

SPHERLSgen

1.0

Generated by Doxygen 1.7.6.1

Tue Jan 1 2013 12:54:22



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	eos Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	6
3.1.2	Constructor & Destructor Documentation . . . . .	6
3.1.2.1	eos . . . . .	6
3.1.2.2	eos . . . . .	6
3.1.2.3	eos . . . . .	7
3.1.2.4	~eos . . . . .	7
3.1.3	Member Function Documentation . . . . .	7
3.1.3.1	dDRhoDP . . . . .	7
3.1.3.2	dGetEnergy . . . . .	7
3.1.3.3	dGetOpacity . . . . .	8
3.1.3.4	dGetPressure . . . . .	8
3.1.3.5	dSoundSpeed . . . . .	8
3.1.3.6	gamma1DelAdC_v . . . . .	9
3.1.3.7	getDlnPDlnTDlnPDlnPDEDT . . . . .	9
3.1.3.8	getEAndDTDE . . . . .	9
3.1.3.9	getEKappa . . . . .	10
3.1.3.10	getPAndDRhoDP . . . . .	10
3.1.3.11	getPEKappa . . . . .	10

3.1.3.12	<a href="#">getPEKappaGamma</a>	11
3.1.3.13	<a href="#">getPEKappaGammaCp</a>	11
3.1.3.14	<a href="#">getPKappaGamma</a>	12
3.1.3.15	<a href="#">operator=</a>	12
3.1.3.16	<a href="#">readAscii</a>	12
3.1.3.17	<a href="#">readBin</a>	12
3.1.3.18	<a href="#">readBobsAscii</a>	12
3.1.3.19	<a href="#">writeAscii</a>	13
3.1.3.20	<a href="#">writeBin</a>	13
3.1.4	<a href="#">Member Data Documentation</a>	13
3.1.4.1	<a href="#">dLogE</a>	13
3.1.4.2	<a href="#">dLogKappa</a>	13
3.1.4.3	<a href="#">dLogP</a>	14
3.1.4.4	<a href="#">dLogRhoDelta</a>	14
3.1.4.5	<a href="#">dLogRhoMin</a>	14
3.1.4.6	<a href="#">dLogTDelta</a>	14
3.1.4.7	<a href="#">dLogTMin</a>	14
3.1.4.8	<a href="#">dXMassFrac</a>	15
3.1.4.9	<a href="#">dYMassFrac</a>	15
3.1.4.10	<a href="#">nNumRho</a>	15
3.1.4.11	<a href="#">nNumT</a>	15
3.2	<a href="#">exception2 Class Reference</a>	15
<b>4</b>	<b><a href="#">File Documentation</a></b>	<b>17</b>
4.1	<a href="#">/home/cgeroux/Documents/WORK/SPHERLS_git/src/eos.cpp File Reference</a>	17
4.1.1	<a href="#">Detailed Description</a>	17
4.2	<a href="#">/home/cgeroux/Documents/WORK/SPHERLS_git/src/eos.h File Reference</a>	17
4.2.1	<a href="#">Detailed Description</a>	17

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">eos</a>	<a href="#">5</a>
<a href="#">exception2</a>	<a href="#">15</a>



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

/home/cgeroux/Documents/WORK/SPHERLS_git/src/ <a href="#">eos.cpp</a>	17
/home/cgeroux/Documents/WORK/SPHERLS_git/src/ <a href="#">eos.h</a>	17
/home/cgeroux/Documents/WORK/SPHERLS_git/src/ <b>exception2.h</b>	??
/home/cgeroux/Documents/WORK/SPHERLS_git/src/ <b>xmlFunctions.h</b>	??





## Chapter 3

# Class Documentation

### 3.1 eos Class Reference

```
#include <eos.h>
```

#### Public Member Functions

- [eos](#) ()
- [eos](#) (int [nNumT](#), int [nNumRho](#))
- [eos](#) (const [eos](#) &ref)
- [~eos](#) ()
- [eos & operator=](#) (const [eos](#) &eosRightSide)
- void [readAscii](#) (std::string sFileName)
- void [readBobsAscii](#) (std::string sFileName)
- void [writeAscii](#) (std::string sFileName)
- void [readBin](#) (std::string sFileName) throw (exception2)
- void [writeBin](#) (std::string sFileName)
- double [dGetPressure](#) (double dT, double dRho)
- double [dGetEnergy](#) (double dT, double dRho)
- double [dGetOpacity](#) (double dT, double dRho)
- double [dDRhoDP](#) (double dT, double dRho)
- double [dSoundSpeed](#) (double dT, double dRho)
- void [getEKappa](#) (double dT, double dRho, double &dE, double &dKappa)
- void [getPEKappa](#) (double dT, double dRho, double &dP, double &dE, double &dKappa)
- void [getPEKappaGamma](#) (double dT, double dRho, double &dP, double &dE, double &dKappa, double &dGamma)
- void [getPEKappaGammaCp](#) (double dT, double dRho, double &dP, double &dE, double &dKappa, double &dGamma, double &dCp)
- void [getPKappaGamma](#) (double dT, double dRho, double &dP, double &dKappa, double &dGamma)

- void [gamma1DelAdC\\_v](#) (double dT, double dRho, double &dGamma1, double &dDelAd, double &dC\_v)
- void [getPAndDRhoDP](#) (double dT, double dRho, double &dP, double &dDRhoDP)
- void [getEAndDTDE](#) (double dT, double dRho, double &dE, double &dDTDE)
- void [getDlnPDlnTDlnPDlnPDEDT](#) (double dT, double dRho, double &dDlnPDlnT, double &dDlnPDlnRho, double &dDEDT)

### Public Attributes

- int [nNumRho](#)
- int [nNumT](#)
- double [dXMassFrac](#)
- double [dYMassFrac](#)
- double [dLogRhoMin](#)
- double [dLogRhoDelta](#)
- double [dLogTMin](#)
- double [dLogTDelta](#)
- double \*\* [dLogP](#)
- double \*\* [dLogE](#)
- double \*\* [dLogKappa](#)

### 3.1.1 Detailed Description

This class holds an equation of state as well as many functions useful for manipulating it

### 3.1.2 Constructor & Destructor Documentation

#### 3.1.2.1 `eos::eos( )`

Constructor, doesn't really do anything

References [dLogE](#), [dLogKappa](#), [dLogP](#), [nNumRho](#), and [nNumT](#).

#### 3.1.2.2 `eos::eos( int nNumT, int nNumRho )`

Constructor, allocates memory for the 2D arrays

#### Parameters

<code>in</code>	<code>nNumT</code>	number of temperatures in the equaiton of state table
<code>in</code>	<code>nNumRho</code>	number of densities in the equaiton of state table

### 3.1.2.3 eos::eos ( const eos & ref )

Copy constructor, simply constructs a new eos object from another eos object

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

### 3.1.2.4 eos::~~eos ( )

Destructor, deletes dynamic arrays

References dLogE, dLogKappa, dLogP, and nNumRho.

## 3.1.3 Member Function Documentation

### 3.1.3.1 double eos::dDRhoDP ( double dT, double dRho )

This function calculates the partial derivative of density w.r.t. pressure

#### Parameters

in	$dT$	temperature at which the derivative is to be computed
in	$dRho$	density at which the derivative is to be computed

#### Returns

the partial derivative of density w.r.t. pressure.

References dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

### 3.1.3.2 double eos::dGetEnergy ( double dT, double dRho )

This function linearly interpolates the energy to a given temperature and density. Note that both  $dT$  and  $dRho$  are not in log space.

#### Parameters

in	$dT$	temperature to interpolate to.
in	$dRho$	density to interpolate to.

#### Returns

the interpolated energy.

References dLogE, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

### 3.1.3.3 `double eos::dGetOpacity ( double dT, double dRho )`

This function linearly interpolates the opacity to a given temperature and density. Note that both `dT` and `dRho` are not in log space.

#### Parameters

<code>in</code>	<code>dT</code>	temperature to interpolate to.
<code>in</code>	<code>dRho</code>	density to interpolate to.

#### Returns

the interpolated opacity.

References `dLogKappa`, `dLogRhoDelta`, `dLogRhoMin`, `dLogTDelta`, `dLogTMin`, `nNumRho`, and `nNumT`.

### 3.1.3.4 `double eos::dGetPressure ( double dT, double dRho )`

This function linearly interpolates the pressure to a given temperature and density. Note that both `dT` and `dRho` are not in log space.

#### Parameters

<code>in</code>	<code>dT</code>	temperature to interpolate to.
<code>in</code>	<code>dRho</code>	density to interpolate to.

#### Returns

the interpolated pressure.

References `dLogP`, `dLogRhoDelta`, `dLogRhoMin`, `dLogTDelta`, `dLogTMin`, `nNumRho`, and `nNumT`.

### 3.1.3.5 `double eos::dSoundSpeed ( double dT, double dRho )`

This function calculates the adiabatic sound speed

#### Parameters

<code>in</code>	<code>dT</code>	temperature at which the derivative is to be computed
<code>in</code>	<code>dRho</code>	density at which the derivative is to be computed

## Returns

the sound speed.

References dLogE, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, n-NumRho, and nNumT.

**3.1.3.6** void eos::gamma1DelAdC\_v( double *dT*, double *dRho*, double & *dGamma1*, double & *dDelAd*, double & *dC\_v* )

This function calculates gamma1 and the adiabatic gradient

## Parameters

in	<i>dT</i>	temperature at which the derivative is to be computed
in	<i>dRho</i>	density at which the derivative is to be computed
out	<i>dGamma1</i>	gamma1
out	<i>dDelAd</i>	adiabatic gradient
out	<i>dC_v</i>	specific heat at constant volume

References dLogE, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, n-NumRho, and nNumT.

**3.1.3.7** void eos::getDlnPDlnTDlnPDlnPDEDT( double *dT*, double *dRho*, double & *dDlnPDlnT*, double & *dDlnPDlnRho*, double & *dDEDT* )

This function calculates various partial derivatives

## Parameters

in	<i>dT</i>	temperature at which the derivative is to be computed
in	<i>dRho</i>	density at which the derivative is to be computed
out	<i>dDlnPDlnT</i>	derivative of ln(P) w.r.t. ln(T)
out	<i>dDlnPDlnRho</i>	derivative of ln(P) w.r.t. ln(Rho)
out	<i>dDEDT</i>	derivative of temperature w.r.t. energy at constant density

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

**3.1.3.8** void eos::getEAndDTDE( double *dT*, double *dRho*, double & *dE*, double & *dDTDE* )

This function calculates the partial derivative of temperature w.r.t. energy and the energy

## Parameters

in	<i>dT</i>	temperature at which the derivative is to be computed
in	<i>dRho</i>	density at which the derivative is to be computed

out	$dE$	energy at $dT$ and $dRho$
out	$dDTDE$	derivative of temperature w.r.t. energy at constant density

References  $dLogE$ ,  $dLogRhoDelta$ ,  $dLogRhoMin$ ,  $dLogTDelta$ ,  $dLogTMin$ ,  $nNumRho$ , and  $nNumT$ .

### 3.1.3.9 void eos::getEKappa ( double $dT$ , double $dRho$ , double & $dE$ , double & $dKappa$ )

This function linearly interpolates the three dependent quantities (Pressure, Energy , Opacity) to a given temperature and density. Note that both  $dT$  and  $dRho$  are not in log space.

#### Parameters

in	$dT$	temperature to interpolate to.
in	$dRho$	density to interpolate to.
out	$dE$	energy at $dT$ and $dRho$ .
out	$dKappa$	opacity at $dT$ and $dRho$ .

References  $dLogE$ ,  $dLogKappa$ ,  $dLogRhoDelta$ ,  $dLogRhoMin$ ,  $dLogTDelta$ ,  $dLogTMin$ ,  $nNumRho$ , and  $nNumT$ .

### 3.1.3.10 void eos::getPAndDRhoDP ( double $dT$ , double $dRho$ , double & $dP$ , double & $dDRhoDP$ )

This function calculates the partial derivative of density w.r.t. pressure and the pressure

#### Parameters

in	$dT$	temperature at which the derivative is to be computed
in	$dRho$	density at which the derivative is to be computed
out	$dP$	pressure at $dT$ and $dRho$
out	$dDRhoDP$	derivative of density w.r.t. pressure at conatant temperature

References  $dLogP$ ,  $dLogRhoDelta$ ,  $dLogRhoMin$ ,  $dLogTDelta$ ,  $dLogTMin$ ,  $nNumRho$ , and  $nNumT$ .

### 3.1.3.11 void eos::getPEKappa ( double $dT$ , double $dRho$ , double & $dP$ , double & $dE$ , double & $dKappa$ )

This function linearly interpolates the three dependent quantities (Pressure, Energy , Opacity) to a given temperature and density. Note that both  $dT$  and  $dRho$  are not in log space.

## Parameters

in	$dT$	temperature to interpolate to.
in	$dRho$	density to interpolate to.
out	$dP$	pressure at $dT$ and $dRho$ .
out	$dE$	energy at $dT$ and $dRho$ .
out	$dKappa$	opacity at $dT$ and $dRho$ .

References  $dLogE$ ,  $dLogKappa$ ,  $dLogP$ ,  $dLogRhoDelta$ ,  $dLogRhoMin$ ,  $dLogTDelta$ ,  $dLogTMin$ ,  $nNumRho$ , and  $nNumT$ .

**3.1.3.12** `void eos::getPEKappaGamma ( double  $dT$ , double  $dRho$ , double &  $dP$ , double &  $dE$ , double &  $dKappa$ , double &  $dGamma$  )`

This function linearly interpolates the energy and opacity to a given temperature and density. Note that both  $dT$  and  $dRho$  are not in log space.

## Parameters

in	$dT$	temperature to interpolate to.
in	$dRho$	density to interpolate to.
out	$dP$	pressure at $dT$ and $dRho$ .
out	$dE$	energy at $dT$ and $dRho$ .
out	$dKappa$	opacity at $dT$ and $dRho$ .
out	$dGamma$	adiabatic index at $dT$ and $dRho$ .

References  $dLogE$ ,  $dLogKappa$ ,  $dLogP$ ,  $dLogRhoDelta$ ,  $dLogRhoMin$ ,  $dLogTDelta$ ,  $dLogTMin$ ,  $nNumRho$ , and  $nNumT$ .

**3.1.3.13** `void eos::getPEKappaGammaCp ( double  $dT$ , double  $dRho$ , double &  $dP$ , double &  $dE$ , double &  $dKappa$ , double &  $dGamma$ , double &  $dCp$  )`

This function linearly interpolates the energy and opacity to a given temperature and density. Note that both  $dT$  and  $dRho$  are not in log space.

## Parameters

in	$dT$	temperature to interpolate to.
in	$dRho$	density to interpolate to.
out	$dP$	pressure at $dT$ and $dRho$ .
out	$dE$	energy at $dT$ and $dRho$ .
out	$dKappa$	opacity at $dT$ and $dRho$ .
out	$dGamma$	adiabatic index at $dT$ and $dRho$ .
out	$dCp$	specific heat at constant pressure at $dT$ and $dRho$ .

References  $dLogE$ ,  $dLogKappa$ ,  $dLogP$ ,  $dLogRhoDelta$ ,  $dLogRhoMin$ ,  $dLogTDelta$ ,  $dLogTMin$ ,  $nNumRho$ , and  $nNumT$ .

**3.1.3.14** `void eos::getPKappaGamma( double dT, double dRho, double & dP, double & dKappa, double & dGamma )`

This function linearly interpolates the energy and opacity to a given temperature and density. Note that both *dT* and *dRho* are not in log space.

#### Parameters

in	<i>dT</i>	temperature to interpolate to.
in	<i>dRho</i>	density to interpolate to.
out	<i>dP</i>	pressure at <i>dT</i> and <i>dRho</i> .
out	<i>dKappa</i>	opacity at <i>dT</i> and <i>dRho</i> .
out	<i>dGamma</i>	adiabatic index at <i>dT</i> and <i>dRho</i> .

References *dLogE*, *dLogKappa*, *dLogP*, *dLogRhoDelta*, *dLogRhoMin*, *dLogTDelta*, *dLogTMin*, *nNumRho*, and *nNumT*.

**3.1.3.15** `eos & eos::operator= ( const eos & eosRightSide )`

Assignment operator, assigns one eos object to another.

References *dLogE*, *dLogKappa*, *dLogP*, *dLogRhoDelta*, *dLogRhoMin*, *dLogTDelta*, *dLogTMin*, *nNumRho*, and *nNumT*.

**3.1.3.16** `void eos::readAscii( std::string sFileName )`

This fuction reads in an ascii file and stores it in the current object.

#### Parameters

in	<i>sFileName</i>	name of the equation of state file to read from.
----	------------------	--

References *dLogE*, *dLogKappa*, *dLogP*, *dLogRhoDelta*, *dLogRhoMin*, *dLogTDelta*, *dLogTMin*, *dXMassFrac*, *dYMassFrac*, *nNumRho*, and *nNumT*.

**3.1.3.17** `void eos::readBin( std::string sFileName ) throw (exception2)`

This fuction reads in a binary file and stores it in the current object.

#### Parameters

in	<i>sFileName</i>	name of the equation of state file to read from.
----	------------------	--

**3.1.3.18** `void eos::readBobsAscii( std::string sFileName )`

This fuction reads in an ascii file and stores it in the current object. The ascii file is in Bob's format.



## Parameters

in	<i>sFileName</i>	name of the equation of state file to read from.
----	------------------	--

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.3.19 void eos::writeAscii( std::string *sFileName* )

This fuction writes the equation of state stored in the current object to an ascii file.

## Parameters

in	<i>sFileName</i>	name of the file to write the equation of state to.
----	------------------	---

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.3.20 void eos::writeBin( std::string *sFileName* )

This fuction writes the equation of state stored in the current object to a binary file.

## Parameters

in	<i>sFileName</i>	name of the file to write the equaiton of state to.
----	------------------	---

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

## 3.1.4 Member Data Documentation

## 3.1.4.1 double\*\* eos::dLogE

2D array of log10 energies. dLogE[i][j] gives the log10 energy at log10 density of eos::dLogRhoDelta\*i+eos::dLogRhoMin, and at log10 temperature of eos::dLogTDelta\*j+eos::dLogTMin.

Referenced by dGetEnergy(), dSoundSpeed(), eos(), gamma1DelAdC\_v(), getDlnPDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), writeAscii(), writeBin(), and ~eos().

## 3.1.4.2 double\*\* eos::dLogKappa

2D array of log10 opacities. dLogKappa[i][j] gives the log10 opacity at log10 density of eos::dLogRhoDelta\*i+eos::dLogRhoMin, and at log10 temperature of eos::dLogTDelta\*j+eos::dLogTMin.

Referenced by `dGetOpacity()`, `eos()`, `getDlnPDlnTDlnPDlnPDEDT()`, `getEKappa()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, `writeBin()`, and `~eos()`.

#### 3.1.4.3 `double** eos::dLogP`

2D array of log10 pressures. `dLogP[i][j]` gives the log10 pressure at log10 density of `eos::dLogRhoDelta*i+eos::dLogRhoMin`, and at log10 temperature of `eos::dLogTDelta*j+eos::dLogTMin`.

Referenced by `dDRhoDP()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDT()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, `writeBin()`, and `~eos()`.

#### 3.1.4.4 `double eos::dLogRhoDelta`

Increment of the density between table entries in log10.

Referenced by `dDRhoDP()`, `dGetEnergy()`, `dGetOpacity()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDT()`, `getEAndDTDE()`, `getEKappa()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

#### 3.1.4.5 `double eos::dLogRhoMin`

Minimum density of the table in log10.

Referenced by `dDRhoDP()`, `dGetEnergy()`, `dGetOpacity()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDT()`, `getEAndDTDE()`, `getEKappa()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

#### 3.1.4.6 `double eos::dLogTDelta`

Increment of the temperature between table entries in log10.

Referenced by `dDRhoDP()`, `dGetEnergy()`, `dGetOpacity()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDT()`, `getEAndDTDE()`, `getEKappa()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

#### 3.1.4.7 `double eos::dLogTMin`

Minimum temperature of the table in log10.

Referenced by `dDRhoDP()`, `dGetEnergy()`, `dGetOpacity()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDt()`, `getEAndDTDE()`, `getEKappa()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

#### 3.1.4.8 double eos::dXMassFrac

Hydrogen mass fraction of the composition used to generate the equation of state table.

Referenced by `eos()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

#### 3.1.4.9 double eos::dYMassFrac

Helium mass fraction of the composition used to generate the equation of state table.

Referenced by `eos()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

#### 3.1.4.10 int eos::nNumRho

Number of densities in the equation of state table

Referenced by `dDRhoDP()`, `dGetEnergy()`, `dGetOpacity()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDt()`, `getEAndDTDE()`, `getEKappa()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, `writeBin()`, and `~eos()`.

#### 3.1.4.11 int eos::nNumT

Number of temperatures in the equation of state table

Referenced by `dDRhoDP()`, `dGetEnergy()`, `dGetOpacity()`, `dGetPressure()`, `dSoundSpeed()`, `eos()`, `gamma1DelAdC_v()`, `getDlnPDlnTDlnPDlnPDEDt()`, `getEAndDTDE()`, `getEKappa()`, `getPAndDRhoDP()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBobsAscii()`, `writeAscii()`, and `writeBin()`.

The documentation for this class was generated from the following files:

- [/home/cgeroux/Documents/WORK/SPHERLS\\_git/src/eos.h](/home/cgeroux/Documents/WORK/SPHERLS_git/src/eos.h)
- [/home/cgeroux/Documents/WORK/SPHERLS\\_git/src/eos.cpp](/home/cgeroux/Documents/WORK/SPHERLS_git/src/eos.cpp)

## 3.2 exception2 Class Reference

The documentation for this class was generated from the following files:

- `/home/cgeroux/Documents/WORK/SPHERLS_git/src/exception2.h`
- `/home/cgeroux/Documents/WORK/SPHERLS_git/src/exception2.cpp`

## Chapter 4

# File Documentation

### 4.1 `/home/cgeroux/Documents/WORK/SPHERLS_git/src/eos.cpp` File Reference

```
#include <string> #include <fstream> #include <sstream>
#include <iostream> #include <cmath> #include <stdlib.-
h> #include "eos.h" #include "exception2.h"
```

#### 4.1.1 Detailed Description

Implements the eos (equation of state) class defined in [eos.h](#)

### 4.2 `/home/cgeroux/Documents/WORK/SPHERLS_git/src/eos.h` File Reference

```
#include <string> #include "exception2.h"
```

#### Classes

- class [eos](#)

#### 4.2.1 Detailed Description

Header file for [eos.cpp](#)

# Index

`~eos`  
    [eos](#), [7](#)  
[/home/cgeroux/Documents/WORK/SPH-ERLS\\_git/src/eos.cpp](#), [17](#)  
[/home/cgeroux/Documents/WORK/SPH-ERLS\\_git/src/eos.h](#), [17](#)

`dDRhoDP`  
    [eos](#), [7](#)  
`dGetEnergy`  
    [eos](#), [7](#)  
`dGetOpacity`  
    [eos](#), [7](#)  
`dGetPressure`  
    [eos](#), [8](#)  
`dLogE`  
    [eos](#), [13](#)  
`dLogKappa`  
    [eos](#), [13](#)  
`dLogP`  
    [eos](#), [14](#)  
`dLogRhoDelta`  
    [eos](#), [14](#)  
`dLogRhoMin`  
    [eos](#), [14](#)  
`dLogTDelta`  
    [eos](#), [14](#)  
`dLogTMin`  
    [eos](#), [14](#)  
`dSoundSpeed`  
    [eos](#), [8](#)  
`dXMassFrac`  
    [eos](#), [15](#)  
`dYMassFrac`  
    [eos](#), [15](#)

[eos](#), [5](#)  
    [~eos](#), [7](#)  
    [dDRhoDP](#), [7](#)  
    [dGetEnergy](#), [7](#)  
    [dGetOpacity](#), [7](#)  
    [dGetPressure](#), [8](#)  
    [dLogE](#), [13](#)  
    [dLogKappa](#), [13](#)  
    [dLogP](#), [14](#)  
    [dLogRhoDelta](#), [14](#)  
    [dLogRhoMin](#), [14](#)  
    [dLogTDelta](#), [14](#)  
    [dLogTMin](#), [14](#)  
    [dSoundSpeed](#), [8](#)  
    [dXMassFrac](#), [15](#)  
    [dYMassFrac](#), [15](#)  
    [eos](#), [6](#)  
    [gamma1DelAdC\\_v](#), [9](#)  
    [getDlnPDlnTDlnPDlnPDEDT](#), [9](#)  
    [getEAndDTDE](#), [9](#)  
    [getEKappa](#), [10](#)  
    [getPAndDRhoDP](#), [10](#)  
    [getPEKappa](#), [10](#)  
    [getPEKappaGamma](#), [11](#)  
    [getPEKappaGammaCp](#), [11](#)  
    [getPKappaGamma](#), [11](#)  
    [nNumRho](#), [15](#)  
    [nNumT](#), [15](#)  
    [operator=](#), [12](#)  
    [readAscii](#), [12](#)  
    [readBin](#), [12](#)  
    [readBobsAscii](#), [12](#)  
    [writeAscii](#), [13](#)  
    [writeBin](#), [13](#)  
    [exception2](#), [15](#)  
    [gamma1DelAdC\\_v](#)  
        [eos](#), [9](#)  
    [getDlnPDlnTDlnPDlnPDEDT](#)  
        [eos](#), [9](#)  
    [getEAndDTDE](#)  
        [eos](#), [9](#)  
    [getEKappa](#)  
        [eos](#), [10](#)  
    [getPAndDRhoDP](#)  
        [eos](#), [10](#)

---

getPEKappa  
    eos, [10](#)  
getPEKappaGamma  
    eos, [11](#)  
getPEKappaGammaCp  
    eos, [11](#)  
getPKappaGamma  
    eos, [11](#)  
  
nNumRho  
    eos, [15](#)  
nNumT  
    eos, [15](#)  
  
operator=  
    eos, [12](#)  
  
readAscii  
    eos, [12](#)  
readBin  
    eos, [12](#)  
readBobsAscii  
    eos, [12](#)  
  
writeAscii  
    eos, [13](#)  
writeBin  
    eos, [13](#)