte Density Function Option = { Leijnse }
Exponential Coefficient[b],
Format: *Char[a], Real[b],*
Elseif: Electrolyte Density Function Option = { Fourth }
Units[b] (1/m^3), Polynomial "a" Coefficient[c], Polynomial "b" Coefficient[d],
Polynomial "c" Coefficient[e], Polynomial "d" Coefficient[f],
Format: *Char[a], Char[b], Real[c] ,Real[d] ,Real[e], Real[f],*
Endif:
Electrolyte Viscosity Function Option[a] = { Leijnse | Fourth }
If: Electrolyte Viscosity Function Option = { Leijnse }
Units[b] (1/m^3), Leijnse "a" Coefficient[c], Leijnse "b" Coefficient[d],
Leijnse "c" Coefficient[e], Leijnse "d" Coefficient[f],
Format: *Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],*
Elseif: Electrolyte Viscosity Function Option = { Fourth }
Units[b] (1/m^3), Polynomial "a" Coefficient[c], Polynomial "b" Coefficient[d],

Polynomial "c" Coefficient[e], Polynomial "d" Coefficient[f],

Solution Control Card (cont'd)

Format: *Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],*
Endif:
Endif:

Number of Interfacial Averaging Variables[a]
Format: *Integer[a],*

For: Number of Interfacial Averaging Variables
Surface Variable Option[a],
{ Air Gas Diffusion | Air Aqueous Diffusion |
Aqueous Density | Aqueous Relative Permeability |
Aqueous Viscosity | Effective Permeability |
Gas Density | Gas Relative Permeability |
Gas Viscosity | Hydraulic Dispersion |
Intrinsic Permeability | NAPL Density |
NAPL Relative Permeability | NAPL Viscosity |
Solute Diffusion | Thermal Conductivity |
Oil Gas Diffusion | Oil Aqueous Diffusion |
Salt Aqueous Diffusion | Solute Diffusion |
Water Gas Diffusion }
Interfacial Averaging Scheme Option[b]
{ Harmonic | Geometric | Arithmetic | Upwind | Downstream | Neiber Downstream }
If: Interfacial Averaging Scheme Option = { Downstream | Neiber Downstream }
Weighting Factor[c],
Format: *Char[a], Char[b], Real[c],*
Else:
Format: *Char[a], Char[b],*
Endif:
Endfor: Number of Interfacial Averaging Variables

Endcard: Solution Control Card

B.33.1 Solution Control Card Examples

Extracted from STOMP-W input file:
~Solution Control Card
Restart,
Water w/transport vadose courant,
1,
2050,yr,12000,yr,0.01,yr,1.0,yr,1.25,8,1.e-06,
20000,
0,

Extracted from STOMP-W-Sc input file:
~Solution Control Card
Restart,
Water w/ PETSc,
1,
0,yr,75,hr,0.1,hr,0.1,hr,1.25,8,1.e-06,
1000,
0,

Extracted from STOMP-W input file:
~Solution Control Card
Normal,
Water w/ Solute Transport,
1,
0,min,6816.95,min,1,s,20,min,1.25,8,1.e-06,
1000,
0,

Extracted from STOMP-W input file:
~Solution Control Card
Normal w/ Scaling w/ Inverse,
Water,
1,
0,s,1980000,s,0.1,s,1800,s,1.25,8,1.e-06,
1000,
0,

Extracted from STOMP-W input file:
~Solution Control Card
Normal,
Water w/TVD transport w/electrolyte courant,0.20,
2,
0,hr,22.2,hr,0.0001,hr,0.1,hr,1.25,8,1.E-6,
22.2,hr,94,hr,0.001,hr,0.1,hr,1.25,8,1.E-6,
100000,
TDS,
Fourth-Order,1/L,1.0,0.000825,,,
Fourth-Order,1/L,1.0,,,,
0,

Solution Control Card Examples (cont'd)

```
#------------------------------------------------
```
Extracted from STOMP-W input file:
```
~Solution Control Card
#------------------------------------------------
Normal,
Water w/TVD transport w/electrolyte courant,0.5,
16,
147.5,day,148,day,1.0,hr,12,hr,1.25,8,1.E-6,
148,day,149,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
149,day,153,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
153,day,154.446,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
154.446,day,192,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
192,day,193.77,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
193.77,day,195,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
195,day,196.38,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
196.38,day,197,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
197,day,198.45,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
198.45,day,216,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
216,day,217.57,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
217.57,day,218,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
218,day,219.35,day,0.001,hr,0.02,hr,1.25,8,1.E-6,
219.35,day,221,day,0.02,hr,0.1,hr,1.25,8,1.E-6,
221,day,226,day,0.001,hr,0.1,hr,1.25,8,1.E-6,
100000,
TDS,
# density=ref_density*(a + bx + cx^2 + dx^3)
# x = aqueous concentration
Fourth-Order,1/L,1.0,0.000825,,,
#Fourth-Order,1/L,1.0,0.00078,,,
# viscosity=ref_viscosity*(a + bx + cx^2 + dx^3)
# x = aqueous concentration
Fourth-Order,1/L,1.0,,,,
0,
```

Extracted from STOMP-W input file:
```
#2
~Solution Control Card
Restart,
Water w/transport vadose courant,
1,
2000,yr,12032,yr,0.001,yr,1.0,yr,1.25,15,1.e-06,
20000,
0,
```

Extracted from STOMP-W-R input file:
```
~Solution Control Card
Restart,
Water w/ ECKEChem,
1,
0,yr,75,hr,0.1,hr,0.1,hr,1.25,8,1.e-06,
```

1000,
0,

Extracted from STOMP-W-R input file:
~Solution Control Card
Restart,
Water w/ ECKEChem w/Guess,
1,
0,day,0.35,day,0.0005,day,0.0005,day,1.25,8,1.e-06,
10000,
0,

Extracted from STOMP-WA input file:
~Solution Control Card
Restart file, TheRestart,
Water-Air Transport,
1,
0,day,200,day,10,s,50,day,1.25,8,1.e-6,
10000,
Constant Aqueous Diffusion, 1.8e-05,cm^2/s,
Constant Gas Diffusion, 1.5e-02,cm^2/s,
,

Extracted from STOMP-WA input file:
~Solution Control Card
Normal,
Water-Air Transport using no vapor pressure lowering,
1,
0,d,150,d,1,s,5,min,1.25,12,1.e-6,
1,yr,1,yr,50000,
variable,
variable,
0,

Extracted from STOMP-WAE input file:
~Solution Control Card
Restart Mode,1,
Water-Air-Energy,
1,
0,yr,500,yr,1,sec,50,yr,1.25,8,1.e-06,
10000,
Variable,
Variable,
0,

Extracted from STOMP-WAE input file:
~Solution Control Card
Normal,
Water-Air-Energy,
1,

0,day,876.6,day,10,S,100,day,1.25,16,1.E-06,
Solution Control Card Examples (cont'd)

1000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
Interfacial average defaults
3,
Gas Relative Permeability, Arithmetic,
Aqueous Relative Permeability, Harmonic,
Intrinsic Permeability, Geometric,

Extracted from STOMP-WO input file:
~Solution Control Card
Normal,
Water-Oil,
2,
0,s,10,d,1,s,1,d,1.25,8,1.e-6,
5,d,10,d,1,s,1,d,1.25,8,1.e-6,
1000,
Variable Aqueous Diffusion,
0,

Extracted from STOMP-WOA input file:

~Solution Control Card

Normal,
Water-Oil-Air,,
1,
0,yr,1954,yr,1,yr,200,yr,1.25,16,1.e-6,
100000,
Variable,
Constant,0.9e-6,m^2/s,0.9e-6,m^2/s,
0,

Extracted from STOMP-WOA-Sc input file:

~Solution Control Card

Restart File w/PETSc,restart90.781,
Water-Oil-Air,,
2,
,,2001.5,yr,,,1,yr,1.25,16,1.e-6,
2001.5,yr,2004,yr,0.1,yr,1,yr,1.25,16,1.e-6,
#6,
#1954,yr,1955.5,yr,0.1,yr,1,yr,1.25,16,1.e-6,
#1955.5,yr,1962.5,yr,0.0001,yr,1,yr,1.25,16,1.e-6,
#1962.5,yr,1993.2444,yr,0.1,yr,1,yr,1.25,16,1.e-6,
#1993.2444,yr,1995,yr,0.0001,yr,1,yr,1.25,16,1.e-6,
#1995,yr,2001.5,yr,0.01,yr,1,yr,1.25,16,1.e-6,
#2001.5,yr,2004,yr,0.1,yr,1,yr,1.25,16,1.e-6,

10000,
Variable,
Solution Control Card Examples (cont'd)

Constant,0.9e-6,m^2/s,0.9e-6,m^2/s,
0,

Extracted from STOMP-WS input file:
~Solution Control Card
Normal,
Water-Salt w/ Surface Tension Effects,
1,
0,day,1.0,hr,1.e-5,min,10.0,min,1.25,8,1.e-06,
1000,
0,

Extracted from STOMP-WCS input file:
~Solution Control Card
Normal,
H2O-CO2-NaCl,
1,
0,day,14600,day,1,hr,40,day,1.25,16,1.e-06,
10000,
Constant Aqueous Diffusion,0.0,ft^2/day,0.0,ft^2/day,
Constant Aqueous Diffusion,4.0e-3,ft^2/day,4.0e-3,ft^2/day,
0,

Extracted from STOMP-WCSE input file:
~Solution Control Card
Restart File,restart.1644,
H2O-NaCl-CO2-Energy,
1,
,,2.0,hr,1.e-6,hr,2.0,hr,1.25,24,1.e-06,0.6,
100000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

Extracted from STOMP-WCMSE input file:
~Solution Control Card
Normal,
H2O-CO2-CH4-NaCl-E w/Isobrine,
1,
0,day,5,day,0.1,s,0.1,day,1.25,8,1.e-06,
100000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
Equilibrium Hydrate,
0,

B.34 Source Card

Card Title[a] { ~Source [Card] }
Format: *Char*[a]

Number of Source Domains[a],
Format: Integer[a],

For: Number of Source Domains

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
Source Type Option[a]
{ Aqueous Volumetric [Density] | Aqueous Mass [Density] |
Flow Well[p] [Packer [No Volume]] [No Volume] |
Slug Well[p] [No Volume] | Pulse Well[p] [No Volume] |
Pressure Well[p] [No Volume] |
X-Direction Injection Well | Y-Direction Injection Well |
Z-Direction Injection Well |
Solute [Density] [Inventory] [Advection] [Diffusion] [Solubility], Solute Name }
Note: [Advection] option: Advection-Dominated Solute Release Model
[Diffusion] option: Diffusion-Dominated Solute Release Model
[Solubility] option: Solubility-Controlled Solute Release Model
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char*[a], *Integer*[b], *Integer*[c], *Integer*[d], *Integer*[e], *Integer*[f], *Integer*[g], *Integer*[h],

Elseif: Operational Mode Option = { **STOMP-WA** }
Source Type Option[a]
{ Aqueous Volumetric | Aqueous Mass | Flow Well[p] |
Gas Volumetric [Mass Fraction] [Relative Humidity] |
Gas Mass [Mass Fraction] [Relative Humidity] |
Solute [Density], Solute Name }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char*[a], *Integer*[b], *Integer*[c], *Integer*[d], *Integer*[e], *Integer*[f], *Integer*[g], *Integer*[h]

Elseif: Operational Mode Option = { **STOMP-WAE** } { **STOMP-WAE-B** }
{ **STOMP-WAE-Sc** }
Source Type Option[a]
{ Power [Density] | Aqueous Volumetric | Aqueous Mass | Condensate |
Gas Volumetric [Mass Fraction] [Relative Humidity] |
Gas Mass [Mass Fraction] [Relative Humidity] |
Solute [Density], Solute Name }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char*[a], *Integer*[b], *Integer*[c], *Integer*[d], *Integer*[e], *Integer*[f], *Integer*[g], *Integer*[h],

Source Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WO** }
Source Type Option[a]
{ Aqueous Volumetric | Aqueous Mass |
Flow Well | [[Elevation] [Depth] [Head] Well |
NAPL Volumetric | NAPL Mass |
Solute [Density], Solute Name }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*

Elseif: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc }**
Source Type Option[a]
If: Source Typ Option = { Aqueous Volumetric | Aqueous Mass |
NAPL Volumetric | NAPL Mass |
Gas Volumetric [Mass Fraction] [Relative Humidity] |
Gas Mass [Mass Fraction] [Relative Humidity] |
Solute [Density], Solute Name }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Elseif: Source Type Option = { SVE }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Well Minimum Gas Pressure[i], Units[j], Well Radius[k], Units[l],
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Real[i], Char[j], Real[k], Char[l],
Endif:

Elseif: Operational Mode Option = **{ STOMP-WOAE }**
Source Type Option[a]
{ Power [Density] | Aqueous Volumetric | Aqueous Mass |
NAPL Volumetric | NAPL Mass] |
Gas Volumetric [Mass Fraction] [Relative Humidity] |
Gas Mass [Mass Fraction] [Relative Humidity] |
Solute [Density], Solute Name }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*

Elseif: Operational Mode Option = { **STOMP-WOD** } { **STOMP-WOM** }
Source Type Option[a]
{ Aqueous Volumetric | Aqueous Mass | Dissolved Oil [Density] |
NAPL Volumetric | NAPL Mass |
Solute [Density], Solute Name }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h],

Source Card (cont'd)

Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*

Elseif: Operational Mode Option = { **STOMP-WS** } { **STOMP-WS-Sc** }
Source Type Option[a]
{ Aqueous Volumetric [Density] | Aqueous Mass [Density] |
Salt [Density] | Solute [Density], Solute Name }
If: Source Type Option = { Aqueous Volumetric [Density] }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Elseif: Source Type Option = { Aqueous Mass [Density] }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Endif:

Elseif: Operational Mode Option = { **STOMP-WAS** }
Source Type Option[a]
{ Aqueous Volumetric | Aqueous Mass |
Gas Volumetric | Gas Mass |
Salt [Density] | Solute [Density], Solute Name }
If: Source Type Option = { Aqueous Volumetric }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Elseif: Source Type Option = { Aqueous Mass }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Elseif: Source Type Option = { Gas Volumetric [Mass Fraction] [Relative Humidity] }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Elseif: Source Type Option = { Gas Mass [Mass Fraction] [Relative Humidity] }

Source Card (cont'd)

I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Endif:

Elseif: Operational Mode Option = { **STOMP-WASE }**
Source Type Option[a]
{ Power [Density] | Aqueous Volumetric |
Aqueous Mass | Gas Volumetric [Mass Fraction] [Relative Humidity] |
Gas Mass [Mass Fraction] [Relative Humidity] |
Salt [Density] | Solute [Density], Solute Name }
If: Source Type Option = { Aqueous Volumetric }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],,*
Elseif: Source Type Option = { Aqueous Mass }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Elseif: Source Type Option = { Gas Volumetric [Mass Fraction] [Relative Humidity] }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Elseif: Source Type Option = { Gas Mass [Mass Fraction] [Relative Humidity] }
I-Start Index[b], I-End Index[c], J-Start Index[d], J-End Index[e],
K-Start Index[f], K-End Index[g], Number of Source Times[h] ,
Format: *Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],*
Endif:

Elseif: Operational Mode Option = { **STOMP-WCS }** { **STOMP-WCS-R }**
{ **STOMP-WCS-R-Sc }** { **STOMP-WCS-Sc }**
Source Type Option[a]
{ Aqueous Volumetric | Aqueous Mass |
Gas Volumetric | Gas Mass |
Salt [Density] | Solute [Density], Solute Name }
If: Source Type Option = { Aqueous Volumetric }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
Dissolved CO_2 Source Option[c],

{ Dissolved CO_2 Aqueous Concentration | Dissolved CO_2 Mass Fraction |
Dissolved CO_2 Relative Saturation | Null }
I-Start Index[d], I-End Index[e], J-Start Index[f], J-End Index[g],
K-Start Index[h], K-End Index[i], Number of Source Times[j] ,
Format: *Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j],*
Elseif: Source Type Option = { Aqueous Mass }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
Dissolved CO_2 Source Option[c],
{ Dissolved CO_2 Aqueous Concentration | Dissolved CO_2 Mass Fraction |
Dissolved CO_2 Relative Saturation | Null }
I-Start Index[d], I-End Index[e], J-Start Index[f], J-End Index[g],
K-Start Index[h], K-End Index[i], Number of Source Times[j] ,
Format: *Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j],*
Elseif: Source Type Option = { Gas Volumetric }
Water Vapor Source Option[b],
{ Water Vapor Gas Relative Humidity | Water Vapor Gas Mass Fraction | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Elseif: Source Type Option = { Gas Mass }
Water Vapor Source Option[b],
{ Water Vapor Gas Relative Humidity | Water Vapor Gas Mass Fraction | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*
Endif:

Elseif: Operational Mode Option = { **STOMP-WCSE }**
Source Type Option[a]
{ Power [Density] | Aqueous Volumetric | Aqueous Mass |
Gas Volumetric | Gas Mass |
Salt [Density] | Solute [Density], Solute Name }
If: Source Type Option = { Aqueous Volumetric }
Dissolved Salt Source Option[b],
{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
Dissolved CO_2 Source Option[c],
{ Dissolved CO_2 Aqueous Concentration | Dissolved CO_2Mass Fraction |
Dissolved CO_2 Relative Saturation | Null }
I-Start Index[d], I-End Index[e], J-Start Index[f], J-End Index[g],
K-Start Index[h], K-End Index[i], Number of Source Times[j] ,
Format: *Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j],*
Elseif: Source Type Option = { Aqueous Mass }
Dissolved Salt Source Option[b],

Source Card (cont'd)

{ Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
Dissolved Salt Relative Saturation | Null }
Dissolved CO_2 Source Option[c],
{ Dissolved CO_2 Aqueous Concentration | Dissolved CO_2 Mass Fraction |
Dissolved CO_2 Relative Saturation | Null }
I-Start Index[d], I-End Index[e], J-Start Index[f], J-End Index[g],
K-Start Index[h], K-End Index[i], Number of Source Times[j] ,
Format: *Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j],*

Elseif: Source Type Option = { Gas Volumetric }
Water Vapor Source Option[b],
{ Water Vapor Gas Relative Humidity | Water Vapor Gas Mass Fraction | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*

Elseif: Source Type Option = { Gas Mass }
Water Vapor Source Option[b],
{ Water Vapor Gas Relative Humidity | Water Vapor Gas Mass Fraction | Null }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*

Endif:

Elseif: Operational Mode Option = { **STOMP-WCMSE }**
Source Type Option[a]
{ Power [Density] | [Initial] Fluid-Mass | Solute [Density], Solute Name }
I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
Format: *Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],*

Endif:

Endif:

Source Card (cont'd)

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Aqueous Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Aqueous Mass }
Aqueous Mass Rate[c], Units[d] (kg/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Aqueous Volumetric Density }
Aqueous Volumetric Density Rate[c], Units[d] (m^3/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Aqueous Mass Density }
Aqueous Mass Density Rate[c], Units[d] (kg/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { [X] [Y] [Z] Direction Injection Well }
Well Pressure[c], Units[d], Well Diameter[e], Units[f], Symetry Factor[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Slug Well [No Volume] } or
{ Pressure Well [No Volume] }
Well Bottom Pressure[c], Units[d] (Pa),
Borehole Radius[e], Units[f] (m),
Casing Radius[g], Units[h] (m),
For: Number of Solutes
Solute Aqueous Conc[i], Units[j] (1/m^3),
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],* < *Real[l], Char[j],* >
Elseif: Source Type Option = { Pulse Well [No Volume] }
Well Bottom Pressure[c], Units[d] (Pa),
Borehole Radius[e], Units[f] (m),
Casing Radius[g], Units[h] (m),
System Volume[i], Units[j] (m^3),
System Compressibility[k], Units[l] (1/Pa),
For: Number of Solutes
Solute Aqueous Conc[m], Units[n] (1/m^3),
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l],* < *Real[m], Char[n],* >
Elseif: Source Type Option = { Flow Well [No Volume] }
Aqueous Volumetric Rate[c], Units[d] (m^3/s),
Borehole Radius[e], Units[f] (m),
Casing Radius[g], Units[h] (m),
Well Bottom Pressure Constraint[i], Units[j] (Pa),
For: Number of Solutes
Solute Aqueous Conc[k], Units[l] (1/m^3),
Endfor: Number of Solutes

Source Card (cont'd)

Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
< Real[k], Char[l], >

Elseif: Source Type Option = { Flow Well Packer [No Volume] }
Aqueous Volumetric Rate[c], Units[d] (m^3/s),
Borehole Radius[e], Units[f] (m),
Casing Radius[g], Units[h] (m),
System Volume[i], Units[j] (m^3),
For: Number of Solutes
Solute Aqueous Conc[m], Units[n] (1/m^3),
Endfor: Number of Solutes
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Real[k], Char[l], < Real[m], Char[n], >

Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*

Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/m^3 s),
Format: *Real[a], Char[b], Real[c], Char[d],*

Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WA** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Pressure[c], Units[d] (Pa), Aqueous Volumetric Rate[e], Units[f] (m^3/s),
Dissolved-Air Relative Saturation[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Aqueous Mass }
Pressure[c], Units[d] (Pa), Aqueous Mass Rate[e], Units[f] (kg/s),
Dissolved-Air Relative Saturation[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Gas Mass w/ Mass Fraction }
Pressure[c], Units[d] (Pa), Gas Mass Rate[e], Units[f] (kg/s),
Water Vapor Mass Fraction[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Gas Mass w/ Relative Humidity }
Pressure[c], Units[d] (Pa), Gas Mass Rate[e], Units[f] (kg/s),
Water Vapor Relative Humidity[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Gas Volumetric w/ Mass Fraction }
Pressure[c], Units[d] (Pa), Gas Volumetric Rate[e], Units[f] (m^3/s),
Water Vapor Mass Fraction[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Gas Volumetric w/ Relative Humidity }
Pressure[c], Units[d] (Pa), Gas Volumetric Rate[e], Units[f] (m^3/s),
Water Vapor Relative Humidity[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Flow Well }
Aqueous Volumetric Rate[c], Units[d] (m^3/s), Borehole Radius[e], Units[f] (m),
Water Level Constraint[g], Units[h] (m), Water Level Control Parameter[i], Units[j] (m),
Dissolved-Air Relative Saturation[k],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WAE** } { **STOMP-WAE-B**} { **STOMP-WAE-Sc** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Power }
Power[c], Units[d] (W),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Power Density }
Power Density[c], Units[d] (W/m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Aqueous Volumetric }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s), Dissolved Relative Saturation[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { Condensate }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Dissolved Relative Saturation[g], Fractional Mass Loss Rate[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Source Type Option = { Aqueous Mass }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Mass Rate[g], Units[h] (kg/s), Dissolved Air Relative Saturation[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { Gas Mass w/ Mass Fraction }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Mass Density Rate[g], Units[h] (kg/s), Water Vapor Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i]*
Elseif: Source Type Option = { Gas Mass w/ Relative Humidity }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Mass Density Rate[g], Units[h] (kg/s), Water Vapor Relative Humidity[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i]*
Elseif: Source Type Option = { Gas Volumetric w/ Mass Fraction }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Volumetric Density Rate[g], Units[h] (m^3/s), Water Vapor Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i]*
Elseif: Source Type Option = { Gas Volumetric w/ Relative Humidity }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Volumetric Rate[g], Units[h] (m^3/s), Water Vapor Relative Humidity[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i]*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d]*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d]*
Endif:

Source Card (cont'd)

Endfor: Number of Source Times

Elseif: Operational Mode Option = { **STOMP-WO** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Aqueous Volumetric Rate[c], Units[d] (m^3/s), Dissolved Oil Mass Fraction[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Source Type Option = { Aqueous Mass }
Aqueous Mass Rate[c], Units[d] (kg/s), Dissolved Oil Mass Fraction[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Source Type Option = { NAPL Volumetric }
NAPL Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { NAPL Mass }
NAPL Mass Rate[c], Units[d] (kg/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Flow Well }
Aqueous Volumetric Rate[c], Units[d] (m^3/s), Borehole Radius[e], Units[f] (m),
Liquid Level Constraint[g], Units[h] (m), Dissolved-Oil Relative Saturation[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { [Elevation] [Head] [Depth] Well }
Liquid Elevation in Well[c], Units[d] (m),
Borehole Radius[e], Units[f] (m),
Dissolved-Oil Relative Saturation[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc }**
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type= { Aqueous Volumetric }
Aqueous Volumetric Rate[c], Units[d] (m^3/s),
Dissolved Air Mass Fraction[g], Dissolved Oil Mass Fraction[f],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Aqueous Mass }
Aqueous Mass Rate[c], Units[d] (kg/s),
Dissolved Air Mass Fraction[e], Dissolved Oil Mass Fraction[f],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Gas Mass w/ Mass Fraction }
Gas Mass Rate[c], Units[c] (kg/s),
Water Vapor Mass Fraction[e], Oil Vapor Mass Fraction[f],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Gas Mass w/ Relative Humidity }
Gas Mass Rate[c], Units[d] (kg/s),
Water Vapor Relative Humidity[e], Oil Vapor Relative Humidity[f],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Gas Volumetric w/ Mass Fraction }
Gas Volumetric Rate[c], Units[d] (m^3/s),
Water Vapor Mass Fraction[e], Oil Vapor Mass Fraction[f],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Gas Volumetric w/ Relative Humidity }
Gas Volumetric Rate[c], Units[d] (m^3/s),
Water Vapor Relative Humidity[e], Oil Vapor Relative Humidity[f],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { NAPL Volumetric }
NAPL Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { NAPL Mass }
NAPL Mass Rate[c], Units[d] (kg/s),
NAPL Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s) ,
NAPL Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/m^3 s) ,
NAPL Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WOAE }**
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Power }
Power[c], Units[d] (W),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Power Density }
Power Density[c], Units[d] (W/m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type= { Aqueous Volumetric }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s),
Dissolved Air Mass Fraction[i], Dissolved Oil Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Source Type Option = { Aqueous Mass }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
Aqueous Mass Rate[g], Units[h] (kg/s),
Dissolved Air Mass Fraction[i], Dissolved Oil Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Source Type Option = { Gas Mass w/ Mass Fraction }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
Gas Mass Rate[g], Units[h] (kg/s),
Water Vapor Mass Fraction[i], Oil Vapor Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Source Type Option = { Gas Mass w/ Relative Humidity }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
Gas Mass Rate[g], Units[h] (kg/s),
Water Vapor Relative Humidity[i], Oil Vapor Relative Humidity[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Source Type Option = { Gas Volumetric w/ Mass Fraction }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
Gas Volumetric Rate[g], Units[h] (m^3/s),
Water Vapor Mass Fraction[i], Oil Vapor Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Source Type Option = { Gas Volumetric w/ Relative Humidity }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
Gas Volumetric Rate[g], Units[h] (m^3/s),
Water Vapor Relative Humidity[i], Oil Vapor Relative Humidity[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Source Type Option = { NAPL Volumetric }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa),
NAPL Volumetric Rate[g], Units[h] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Elseif: Source Type Option = { NAPL Mass }
Temperature[c], Units[d], Pressure[e], Units[f] (Pa), NAPL Mass Rate[g], Units[h] (kg/s),

Source Card (cont'd)

Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] ($1/m^3$ s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WOD** } { **STOMP-WOM** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Aqueous Volumetric Rate[c], Units[d] (m^3/s),
Dissolved Oil Conc[e], Units[f] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Aqueous Mass }
Aqueous Mass Rate[c], Units[d] (kg/s),
Dissolved Oil Conc[e], Units[f] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Source Type Option = { Dissolved Oil }
Dissolved Oil Mass Rate[c], Units[d] (kg/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Dissolved Oil Density }
Dissolved Oil Mass Density Rate[c], Units[d] (kg/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { NAPL Volumetric }
NAPL Volumetric Rate[c], Units[d] (m^3/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { NAPL Mass }
NAPL Mass Rate[c], Units[d] (kg/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/m^3 s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Endfor: Number of Source Times

Elseif: Operational Mode Option = { **STOMP-WS** } { **STOMP-WS-Sc** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Aqueous Volumetric Rate[c], Units[d] (m^3/s),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[e], Units[f] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Elseif: Source Type Option = { Aqueous Volumetric Density }
Aqueous Volumetric Density Rate[c], Units[d] (m^3/s m^3),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[e], Units[f] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Elseif: Source Type Option = { Aqueous Mass }
Aqueous Mass Rate[c], Units[d] (kg/s),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[e], Units[f] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Source Card (cont'd)

Elseif: Source Type Option = { Aqueous Mass Density }
Aqueous Mass Density Rate[c], Units[d] (kg/s m^3),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[e], Units[f] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[e],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Elseif: Source Type Option = { Salt }
Salt Mass Rate[c], Units[d] (kg/s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Salt Density }
Salt Mass Density Rate[c], Units[d] (kg/m^3 s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Elseif: Operational Mode Option = { **STOMP-WAS** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Pressure[c], Units[d] (Pa), Aqueous Volumetric Rate[e], Units[f] (m^3/s),
Dissolved-Air Relative Saturation[g],
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[h], Units[i] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],Char[i],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Endif:
Elseif: Source Type Option = { Aqueous Mass }
Pressure[c], Units[d] (Pa), Aqueous Mass Rate[e], Units[f] (kg/s),
Dissolved-Air Relative Saturation[g],
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[h], Units[i] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],Char[i],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Endif:
Elseif: Source Type Option = { Gas Mass w/ Mass Fraction }
Pressure[c], Units[d] (Pa), Gas Mass Rate[e], Units[f] (kg/s),
Water Vapor Mass Fraction[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Gas Mass w/ Relative Humidity }
Pressure[c], Units[d] (Pa), Gas Mass Rate[e], Units[f] (kg/s),
Water Vapor Relative Humidity[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Gas Volumetric w/ Mass Fraction }
Pressure[c], Units[d] (Pa), Gas Volumetric Rate[e], Units[f] (m^3/s),
Water Vapor Mass Fraction[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*

Source Card (cont'd)

Elseif: Source Type Option = { Gas Volumetric w/ Relative Humidity }
Pressure[c], Units[d] (Pa), Gas Volumetric Rate[e], Units[f] (m^3/s),
Water Vapor Relative Humidity[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Source Type Option = { Salt }
Salt Mass Rate[c], Units[d] (kg/s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Salt Density }
Salt Mass Density Rate[c], Units[d] (kg/m^3 s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = { **STOMP-WASE** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Power }
Power[c], Units[d] (W),
Format: *Real[a], Char[b], Real[c], Char[d],Real[e],*
Elseif: Source Type Option = { Power Density }
Power Density[c], Units[d] (W/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e],*
Elseif: Source Type Option = { Aqueous Volumetric }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s), Dissolved Air Mass Fraction[i],
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[j], Units[k] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Endif:
Elseif: Source Type Option = { Aqueous Mass }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Mass Rate[g], Units[h] (kg/s), Dissolved Air Mass Fraction[i],
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[j], Units[k] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k],*
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Endif:
Elseif: Source Type Option = { Gas Mass w/ Mass Fraction }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Mass Density Rate[g], Units[h] (kg/s), Water Vapor Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d],Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { Gas Mass w/ Relative Humidity }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),

Gas Mass Density Rate[g], Units[h] (kg/s), Water Vapor Relative Humidity[i],
Format: *Real[a], Char[b], Real[c], Char[d],Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { Gas Volumetric w/ Mass Fraction }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Volumetric Density Rate[g], Units[h] (m^3/s), Water Vapor Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d],Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { Gas Volumetric w/ Relative Humidity }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Gas Volumetric Rate[g], Units[h] (m^3/s), Water Vapor Relative Humidity[i],
Format: *Real[a], Char[b], Real[c], Char[d],Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Source Type Option = { Salt }
Salt Mass Rate[c], Units[d] (kg/s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Salt Density }
Salt Mass Density Rate[c], Units[d] (kg/m^3 s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Format: *Real[a], Char[b], Real[c], Char[d],Real[e], Char[f], Real[g], Char[h], Real[i],*

Endfor: Number of Source Times

Elseif: Operational Mode Option = { **STOMP-WCS** } { **STOMP-WCS-R** } { **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** }
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Aqueous Volumetric }
Pressure[c], Units[d] (Pa), Aqueous Volumetric Rate[e], Units[f] (m^3/s),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[g], Units[h] (kg/m^3),
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[i], Units[j] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Endif:
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[g],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[h], Units[i] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[g],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[h], Units[i] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Endif:
Elseif: Source Type Option = { Aqueous Mass }
Pressure[c], Units[d] (Pa), Aqueous Mass Rate[e], Units[f] (kg/s),
If: Dissolved Salt Source Option = { Aqueous Concentration }

Dissolved Salt Aqueous Concentration[g], Units[h] (kg/m^3),
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[i], Units[j] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],*
Endif:
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[g],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[h], Units[i] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[g],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[h], Units[i] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[h],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Endif:
Elseif: Source Type Option = { Gas Mass }
Pressure[c], Units[d] (Pa), Gas Mass Rate[e], Units[f] (kg/s),
If: Water Vapor Source Option = { Mass Fraction }
Water Vapor Mass Fraction[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Water Vapor Source Option = { Relative Saturation }
Water Vapor Relative Humidity[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Else:

Source Card (cont'd)

Null[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Endif:
Elseif: Source Type Option = { Gas Volumetric }
Pressure[c], Units[d] (Pa), Gas Volumetric Rate[e], Units[f] (m^3/s),
If: Water Vapor Source Option = { Mass Fraction }
Water Vapor Mass Fraction[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Elseif: Water Vapor Source Option = { Relative Saturation }
Water Vapor Relative Humidity[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Else:
Null[g],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Endif:
Elseif: Source Type Option = { Salt }
Salt Mass Rate[c], Units[d] (kg/s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Salt Density }
Salt Mass Density Rate[c], Units[d] (kg/m^3 s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:

Endfor: Number of Source Times

Source Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WCSE }**
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Power }
Power[c], Units[d] (W),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Power Density }
Power Density[c], Units[d] (W/m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
If: Source Type Option = { Aqueous Volumetric }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[i], Units[j] (kg/m^3),
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[k], Units[l] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[k],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[k],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k],*
Endif:
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[i],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[j], Units[k] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],Real[i], Real[j],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[i],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[j], Units[k] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Char[k],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*

Source Card (cont'd)

Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Else:
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Endif:
Elseif: Source Type Option = { Aqueous Mass }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s),
If: Dissolved Salt Source Option = { Aqueous Concentration }
Dissolved Salt Aqueous Concentration[i], Units[j] (kg/m^3),
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[k], Units[l] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[k],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[k],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k],*
Endif:
Elseif: Dissolved Salt Source Option = { Mass Fraction }
Dissolved Salt Mass Fraction[i],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[j], Units[k] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Real[k],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Dissolved Salt Source Option = { Relative Saturation }
Dissolved Relative Saturation[i],
If: Dissolved CO_2 Source Option = { Aqueous Concentration }
Dissolved CO_2 Aqueous Concentration[j], Units[k] (kg/m^3),
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j], Real[k],*
Elseif: Dissolved CO_2 Source Option = { Mass Fraction }
Dissolved CO_2 Mass Fraction[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Elseif: Dissolved CO_2 Source Option = { Relative Saturation }
Dissolved CO_2 Relative Saturation[j],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],*
Else:

Source Card (cont'd)

Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],*
Endif:
Elseif: Source Type Option = { Gas Mass }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s),
If: Water Vapor Source Option = { Mass Fraction }
Water Vapor Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h], Real[i],*
Elseif: Water Vapor Source Option = { Relative Saturation }
Water Vapor Relative Humidity[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h], Real[i],*
Else:
Null[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h], Real[i],*
Endif:
Elseif: Source Type Option = { Gas Volumetric }
Temperature[c], Units[d] (C), Pressure[e], Units[f] (Pa),
Aqueous Volumetric Rate[g], Units[h] (m^3/s),
If: Water Vapor Source Option = { Mass Fraction }
Water Vapor Mass Fraction[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h], Real[i],*
Elseif: Water Vapor Source Option = { Relative Saturation }
Water Vapor Relative Humidity[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h], Real[i],*
Else:
Null[i],
Format: *Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Reals[g], Char[h], Real[i],*
Endif:
Elseif: Source Type Option = { Salt }
Salt Mass Rate[c], Units[d] (kg/s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Salt Density }
Salt Mass Density Rate[c], Units[d] (kg/m^3 s)
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Endfor: Number of Source Times
Endfor: Number of Source Domains

Source Card (cont'd)

Elseif: Operational Mode Option = **{ STOMP-WCMSE }**
For: Number of Source Times
Source Time[a], Units[b] (s),
If: Source Type Option = { Power }
Power[c], Units[d] (W),
Source Card (cont'd)
Elseif: Source Type Option = { Power Density }
Power Density[c], Units[d] (W/m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Fluid Mass }
Mass Rate[c], Units[d] (W/m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute }
Solute Rate[c], Units[d] (1/s),
Format: *Real[a], Char[b], Real[c], Char[d],*
Elseif: Source Type Option = { Solute Density }
Solute Density Rate[c], Units[d] (1/s m^3),
Format: *Real[a], Char[b], Real[c], Char[d],*
Endif:
Endfor: Number of Source Times
Endfor: Number of Source Domains

Endcard: Source Card

B.34.1 Source Card Examples

Extracted from STOMP-W input file:

```
~Source Card
1,
Aqueous Volumetric,2,2,1,1,71,71,7,
0,min,0.8320722,gal/min,
66.046844,min,1.0441576,gal/min,
135.56322,min,1.0438042,gal/min,
205.09055,min,0.98627841,gal/min,
274.60406,min,0.99331003,gal/min,
344.13425,min,0.91488999,gal/min,
413.64774,min,1.0228366,gal/min,
```

Extracted from STOMP-WA input file:

```
~Source Card
1,
Gas Volumetric w/ Mass Fraction,29,31,1,1,7,9,2,
0,min,101326.2,Pa,0.555556,l/min,,
10,min,101326.2,Pa,5,l/min,,
```

Extracted from STOMP-WAE-B input file:

```
~Source Card
2,
Root-Water Uptake,Sage Brush,1,1,1,1,7,10,1,
0.0,yr,0.1,plant/m^2,
```

Extracted from input file:

```
# ------------------------------------------------------------------------
~Source Card
# ------------------------------------------------------------------------
1,
# well 299-W15-6, lower interval
Well for SVE,75,75,75,75,29,31,23,6.1325e+4,Pa,8,in,
1995.7139,yr,-4.10,m^3/min,153.8,m,147.4,m,
1995.9728,yr,-4.10,m^3/min,153.8,m,147.4,m,
1995.9728,yr,0.0,m^3/min,153.8,m,147.4,m,
1996.0014,yr,0.0,m^3/min,153.8,m,147.4,m,
1996.0014,yr,-3.33,m^3/min,153.8,m,147.4,m,
1996.8428,yr,-3.33,m^3/min,153.8,m,147.4,m,
1996.8428,yr,0.0,m^3/min,153.8,m,147.4,m,
1997.5428,yr,0.0,m^3/min,153.8,m,147.4,m,
1997.5428,yr,-1.73,m^3/min,153.8,m,147.4,m,
1997.7481,yr,-1.73,m^3/min,153.8,m,147.4,m,
1997.7481,yr,0.0,m^3/min,153.8,m,147.4,m,
1998.5722,yr,0.0,m^3/min,153.8,m,147.4,m,
1998.5722,yr,-2.06,m^3/min,153.8,m,147.4,m,
1998.7469,yr,-2.06,m^3/min,153.8,m,147.4,m,
1998.7469,yr,0.0,m^3/min,153.8,m,147.4,m,
1999.2481,yr,0.0,m^3/min,153.8,m,147.4,m,
1999.2481,yr,-2.18,m^3/min,153.8,m,147.4,m,
```

Source Card Examples (cont'd)

1999.4880,yr,-2.18,m^3/min,153.8,m,147.4,m,
1999.4880,yr,0.0,m^3/min,153.8,m,147.4,m,
2002.4654,yr,0.0,m^3/min,153.8,m,147.4,m,
2002.4654,yr,-0.79,m^3/min,153.8,m,147.4,m,
2002.5537,yr,-0.79,m^3/min,153.8,m,147.4,m,
2002.5537,yr,0.0,m^3/min,153.8,m,147.4,m,

Extracted from STOMP-WS input file:
~Source Card
2,
Aqueous Volumetric, Dissolved Salt Aqueous Concentration,30,30,1,1,119,120,2,
0.0,min,1.0,ml/min,312.6,kg/m^3,
5.0,min,1.0,ml/min,312.6,kg/m^3,
Aqueous Volumetric,,60,60,1,1,119,120,2,
0.0,min,1.0,ml/min,
5.0,min,1.0,ml/min,

Extracted from STOMP-WCS input file:
~Source Card
1,
Gas Mass Rate,Water-Vapor Mass Fraction,1,1,1,1,1,9,2,
0,day,,,0.352413,kg/s,0.0,
7300,day,,,0.352413,kg/s,0.0,

Extracted from STOMP-WCSE input file:
~Source Card
1,
Gas Volumetric Rate,Water-Vapor Mass Fraction,3,3,1,1,3,3,2,
0,hr,25.0,C,138.0,bar,4.2908,cm^3/min,0.0,
0.2,hr,25.0,C,138.0,bar,4.2908,cm^3/min,0.0,

B.35 Species Link Card

Card Title[a] { ~Species Link [Card] }
Format: *Char*[a],

Number of Species Links[a],
Format: *Integer*[a],

For: Number of Species Links
  Species Name[a],
  Coupled Flow Component Option[b],
  { Aqueous pH | Aqueous Water | Gas Water | Aqueous Air | Gas Air |
  Aqueous CO2 | Gas CO2 | Aqueous CH4 | Gas CH4 }
  Format: *Char*[a], *Char*[b],
Endfor: Number of Species Links

EndCard: Species Link

B.35.1 Species Link Card Examples

Extracted from a STOMP-WCS-R input file:

```
~Species Link Card
7,
H+,aqueous pH,
CaCO3(aq),aqueous CO2,
HCO3-,aqueous CO2,
CO3--,aqueous CO2,
CaHCO3+,aqueous CO2,
H2CO3,aqueous CO2,
CO2(g),gas CO2,
```

B.36 Surface Flux Card

Card Title[a] { ~Surface [Flux Card] }
Format: *Char*[a]

Number of Surface Flux Inputs[a]
Format: Integer[a],

Note: The number of surface flux inputs may be written to one or more user specified output files. For each user specified file, the following input line has to be specified. An example has been included in the Surface Flux Card Examples section (first example). The sum of the "Number of Surface Flux Inputs in File" has to be equal to the "Number of Surface Flux Inputs".

Number of Surface Flux Inputs in File[a], Filename[b],
Format: Integer[a], Char[b],

For: Number of Surface Flux Inputs:

If: Operational Mode Option = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WA** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WAE** } { **STOMP-WAE-Sc** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Condensate Water Mass | |
Gas Volumetric | Gas Advective Heat | Gas Advective Water Mass |
Gas Advective Air Mass | Gas Diffusive Heat |
Gas Diffusive Water Mass | Gas Diffusive Air Mass | Heat |
Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WAE-B** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Advective Heat | Gas Advective Water Mass |
Gas Advective Air Mass | Gas Diffusive Heat |
Gas Diffusive Water Mass | Gas Diffusive Air Mass |
Solute, Solute Name | Heat | Condensate Water Mass |
Actual Evaporation | Potential Evaporation |
Actual Transpiration | Potential Evaporation |
Net Long-Wave Radiation | Net Short-Wave Radiation |
Net Total Radiation | Water-Mass Balance }

Surface Flux Card (cont'd)

Elseif: Operation Mode Option = { **STOMP-WO** } { **STOMP-WOD** } { **STOMP-WOM** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | NAPL Volumetric |
NAPL Mass | Dissolved Oil | Water Mass | Solute Flux, Solute Name }

Elseif: Operation Mode Option = **{ STOMP-WOA } { STOMP-WOA-Sc }**
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | NAPL Volumetric | NAPL Mass |
Gas-Advective Oil Mass | Gas-Diffusive Oil Mass | Gas-Total Oil Mass |
Dissolved-Oil | Water Mass | Solute, Solute Name }

Elseif: Operation Mode Option = **{ STOMP-WOAE }**
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | NAPL Volumetric | NAPL Mass |
Heat Flux | Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WS** } { **STOMP-WS-Sc** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Salt Mass | Solute Flux, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WAS** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | Salt Mass | Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WASE** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | Salt Mass | Heat | Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WCS** } { **STOMP-WCS-R** }
{ **STOMP-WCS-R-Sc** } { **STOMP-WCS-Sc** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | Salt Mass | CO_2-Mass |
Aqueous-Phase CO_2 Mass | Gas-Phase CO_2 Mass |
Solute, Solute Name }

Elseif: Operation Mode Option = { **STOMP-WCSE** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | Salt Mass | CO_2-Mass |
Aqueous-Phase CO_2 Mass | Gas-Phase CO_2 Mass |
Heat | Solute, Solute Name }

Surface Flux Card (cont'd)

Elseif: Operation Mode Option = { **STOMP-WCMSE** }
Surface Flux Type Option[a],
{ Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
Gas Mass | Salt Mass |
Aqueous-Phase CO_2 Mass | Gas-Phase CO_2 Mass |
Aqueous-Phase CH_4 Mass | Gas-Phase CH_4 Mass |
CO_2-Mass | CH_4-Mass | Heat | Solute, Solute Name }

Endif:

If: Surface Flux Type Option = { Heat } { Radiation }
Units[b] (W), Units[c] (J),
ElseIf: Surface Flux Type Option = { Volumetric }
Units[b] (m^3/s), Units[c] (m^3),
Elseif: Surface Flux Type Option = { Mass } { Dissolved Oil }
Units[b] (kg/s), Units[c] (kg),
Elseif: Surface Flux Type Option = { Solute }
Units[b] (sol/s), Units[c] (sol),
Endif:

Surface Flux Orientation Option[d]
{ West } { East } { South } { North } { Top } { Bottom }
I-Start Index[e], I-End Index[f],
J-Start Index[g], J-End Index[h],
K-Start Index[i], K-End Index[j],
Format: *Char[a], Char[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j],*

Endfor: Number of Surface Flux Inputs

Endcard: Surface Flux Card

B.36.1 Surface Flux Card Examples

Extracted from STOMP-W input file:
~Surface Flux Card
13,
8,bottom_flux_z16m.srf,
Aqueous Volumetric Flux,L/day,L,bottom,1,180,1,1,17,17,
Solute Flux,U:0.01,1/day,, bottom,1,180,1,1,17,17,
Solute Flux,U:0.03,1/day,, bottom,1,180,1,1,17,17,
Solute Flux,U:0.10, 1/day,, bottom,1,180,1,1,17,17,
Solute Flux,U:0.30,1/day,, bottom,1,180,1,1,17,17,
Solute Flux,U:0.60,1/day,, bottom,1,180,1,1,17,17,
Solute Flux,U:1.00, 1/day,, bottom,1,180,1,1,17,17,
Solute Flux,Tc,1/day,, bottom,1,180,1,1,17,17,
5,east_flux_x61m.srf,
Aqueous Volumetric Flux,L/day,L,East, 61,61,1,1,1,18,
Solute Flux,U:0.01,1/day,, East, 61,61,1,1,1,18,
Solute Flux,U:0.03,1/day,, East, 61,61,1,1,1,18,
Solute Flux,U:0.10,1/day,, East, 61,61,1,1,1,18,
Solute Flux,U:0.30,1/day,, East, 61,61,1,1,1,18,

Extracted from STOMP-W input file:
~Surface Flux Card
5,
Aqueous Volumetric,gal/min,gal,East,2,2,1,1,14,33,
Aqueous Volumetric,gal/min,gal,East,2,2,1,1,71,100,
Aqueous Volumetric,gal/min,gal,East,52,52,1,1,1,113,
Solute Flux,TCE,1/min,,East,2,2,1,1,14,33,
Solute Flux,TCE,1/min,,East,2,2,1,1,71,100,

Extracted from STOMP-W input file:
~Surface Flux Card
4,
Solute Flux,Tc-99,1/yr,,Bottom,1,10,1,1,1,1,
Solute Flux,U-238,1/yr,,Bottom,1,10,1,1,1,1,
Solute Flux,Np-237,1/yr,,Bottom,1,10,1,1,1,1,
Solute Flux,Pu-239,1/yr,,Bottom,1,10,1,1,1,1,

Extracted from STOMP-WO input file:
~Surface Flux Card
1,
NAPL Volumetric Flux,cm^3/min,cm^3,Top,1,1,1,1,10,10,

Extracted from STOMP-WOA input file:

~Surface Flux Card

25,
bottom of trench
NAPL Volumetric Flux,m^3/yr,m^3,top,13,15,13,17,79,79,
top of pplz

Surface Flux Card Examples (cont'd)

NAPL Volumetric Flux,m^3/yr,m^3,top,1,27,1,34,60,60,
bottom of pplc
NAPL Volumetric Flux,m^3/yr,m^3,top,1,27,1,34,38,38,
plane through minimum elevation of water table
NAPL Volumetric Flux,m^3/yr,m^3,top,1,27,1,34,27,27,
South boundary of model domain
NAPL Volumetric Flux,m^3/yr,m^3,south,1,27,1,1,1,85,
North boundary of model domain
NAPL Volumetric Flux,m^3/yr,m^3,north,1,27,34,34,1,85,
West boundary of model domain
NAPL Volumetric Flux,m^3/yr,m^3,west,1,1,1,34,1,85,
East boundary of model domain
NAPL Volumetric Flux,m^3/yr,m^3,east,27,27,1,34,1,85,
Gas-total oil mass flux,kg/yr,kg,top,1,27,1,34,85,85,
Gas-total oil mass flux,kg/yr,kg,south,1,27,1,1,61,85,
Gas-total oil mass flux,kg/yr,kg,south,1,27,1,1,39,60,
Gas-total oil mass flux,kg/yr,kg,south,1,27,1,1,1,38,
Gas-total oil mass flux,kg/yr,kg,north,1,27,34,34,61,85,
Gas-total oil mass flux,kg/yr,kg,north,1,27,34,34,39,60,
Gas-total oil mass flux,kg/yr,kg,north,1,27,34,34,1,38,
Gas-total oil mass flux,kg/yr,kg,west,1,1,1,34,61,85,
Gas-total oil mass flux,kg/yr,kg,west,1,1,1,34,39,60,
Gas-total oil mass flux,kg/yr,kg,west,1,1,1,34,1,38,
Gas-total oil mass flux,kg/yr,kg,east,27,27,1,34,61,85,
Gas-total oil mass flux,kg/yr,kg,east,27,27,1,34,39,60,
Gas-total oil mass flux,kg/yr,kg,east,27,27,1,34,1,38,
Dissolved oil mass flux,kg/yr,kg,top,1,27,1,34,85,85,
Dissolved oil mass flux,kg/yr,kg,south,1,27,1,1,1,85,
Dissolved oil mass flux,kg/yr,kg,north,1,27,34,34,1,85,
Dissolved oil mass flux,kg/yr,kg,west,1,1,1,34,1,85,
Dissolved oil mass flux,kg/yr,kg,east,27,27,1,34,1,85,

Extracted from STOMP-WAE input file:

```
#-------------------------------------------------------
~Surface Flux Card
#-------------------------------------------------------
10,
Aqueous Volumetric,cm^3/hr,cm^3,Top,   1,1,1,1,113,113, #0.0 cm
Aqueous Volumetric,cm^3/hr,cm^3,Top,   1,1,1,1,112,112, #0.2 cm
Aqueous Volumetric,cm^3/hr,cm^3,Top,   1,1,1,1,50,50,   #1 m
Aqueous Volumetric,cm^3/hr,cm^3,Top,   1,1,1,1,19,19,   #2 m
Aqueous Volumetric,cm^3/hr,cm^3,TOP,   1,1,1,1, 1, 1,   #3 m
Heat Flux,        W,J,Top,   1,1,1,1,113,113, #0.0 cm
Heat Flux,        W,J,Top,   1,1,1,1,112,112, #0.2 cm
Heat Flux,        W,J,Top,   1,1,1,1,50,50,   #1 m
Heat Flux,        W,J,Top,   1,1,1,1,19,19,   #2 m
```

Surface Flux Card Examples (cont'd)

Extracted from STOMP-WAE-B input file:

```
#-------------------------------------------------------
~Surface Flux Card
#-------------------------------------------------------
8,
Potential Evaporation,kg/day,kg,Top,1,1,1,1,145,145,
Actual Evaporation,kg/day,kg,Top,1,1,1,1,145,145,
Potential Transpiration,kg/day,kg,Top,1,1,1,1,145,145,
Actual Transpiration,kg/day,kg,Top,1,1,1,1,145,145,
Aqueous Volumetric,cm^3/day,cm^3,Bottom,1,1,1,1,131,131,  # 35.0 cm Sandberg's bluegrass
Aqueous Volumetric,cm^3/day,cm^3,Bottom,1,1,1,1,125,125,  # 50.0 cm Cheatgrass
Aqueous Volumetric,cm^3/day,cm^3,Bottom,1,1,1,1,  6, 6,    # 350 cm
Aqueous Volumetric,cm^3/day,cm^3,Bottom,1,1,1,1,  2, 2,    # Water table
```

B.37 Thermal Properties Card

Card Title[a] { ~Thermal [Properties Card] }
Format: *Char[a]*

If: Operational Mode = **{ STOMP-WAE-B }** -with Water-Vapor Enhanced Diffusion,
-without Ground-Surface Albedo

For: Number of Rock/Soil Types
Rock/Soil Name[a],
Thermal Conductivity Function Option[b],
{ Constant } { Parallel } { Linear } { Somerton } { Campbell } { Cass }

If: Thermal Conductivity Function Option = { Constant }
X-Dir. Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Thermal Conductivity[g], Units[h] (W/m K), Specific Heat[i], Units[j] (J/kg K),
Water-Vapor Diffusion Option[k], { Enhanced },
Parameter[l], Parameter[m], Parameter[n], Parameter[o], Parameter[p],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Realo, Real[p],*

Elseif: Thermal Conductivity Function Option = { Parallel }
X-Dir. Rock/Soil Grain Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Grain Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Grain Thermal Conductivity[g], Units[h] (W/m K),
Specific Heat[i], Units[j] (J/kg K),
Water-Vapor Diffusion Option[k], { Enhanced },
Parameter[l], Parameter[m], Parameter[n], Parameter[o], Parameter[p],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p],*

Elseif: Thermal Conductivity Function Option = { Linear }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Water-Vapor Diffusion Option[q], { Enhanced },
Parameter[r], Parameter[s], Parameter[t], Parameter[u], Parameter[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v],*

Elseif: Thermal Conductivity Function Option = { Somerton }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),

Thermal Properties Card (cont'd)

Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Water-Vapor Diffusion Option[q], { Enhanced },
Parameter[r], Parameter[s], Parameter[t], Parameter[u], Parameter[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v],*

Elseif: Thermal Conductivity Function Option = { Campbell }
Parameter a[c] (0.734), Units[d] (W/m K), Parameter b[e] (1.45), Units[f] (W/m K),
Parameter c[g] (2.01), Parameter d[h] (0.204), Units[i] (W/m K),
Parameter e[j] (4.0), Specific Heat[k], Units[l] (J/kg K),
Water-Vapor Diffusion Option[m], { Enhanced },
Parameter[n], Parameter[o], Parameter[p], Parameter[q], Parameter[r],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Chai[j],Real[j], Real[k], Char[l], Char[m], Real[m], Real[o], Real[p], Real[q], Real[r],*

Elseif: Thermal Conductivity Function Option = { Cass }
Parameter a[c], Units[d] (W/m K), Parameter b[e], Units[f] (W/m K),
Parameter c[g], Parameter d[h], Units[i] (W/m K),
Parameter e[j], Specific Heat[k], Units[l] (J/kg K),
Water-Vapor Diffusion Option[m], { Enhanced },
Parameter[n], Parameter[o], Parameter[p], Parameter[q], Parameter[r],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Chai[j],Real[j], Real[k], Char[l], Char[m], Real[m], Real[o], Real[p], Real[q], Real[r],*

Endif:

Endfor: Number of Rock/Soil Types

Endif:

Thermal Properties Card (cont'd)

If: Operational Mode = **{ STOMP-WAE-B }** -without Water-Vapor Enhanced Diffusion,
-with Ground-Surface Albedo

For: Number of Rock/Soil Types
Rock/Soil Name[a],
Thermal Conductivity Function Option[b],
{ Constant } { Parallel } { Linear } { Somerton } { Campbell } { Cass }

If: Thermal Conductivity Function Option = { Constant }
X-Dir. Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Thermal Conductivity[g], Units[h] (W/m K), Specific Heat[i], Units[j] (J/kg K),
Ground-Surface Albedo Option[k],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo}
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[l], Wet-Soil Albedo[m], Albedo Attenuation Factor[n],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n],*
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[l], Wet-Soil Albedo[m], Albedo Attenuation Factor[n],
Reference Albedo@Solar Zenith = 60[o] deg[o],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o],*
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[l], Wet-Soil Albedo[m], Albedo Attenuation Factor[n],
Reference Albedo@Solar Zenith = 60[o] deg[o], Parameter C[p],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p],*
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[l] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l],*
Endif:

Elseif: Thermal Conductivity Function Option = { Parallel }
X-Dir. Rock/Soil Grain Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Grain Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Grain Thermal Conductivity[g], Units[h] (W/m K),
Specific Heat[i], Units[j] (J/kg K),
Ground-Surface Albedo Option[k],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[l], Wet-Soil Albedo[m], Albedo Attenuation Factor[n],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n],*

Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[l], Wet-Soil Albedo[m], Albedo Attenuation Factor[n],
Reference Albedo@Solar Zenith = 60[o] deg[o],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o],*
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[l], Wet-Soil Albedo[m], Albedo Attenuation Factor[n],
Reference Albedo@Solar Zenith = 60[o] deg[o], Parameter C[p],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p],*
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[l] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l],*
Endif:
Elseif: Thermal Conductivity Function Option = { Linear }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Ground-Surface Albedo Option[q],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r], Real[s], Real[t],*
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r], Real[s], Real[t], Real[u],*
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u], Parameter C[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v],*
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[r] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r],*
Endif:

Elseif: Thermal Conductivity Function Option = { Somerton }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Ground-Surface Albedo Option[q],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r], Real[s], Real[t],
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q],
Real[r], Real[s], Real[t], Real[u],
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u], Parameter C[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q],
Real[r], Real[s], Real[t], Real[u], Real[v],
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[r] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Char[q], Real[r],
Endif:

Elseif: Thermal Conductivity Function Option = { Campbell }
Parameter a[c] (0.734), Units[d] (W/m K), Parameter b[e] (1.45), Units[f] (W/m K),
Parameter c[g] (2.01), Parameter d[h] (0.204), Units[i] (W/m K),
Parameter e[j] (4.0), Specific Heat[k], Units[l] (J/kg K),
Ground-Surface Albedo Option[m],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[n], Wet-Soil Albedo[o], Albedo Attenuation Factor[p],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p],
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[n], Wet-Soil Albedo[o], Albedo Attenuation Factor[p],
Reference Albedo@Solar Zenith = 60[o] deg[q],

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*

Thermal Properties Card (cont'd)

Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[n], Wet-Soil Albedo[o], Albedo Attenuation Factor[p],
Reference Albedo@Solar Zenith = 60[o] deg[q], Parameter C[r],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q], Real[r],
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[n] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Real[m], Real[n],
Endif:

Elseif: Thermal Conductivity Function Option = { Cass }
Parameter a[c], Units[d] (W/m K), Parameter b[e], Units[f] (W/m K),
Parameter c[g], Parameter d[h], Units[i] (W/m K),
Parameter e[j], Specific Heat[k], Units[l] (J/kg K),
Ground-Surface Albedo Option[m],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[n], Wet-Soil Albedo[o], Albedo Attenuation Factor[p],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p],
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[n], Wet-Soil Albedo[o], Albedo Attenuation Factor[p],
Reference Albedo@Solar Zenith = 60[o] deg[q],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[n], Wet-Soil Albedo[o], Albedo Attenuation Factor[p],
Reference Albedo@Solar Zenith = 60[o] deg[q], Parameter C[r],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q], Real[r],
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[n] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Real[m], Real[n],
Endif:

Endif:

Endfor: Number of Rock/Soil Types

Endif:

Thermal Properties Card (cont'd)

If: Operational Mode = **{ STOMP-WAE-B } -**with Water-Vapor Enhanced Diffusion,
-with Ground-Surface Albedo

For: Number of Rock/Soil Types
Rock/Soil Name[a],
Thermal Conductivity Function Option[b],
{ Constant } { Parallel } { Linear } { Somerton } { Campbell } { Cass }

If: Thermal Conductivity Function Option = { Constant }
X-Dir. Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Thermal Conductivity[g], Units[h] (W/m K), Specific Heat[i], Units[j] (J/kg K),
Water-Vapor Diffusion Option[k], { Enhanced },
Parameter[l], Parameter[m], Parameter[n], Parameter[o], Parameter[p],
Ground-Surface Albedo Option[q],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo}
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t],*
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u],*
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u], Parameter C[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v],*
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[r] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r],*
Endif:

Elseif: Thermal Conductivity Function Option = { Parallel }
X-Dir. Rock/Soil Grain Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Grain Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Grain Thermal Conductivity[g], Units[h] (W/m K),
Specific Heat[i], Units[j] (J/kg K),
Water-Vapor Diffusion Option[k], { Enhanced },
Parameter[l], Parameter[m], Parameter[n], Parameter[o], Parameter[p],
Ground-Surface Albedo Option[q],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo}
If: Ground-Surface Albedo Option = { Pleim and Xiu }

Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t],*

Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u],*

Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[r], Wet-Soil Albedo[s], Albedo Attenuation Factor[t],
Reference Albedo@Solar Zenith = 60[o] deg[u], Parameter C[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v],*

Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[r] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r],*

Endif:

Elseif: Thermal Conductivity Function Option = { Linear }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Water-Vapor Diffusion Option[q], { Enhanced },
Parameter[r], Parameter[s], Parameter[t], Parameter[u], Parameter[v],
Ground-Surface Albedo Option[x],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo}
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[y], Wet-Soil Albedo[z], Albedo Attenuation Factor[aa],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y], Real[z], Real[aa],*

Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[y], Wet-Soil Albedo[z], Albedo Attenuation Factor[aa],
Reference Albedo@Solar Zenith = 60[o] deg[bb],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y], Real[z], Real[aa], Real[bb],*

Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[y], Wet-Soil Albedo[z], Albedo Attenuation Factor[aa],
Reference Albedo@Solar Zenith = 60[o] deg[bb], Parameter C[cc],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*

Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y], Real[z], Real[aa], Real[bb], Real[cc],

Elseif: Ground-Surface Albedo Option = { Constant }

Mean Soil Albedo[y] (0.20),

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y],*

Endif:

Elseif: Thermal Conductivity Function Option = { Somerton }

X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Water-Vapor Diffusion Option[q], { Enhanced },
Parameter[r], Parameter[s], Parameter[t], Parameter[u], Parameter[v],
Ground-Surface Albedo Option[x],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo}

If: Ground-Surface Albedo Option = { Pleim and Xiu }

Dry-Soil Albedo[y], Wet-Soil Albedo[z], Albedo Attenuation Factor[aa],

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y], Real[z], Real[aa],*

Elseif: Ground-Surface Albedo Option = { Wang }

Dry-Soil Albedo[y], Wet-Soil Albedo[z], Albedo Attenuation Factor[aa],
Reference Albedo@Solar Zenith = 60[o] deg[bb],

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y], Real[z], Real[aa], Real[bb],*

Elseif: Ground-Surface Albedo Option = { Briegleb }

Dry-Soil Albedo[y], Wet-Soil Albedo[z], Albedo Attenuation Factor[aa],
Reference Albedo@Solar Zenith = 60[o] deg[bb], Parameter C[cc],

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y], Real[z], Real[aa], Real[bb], Real[cc],*

Elseif: Ground-Surface Albedo Option = { Constant }

Mean Soil Albedo[y] (0.20),

Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Char[k], Real[l], Real[m], Real[n], Real[o], Real[p], Char[q], Real[r], Real[s], Real[t], Real[u], Real[v], Char[s], Real[y],*

Endif:

Elseif: Thermal Conductivity Function Option = { Campbell }
Parameter a[c] (0.734), Units[d] (W/m K), Parameter b[e] (1.45), Units[f] (W/m K),
Parameter c[g] (2.01), Parameter d[h] (0.204), Units[i] (W/m K),
Parameter e[j] (4.0), Specific Heat[k], Units[l] (J/kg K),
Water-Vapor Diffusion Option[m], { Enhanced },
Parameter[n], Parameter[o], Parameter[p], Parameter[q], Parameter[r],
Ground-Surface Albedo Option[s],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[t], Wet-Soil Albedo[u], Albedo Attenuation Factor[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t], Real[u], Real[v],
Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[t], Wet-Soil Albedo[u], Albedo Attenuation Factor[v],
Reference Albedo@Solar Zenith = 60[o] deg[x],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t], Real[u], Real[v],Real[x],
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[t], Wet-Soil Albedo[u], Albedo Attenuation Factor[v],
Reference Albedo@Solar Zenith = 60[o] deg[x], Parameter C[y],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t], Real[u], Real[v],Real[x],Real[y],
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[t] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t],
Endif:

Elseif: Thermal Conductivity Function Option = { Cass }
Parameter a[c], Units[d] (W/m K), Parameter b[e], Units[f] (W/m K),
Parameter c[g], Parameter d[h], Units[i] (W/m K),
Parameter e[j], Specific Heat[k], Units[l] (J/kg K),
Water-Vapor Diffusion Option[m], { Enhanced },
Parameter[n], Parameter[o], Parameter[p], Parameter[q], Parameter[r],
Ground-Surface Albedo Option[s],
{ Pleim and Xiu Albedo | Wang Albedo | Briegleb Albedo | Constant Albedo }
If: Ground-Surface Albedo Option = { Pleim and Xiu }
Dry-Soil Albedo[t], Wet-Soil Albedo[u], Albedo Attenuation Factor[v],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t], Real[u], Real[v],

Thermal Properties Card (cont'd)

Elseif: Ground-Surface Albedo Option = { Wang }
Dry-Soil Albedo[t], Wet-Soil Albedo[u], Albedo Attenuation Factor[v],
Reference Albedo@Solar Zenith = 60[o] deg[x],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t], Real[u], Real[v],Real[x],
Elseif: Ground-Surface Albedo Option = { Briegleb }
Dry-Soil Albedo[t], Wet-Soil Albedo[u], Albedo Attenuation Factor[v],
Reference Albedo@Solar Zenith = 60[o] deg[x], Parameter C[y],
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t], Real[u], Real[v],Real[x],Real[y],
Elseif: Ground-Surface Albedo Option = { Constant }
Mean Soil Albedo[t] (0.20),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],*
Char[i], Real[j], Real[k], Char[l], Char[m], Real[n], Real[o], Real[p], Real[q],
Real[r], Char[s], Real[t],
Endif:

Endif:
Endfor: Number of Rock/Soil Types
Endif:

Thermal Properties Card (cont'd)

If: Operational Mode Option = **{ STOMP-WOAE }**
For: Number of Rock/Soil Types
Rock/Soil Name[a],
Thermal Conductivity Function Option[b],
{ Constant } { Parallel } { Linear } { Somerton } { Campbell } { Cass }

If: Thermal Conductivity Function Option = { Constant }
X-Dir. Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Thermal Conductivity[g], Units[h] (W/m K), Specific Heat[i], Units[j] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Thermal Conductivity Function Option = { Parallel }
X-Dir. Rock/Soil Grain Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Grain Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Grain Thermal Conductivity[g], Units[h] (W/m K),
Specific Heat[h], Units[i] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Thermal Conductivity Function Option = { Linear }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
X-Dir. Rock/Soil NAPL Saturated Thermal Conductivity[o], Units[p] (W/m K),
Y-Dir. Rock/Soil NAPL Saturated Thermal Conductivity[q], Units[r] (W/m K),
Z-Dir. Rock/Soil NAPL Saturated Thermal Conductivity[s], Units[t] (W/m K),
Specific Heat[u], Units[v] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Real[q], Char[r],
Real[s], Char[t], Real[u], Char[v],
Elseif: Thermal Conductivity Function Option = { Somerton }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
X-Dir. Rock/Soil NAPL Saturated Thermal Conductivity[o], Units[p] (W/m K),
Y-Dir. Rock/Soil NAPL Saturated Thermal Conductivity[q], Units[r] (W/m K),
Z-Dir. Rock/Soil NAPL Saturated Thermal Conductivity[s], Units[t] (W/m K),
Specific Heat[u], Units[v] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],*
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p], Real[q], Char[r],
Real[s], Char[t], Real[u], Char[v],

Thermal Properties Card (cont'd)

Elseif: Thermal Conductivity Function Option = { Campbell }
Parameter a[c] (0.734), Units[d] (W/m K), Parameter b[e] (1.45), Units[f] (W/m K),
Parameter c[g] (2.01), Parameter d[h] (0.204), Units[i] (W/m K),
Parameter e[j] (4.0), Specific Heat[k], Units[l] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Real[h], Char[i], Real[j], Real[k], Char[l],
Elseif: Thermal Conductivity Function Option = { Cass }
Parameter a[c], Units[d] (W/m K), Parameter b[e] , Units[f] (W/m K),
Parameter c[g], Parameter d[h], Units[i] (W/m K),
Parameter e[j], Specific Heat[k], Units[l] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Real[h], Char[i], Real[j], Real[k], Char[l],
Endif:
Endfor: Number of Rock/Soil Types
Endif:

Thermal Properties Card (cont'd)

If: Operational Mode Option = **{ STOMP-WASE } { STOMP-WCSE } { STOMP-WCMSE }**
For: Number of Rock/Soil Types
Rock/Soil Name[a],
Thermal Conductivity Function Option[b],
{ Constant } { Parallel } { Linear } { Somerton } { Campbell } { Cass }

If: Thermal Conductivity Function Option = { Constant }
X-Dir. Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Thermal Conductivity[g], Units[h] (W/m K), Specific Heat[i], Units[j] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Thermal Conductivity Function Option = { Parallel }
X-Dir. Rock/Soil Grain Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Grain Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Grain Thermal Conductivity[g], Units[h] (W/m K),
Specific Heat[h], Units[i] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],*
Elseif: Thermal Conductivity Function Option = { Linear }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p],*
Elseif: Thermal Conductivity Function Option = { Somerton }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p],*
Elseif: Thermal Conductivity Function Option = { Campbell }
Parameter a[c] (0.734), Units[d] (W/m K), Parameter b[e] (1.45), Units[f] (W/m K),
Parameter c[g] (2.01), Parameter d[h] (0.204), Units[i] (W/m K),
Parameter e[j] (4.0), Specific Heat[k], Units[l] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i], Real[j], Real[k], Char[l],*
Elseif: Thermal Conductivity Function Option = { Cass }
Parameter a[c], Units[d] (W/m K), Parameter b[e] , Units[f] (W/m K),
Parameter c[g], Parameter d[h], Units[i] (W/m K),

Thermal Properties Card (cont'd)

Parameter e[j], Specific Heat[k], Units[l] (J/kg K),
Format: *Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],*
Real[h], Char[i], Real[j], Real[k], Char[l],
Endif:
Endfor: Number of Rock/Soil Types
Endif:

Endcard: Thermal Properties Card

B.37.1 Thermal Properties Card Examples

Extracted from a STOMP-WAE input file:
~Thermal Properties Card
Soil,Constant,0.5,W/m K,0.5,W/m K,0.5,W/m K,750,J/kg K

Extracted from a STOMP-WAE input file:
#---
~Thermal Properties Card
#---
COMPGRAV,Cass Conductivity Model,0.6,W/m K,0.8,W/m K,4.5,0.22,W/m K,6.0,901.88,J/kg K,
COMPOS1,Cass Conductivity Model,0.6,W/m K,0.8,W/m K,4.5,0.22,W/m K,6.0,901.88,J/kg K,
GRAVEL1,Cass Conductivity Model,0.6,W/m K,0.7,W/m K,8.0,0.26,W/m K,3.0,513.21,J/kg K,

Extracted from a STOMP-WAE input file:
~Thermal Properties Card
Silt Loam-Gravel Admix,Somerton,0.25,W/m K,0.25,W/m K,0.25,W/m K,2.0,W/m K,2.0,W/m K,2.0,W/m K,750,J/kg K,
Gravel Drainage,Somerton,0.25,W/m K,0.25,W/m K,0.25,W/m K,2.0,W/m K,2.0,W/m K,2.0,W/m K,750,J/kg K,
Gravel Filter,Somerton,0.25,W/m K,0.25,W/m K,0.25,W/m K,2.0,W/m K,2.0,W/m K,2.0,W/m K,750,J/kg K,
Riprap,Somerton,0.25,W/m K,0.25,W/m K,0.25,W/m K,2.0,W/m K,2.0,W/m K,2.0,W/m K,750,J/kg K,
Compacted Silt Loam,Somerton,0.25,W/m K,0.25,W/m K,0.25,W/m K,2.0,W/m K,2.0,W/m K,2.0,W/m K,750,J/kg K,

Extracted from a STOMP-WAE-B input file:
#---
~Thermal Properties Card
#---
Silt loam/gravel,Somerton,0.0,W/m K,0.0,W/m K,0.272,W/m K,0.0,W/m K,0.0,W/m K,1.752,W/m K,730,J/kg K,albedo,Wang,0.25,0.1,3.5058,0.04,
silt loam, Somerton,0.0,W/m K,0.0,W/m K,0.188,W/m K,0.0,W/m K,0.0,W/m K,1.374,W/m K,730,J/kg K,albedo,Wang,0.25,0.1,3.5058,0.04,

Extracted from a STOMP-WAE-B input file:
#---
~Thermal Properties Card
#---
silt loam/gravel,Cass,1.16,W/m K,0.922,W/m K,3.468,0.207,W/m K,2.177,712.250,J/kg K,enhanced,9.5,2.0,8.0,0.50,3.0,albedo,Wang,0.25,0.1,3.5058,0.04,
silt loam,Cass,1.13,W/m K,0.943,W/m K,3.477,0.188,W/m K,2.325,793.1,J/kg K,enhanced,9.5,2.0,8.0,0.50,3.0,albedo,Wang,0.25,0.1,3.5058,0.04,

B.38 UCODE Control Card

Card Title[a] { ~Ucode [Control Card] }
Format: *Char*[a]

If: Operational Mode Option Card = { **STOMP-W** } { **STOMP-W-R** } { **STOMP-W-Sc** } { **STOMP-WAE** } { **STOMP-W-B** } { **STOMP-W-Sc** }

If: Execution Mode Option = { Normal w/ Inverse } { Restart w/ Inverse }

Ucode Phase[a], { 1 | 2 | 3 | 11 | 22 | 33 | 44 | 45 }
Note: 1 = Forward Modeling
2 = Sensitivities at Starting Parameters
3 = Perfrom Regression
11 = Calculates Sum of Squares
22 = Sensitivities at Starting Parameters using Central Differences
33 = Calculate Model Linearity
44 = Calculate Prediction Intervals
45 = Calculate Differnences and Prediction Intervals

Ucode Differencing Index[b], { 1 | 2 }
Note: 1 = Forward Differencing (Recommended)
2 = Central Differencing

Ucode Tolerance[c], Ucode Sum-of-Squared Residual Factor[d],
Ucode Quasi-Newton Updating Index[e], { 0 | 1 }
0 = No Quasi-Newton Updating
1 = Quasi-Newton Updating

Maximum Change Factor[f],
Format: *Integer*[a], *Integer*[b], *Real*[c], *Real*[d], *Integer*[e], *Real*[f],

Ucode Path and Name of Inverse Code[a],
Format: *Char*[a],
Ucode Number of Application Models[a],
Format: *Integer*[a],
For: Number of Application Models
Ucode Application Model Execution Commands[a],
Format: *Char*[a],
Endfor:

Ucode Scale Sensitivities Index[a], { 0 | 1 | 2 | 3 }
Note: 0 = No Scaling is Applied and Unscaled Sensitivities are Printed
1 = Dimensionless Scaled Sensitivities are Printed
2 = One-Percent Scaled Sensitivities are Printed
3 = Both Dimensionless and One-Percent Scaled Sensitivities are Printed

Ucode Control Card (cont'd)

Ucode Print Intermediate Index[b], { 0 | 1 }
Note: 0 = No Printing for Intermediate Iterations
1 = Printing for Intermediate Iterations

Ucode Print Graph Index[c], { 0 | 1 }
Note: 0 = Do not Print Post-Processing Files
1 = Print Post-Processing Files

Number of Residual Sets[d],
Format: Integer[a], Integer[b], Integer[c], Integer[d],
Endif:

Endif:

Endcard: Ucode Control Card

B.39 Well Card

Card Title[a] { ~Well Card] }
Format: *Char*[a]

If: Operational Mode Option = **{ STOMP-WO }**

Well Type[a], { Monitoring | Pumping }
Well Bore Radius[b], Units[c],
Initial Well Aqueous Depth[d], Units[e],
Initial Well NAPL Depth[f], Units[g],
Initial Well Head Pressure[h], Units[i],
Initial Well Dissolved-Oil Saturation[j],
Well Bore Radius for Storage[k], Units[l],
Format: *Char*[a], *Real*[b], *Char*[c], *Real*[d], *Char*[e], *Real*[f], *Char*[g], *Real*[h], *Char*[i], *Real*[j], *Real*[k], *Char*[l],

Well I Index[a], Well J Index[b], Well Lower K Index[c], Well Upper K Index[d],
Number of Well Screen Intervals[e],
For: Number of Well Screen Intervals :
<Screen Lower K Index[f], Screen Upper K Index[g],>
Number of Well Times[h],
Format: *Integer*[a], *Integer*[b], *Integer*[c], *Integer*[d], *Integer*[e], <*Integer*[f], *Integer*[g],> *Integer*[h],

Well Time[a], Units[b],
If: Well Type = { Monitoring }
Well Head Pressure[c], Units[d],
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d],
Elseif: Well Type = { Pumping }
Well Head Pressure[c], Units[d],
Injection/Withdrawal Elevation Above Well Bottom[e], Units[f],
Volumetric Pumping Rate[g], Units[h],
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h],
If: Volumetric Pumping Rate > 0 (Injection)
Injection Aqueous Fraction[i],
Injection Dissolved-Oil Saturation[j],
Format: *Real*[a], *Char*[b], *Real*[c], *Char*[d], *Real*[e], *Char*[f], *Real*[g], *Char*[h], *Real*[i], *Real*[j],
Endif
Endif:

Endif:

Endcard:

B.39.1 Well Card Examples

Extracted from STOMP-WO input file:
1,
Monitoring Well,4.0,in,10,m,0,m,101325.0,Pa,0,4.0,in,
1,1,80,100,1,80,100,1,
0.0,day,101325.0,Pa,

Extracted from STOMP-WO input file:
1,
Pumping Well,10.0,cm,5.0,m,5.0,m,101325.0,Pa,1.0,10.0,cm,
1,1,80,100,1,80,90,2,
0.0,day,101325.0,Pa,2.0,m,1,l/hr,0.5,1.0,
100,day,101325.0,Pa,2.0,m,1,l/hr,0.5,1.0,

Extracted from STOMP-WO input file:
1,
Pumping Well,4.0,in,5.0,ft,5.0,ft,101325.0,Pa,1.0,4.0,in,
1,1,80,100,1,80,90,1,
0.0,day,101325.0,Pa,2.0,ft,-0.1,gal/hr,