# PECANS (Python Ediable Chemical Atmospheric Numerical Solver)

Generated by Doxygen 1.8.11

# Contents

# Chapter 1

# Format for mechanism input files

PECANS can read the chemical mechanism from any one of several input formats.

# Chapter 2

# PECANS

**(Python Editable Chemical Atmospheric Numeric Solver)**

The goal of the PECANS multi-box model is to provide a relatively straightforward but efficient and flexible idealized atmospheric chemistry modeling framework. It is not intended to supplant global or regional chemical transport models such as GEOS-Chem, WRF-Chem, or CMAQ, but instead to offer the capability to carry out one box to 3D multi-box modeling with idealized (rather than real world) transport.

# Chapter 3

# Namespace Index

## 3.1 Namespace List

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

# Chapter 4

# Hierarchical Index

## 4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

# Chapter 5

# Class Index

## 5.1 Class List

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

# Chapter 6

# Namespace Documentation

## 6.1 pecans Namespace Reference

### Namespaces

- mechgen

### 6.1.1 Detailed Description

```
PECANS: Python Editable Chemical Atmospheric Numeric Solver
```

## 6.2 pecans.mechgen Namespace Reference

### Classes

- class ChemError
- class Derivative
- class RateDefError
- class RateExpression
- class Reaction
- class ReactionDefError
- class ReactionSpecie
- class Specie
- class SpeciesDefError

### Functions

- def generate_pecans_mechanism (species_file, reactions_file, extra_rate_def_files)
- def generate_chemderiv_file (reactions)

### Variables

- **derivative_file** = os.path.join(_mydir, 'chemderiv.pyx')
- string **pyx_indent** = ' '
- string **temperature_variable** = 'TEMP'
- string **ndens_air_variable** = 'CAIR'
- **rate_expr_include_dir** = os.path.join(_mydir, 'Rates')
- list **c_math_fxns** = ['exp', 'sqrt', 'log', 'log10']

### 6.2.1 Detailed Description

```
Generate mechanism solver file from a KPP-like mechanism file or one following PECANS style
```

### 6.2.2 Function Documentation

#### 6.2.2.1 def pecans.mechgen.generate_chemderiv_file ( *reactions* )

```
Generates the chemical derivative .pyx file that contains all the derivative functions
 and rate constant functions plus the interface function that should be called from
 another Python program.
:param reactions: a list of instances of Reactions
:return: nothing
```

#### 6.2.2.2 def pecans.mechgen.generate_pecans_mechanism ( *species_file, reactions_file, extra_rate_def_files* )

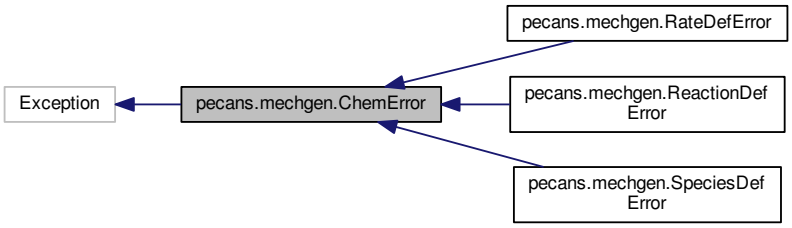
```
Main function that generates a mechanism file for the PECANS style inputs. Any other
input file style must have an equivalent primary function that reads rate expression
files, reads the species file, and reads the reactions file.
:param species_file: the file defining all the species included in the mechanism
:param reactions_file: the file defining the reactions that comprise the mechanism
:param *extra_rate_def_files: any additional files beyond those in the "Rates" subdirectory
 that define rate constant expressions.
:return:
```

# Chapter 7

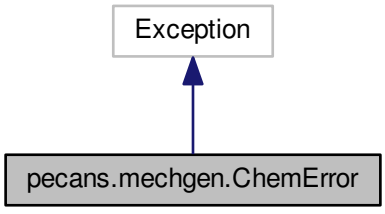
# Class Documentation

## 7.1 pecans.mechgen.ChemError Class Reference

Inheritance diagram for pecans.mechgen.ChemError:




Collaboration diagram for pecans.mechgen.ChemError:




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

- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.2 pecans.mechgen.Derivative Class Reference

### Public Member Functions

- def changed_specie (self)
- def input_species (self)
- def reactions (self)
- def coefficients (self)
- def func_signature (self)
- def __init__ (self, specie)
- def add_rxn_if_relevant (self, rxn)
- def finalize (self)
- def func_def (self)

### 7.2.1 Detailed Description

```
Represents the derivative needed to calculate the change in a chemical specie
```

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 def pecans.mechgen.Derivative.__init__ ( *self, specie* )

```
Initialize an instance of Derivative that can be used to represent the change of a chemical
specie in a single timestep.
:param specie: The specie that the derivative is for; an instance of Specie
:return: an instance of Derivative
```

### 7.2.3 Member Function Documentation

#### 7.2.3.1 def pecans.mechgen.Derivative.add_rxn_if_relevant ( *self, rxn* )

```
Given a reaction, this will determine if the changed_specie appears on either side of the reaction
and if so, adds the reaction to the list of information that the derivative will need
:param rxn: an instance of Reaction
:return: a boolean, True if the reaction was relevant, False otherwise
```

#### 7.2.3.2 def pecans.mechgen.Derivative.changed_specie ( *self* )

```
The specie that the derivative is for
:return: an instance of Specie
```

#### 7.2.3.3 def pecans.mechgen.Derivative.coefficients ( *self* )

```
The coefficients for each reaction, tracking both how many molecules of the changed_specie
are produced or consumed and the sign (positive for produced, negative for consumed). This
is an overall coefficient, so if a specie appears on both sides, there will still be only
one coefficient for that reaction.
:return: a list of floats
```

#### 7.2.3.4 def pecans.mechgen.Derivative.finalize ( *self* )

```
To be called once all reactions are added. This, internally, converts the collection of input species
from a set to a tuple, to ensure that the order is consistent at all times.
:return: nothing
```

#### 7.2.3.5 def pecans.mechgen.Derivative.func_def ( *self* )

```
Creates the Cython function definition for the function that will calculate this derivative.
:return: A list of strings, each string is a line in the function definition
```

#### 7.2.3.6 def pecans.mechgen.Derivative.func_signature ( *self* )

```
The signature of the function that will be called to calculate the derivative. Includes the
function name, argument types, and argument names, i.e. dNO_dt(double conc_NO, double conc_O3)
Only defined after calling finalize() on this instance.
:return: a string that represents the function signature
```

#### 7.2.3.7 def pecans.mechgen.Derivative.input_species ( *self* )

```
The species whose concentrations are needed as inputs to calculate the derivative
:return: a set of instances of Specie
```

#### 7.2.3.8 def pecans.mechgen.Derivative.reactions ( *self* )

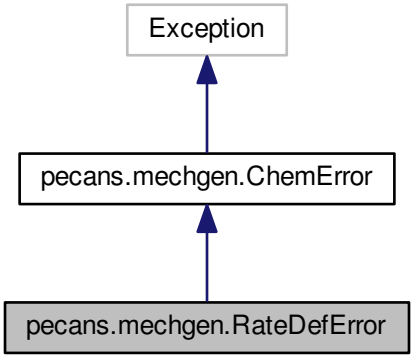
```
The reactions that involve the changed_specie
:return: a list of instances of Reaction
```

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

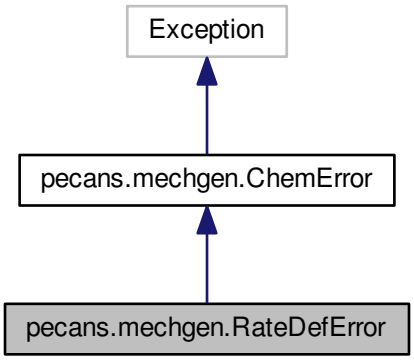
- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.3 pecans.mechgen.RateDefError Class Reference

Inheritance diagram for pecans.mechgen.RateDefError:




Collaboration diagram for pecans.mechgen.RateDefError:




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

- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.4 pecans.mechgen.RateExpression Class Reference

### Public Member Functions

- def __init__ (self, rate_string, rate_file, file_line_number)
- def find_rate_by_name (cls, rate_name)
- def mark_rate_as_needed (cls, rate_call)

### Public Attributes

- **used_in_mech**
- **declaring_file**
- **file_line**
- **ndens_air_ind**
- **body**
- **rate_name**

### Static Public Attributes

- list **instances** = [ ]

## 7.4.1 Detailed Description

```
Class that holds all the defined rate constant expression functions
```

## 7.4.2 Constructor & Destructor Documentation

#### 7.4.2.1 def pecans.mechgen.RateExpression.__init__ ( *self, rate_string, rate_file, file_line_number* )

```
Create a new instance of RateExpression that holds information about a rate expression function.
That instance is stored in the class variable instances (RateExpression.instances), so there is
no need to store the returned instance.
:param rate_string: the string containing the entire function definition for the rate expression.
:param rate_file: the file that the definition was read from. Used to print more helpful error messages.
:param file_line_number: the line number in the file where the definition began. Again, for helpful error
messages.
:return: instance of RateExpression, which is also stored in RateExpression.instances.
```

## 7.4.3 Member Function Documentation

#### 7.4.3.1 def pecans.mechgen.RateExpression.find_rate_by_name ( *cls, rate_name* )

```
Given a rate expression name, finds the instance that corresponds to it.
:param rate_name: The name of the rate expression, as a string
:return: the instance of RateExpression with that name. Raises a KeyError if one is not found.
```

#### 7.4.3.2 def pecans.mechgen.RateExpression.mark_rate_as_needed ( *cls, rate_call* )

```
Marks that the specified rate expression is used in the current mechanism, which causes it to be
inlined in the chemderiv.pyx file.
:param rate_call: the call to the rate expression, e.g. ARR2( 1.2e-12, 1310.0, TEMP )
:return: none
```

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

- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.5 pecans.mechgen.Reaction Class Reference

### Public Member Functions

- def id (self)
- def reactants (self)
- def reactant_species (self)
- def products (self)
- def product_species (self)
- def rate_str (self)
- def __init__ (self, reactants, products, rate_fxn)
- def get_reactant_specie (self, specie)
- def get_product_specie (self, specie)
- def reset (cls)
- def **__str__** (self)
- def **__repr__** (self)

### Public Attributes

- **next_id**

### Static Public Attributes

- int **next_id** = 0

### 7.5.1 Detailed Description

```
Class representing chemical reactions in a mechanism
```

### 7.5.2 Constructor & Destructor Documentation

#### 7.5.2.1 def pecans.mechgen.Reaction.__init__ ( *self, reactants, products, rate_fxn* )

```
Create an instance of Reaction that contains the reactants, products, and rate constant
:param reactants: a list, tuple, or set of instances of ReactionSpecie representing the reactants
:param products: a list, tuple, or set of instances of ReactionSpecie representing the products
:param rate_fxn: a representation of the rate constant, either as a function or value
:return: instance of Reaction
```

### 7.5.3 Member Function Documentation

#### 7.5.3.1 def pecans.mechgen.Reaction.get_product_specie ( *self, specie* )

```
Given an instance of Specie, returns the first corresponding product ReactionSpecie of that type of Specie
:param specie: an instance of Specie representing the chemical specie in question
:return: an instance of ReactionSpecie
```

#### 7.5.3.2 def pecans.mechgen.Reaction.get_reactant_specie ( *self, specie* )

```
Given an instance of Specie, returns the first corresponding reactant ReactionSpecie of that type of Specie
:param specie: an instance of Specie representing the chemical specie in question
:return: an instance of ReactionSpecie
```

#### 7.5.3.3 def pecans.mechgen.Reaction.id ( *self* )

```
The unique numerical ID of the reaction
:return: the ID as an integer
```

#### 7.5.3.4 def pecans.mechgen.Reaction.product_species ( *self* )

```
Returns the unique species present as products
:return: a tuple of instances of Specie
```

#### 7.5.3.5 def pecans.mechgen.Reaction.products ( *self* )

```
Returns the products of the reaction
:return: a tuple of instances of ReactionSpecie
```

#### 7.5.3.6 def pecans.mechgen.Reaction.rate_str ( *self* )

```
Returns a representation of the rate constant as a string
NOTE: behavior may change
:return: a string, a number as a string or the name of the function given
```

#### 7.5.3.7 def pecans.mechgen.Reaction.reactant_species ( *self* )

```
Returns the unique species present as reactants
:return: a tuple of instances of Specie
```

#### 7.5.3.8 def pecans.mechgen.Reaction.reactants ( *self* )

```
Returns the reactants of the reaction
:return: a tuple of instances of ReactionSpecie
```

#### 7.5.3.9 def pecans.mechgen.Reaction.reset ( *cls* )

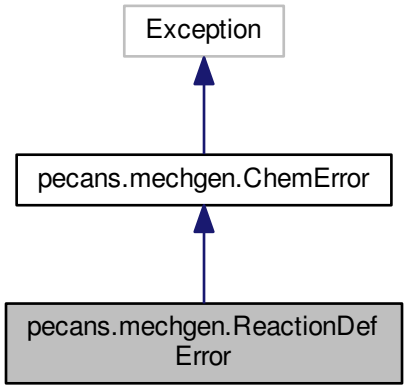
```
Restores the class variables to their initial state
:return: nothing
```

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

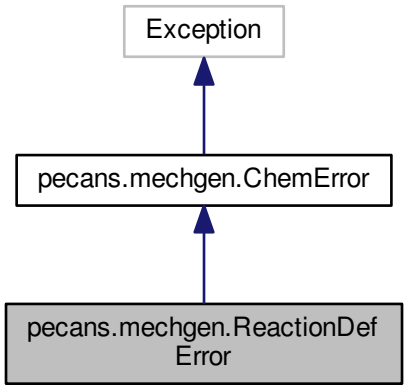
- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.6 pecans.mechgen.ReactionDefError Class Reference

Inheritance diagram for pecans.mechgen.ReactionDefError:




Collaboration diagram for pecans.mechgen.ReactionDefError:




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

- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.7 pecans.mechgen.ReactionSpecie Class Reference

### Public Member Functions

- def **specie** (self)
- def **name** (self)
- def **coefficient** (self)
- def __init__ (self, specie, coefficient=1.0)
- def **__str__** (self)
- def **__repr__** (self)

### 7.7.1 Detailed Description

```
Wrapper class around Specie that matches it with a coefficient
```

### 7.7.2 Constructor & Destructor Documentation

#### 7.7.2.1 def pecans.mechgen.ReactionSpecie.__init__ ( *self, specie, coefficient* = `1.0` )

```
Create a new instance of ReactionSpecie that contains the given specie and
its coefficient in the reaction.
:param specie: An instance of Specie, representing the unique chemical specie in question
:param coefficient: The coefficient (as a float). Defaults to 1.0 if not given
:return: instance of ReactionSpecie
```

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

- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.8 pecans.mechgen.Specie Class Reference

### Public Member Functions

- def name (self)
- def spec_id (self)
- def __init__ (self, name)
- def **__repr__** (self)
- def reset (cls)
- def find_by_name (cls, name, case_sensitive=False)

### Public Attributes

- **instances**
- **next_id**

### Static Public Attributes

- list **instances** = [ ]
- int **next_id** = 0

### 7.8.1 Detailed Description

```
Class which represents unique chemical species in the mechanism
```

### 7.8.2 Constructor & Destructor Documentation

#### 7.8.2.1 def pecans.mechgen.Specie.__init__ ( *self, name* )

```
Create a new, unique species. Each instance is automatically registered with the
class variable "instances", and an error is thrown if the name matches one that
already exists.
:param name:
:return: instance of Specie
```

### 7.8.3 Member Function Documentation

#### 7.8.3.1 def pecans.mechgen.Specie.find_by_name ( *cls, name, case_sensitive =* `False` )

```
Finds the species instance by its name
:param name: the name to search for
:param case_sensitive: if true, matches case, if false (default), does not
:return: an instance of Specie
```

#### 7.8.3.2 def pecans.mechgen.Specie.name ( *self* )

```
The name of the species
:return: the name as a string
```

#### 7.8.3.3 def pecans.mechgen.Specie.reset ( *cls* )

```
Clears the instances list and resets the ID counter
:return: nothing
```

#### 7.8.3.4 def pecans.mechgen.Specie.spec_id ( *self* )

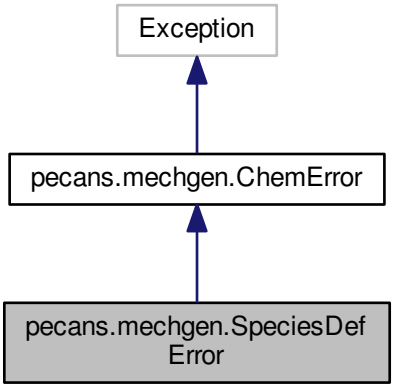
```
The numerical ID of the species
:return: the ID as an integer
```

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

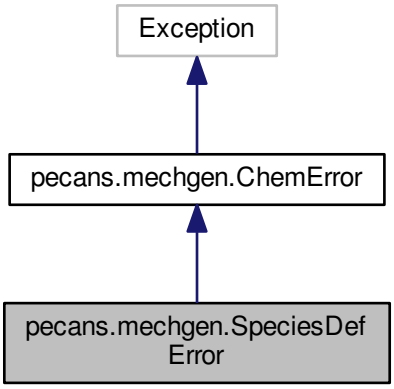
- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

## 7.9 pecans.mechgen.SpeciesDefError Class Reference

Inheritance diagram for pecans.mechgen.SpeciesDefError:




Collaboration diagram for pecans.mechgen.SpeciesDefError:




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

- /home/josh/Documents/Python/VirtEnvs/ChemModel/PECANS/pecans/mechgen.py

# Index