

# OpenKIMTests

## 0.1

Generated by Doxygen 1.7.1

Thu Feb 10 2011 11:13:48



# Contents



# Chapter 1

## Class Index

### 1.1 Class Hierarchy

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

BaseTest::BaseTest . . . . .	??
BCCLattice::BCCLattice . . . . .	??
ElasticModulus::ElasticModulus . . . . .	??
FCCLattice::FCCLattice . . . . .	??
NullTest::NullTest . . . . .	??



# Chapter 2

## Class Index

### 2.1 Class List

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

<a href="#">BaseTest::BaseTest</a>	??
<a href="#">BCCLattice::BCCLattice</a>	??
<a href="#">ElasticModulus::ElasticModulus</a>	??
<a href="#">FCCLattice::FCCLattice</a>	??
<a href="#">NullTest::NullTest</a>	??



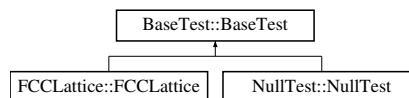


# Chapter 3

## Class Documentation

### 3.1 BaseTest::BaseTest Class Reference

Inheritance diagram for BaseTest::BaseTest:



#### Public Member Functions

- def `__init__`
- def `RequireTest`
- def `XMLWriter`
- def `getASEPotentialByName`
- def `TestResults`
- def `Verify`
- def `main`

#### Public Attributes

- `potentialname`
- `potential`
- `element`
- `verify`
- `write`
- `TestDependencies`

#### 3.1.1 Detailed Description

This is the Base Test from which all other Tests inherit.

Definition at line 48 of file BaseTest.py.

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `def BaseTest::BaseTest::getASEPotentialByName ( self, name )`

A little helper method to call ASE potentials by name.

In the future, to be extended to include KIM potentials

Definition at line 135 of file BaseTest.py.

#### 3.1.2.2 `def BaseTest::BaseTest::main ( self )`

Main is called when the Test is run from the command line. currently runs tests and passes the dictionary of results to the XMLWriter Method

Definition at line 158 of file BaseTest.py.

#### 3.1.2.3 `def BaseTest::BaseTest::TestResults ( self )`

The Test Results method, runs the test and packages the result in a dictionary

Reimplemented in [BCCLattice::BCCLattice](#), [ElasticModulus::ElasticModulus](#), [FCCLattice::FCCLattice](#), and [NullTest::NullTest](#).

Definition at line 148 of file BaseTest.py.

#### 3.1.2.4 `def BaseTest::BaseTest::Verify ( self )`

Optional verify method, creates an easy to check visual verification of test results

Reimplemented in [BCCLattice::BCCLattice](#), [ElasticModulus::ElasticModulus](#), [FCCLattice::FCCLattice](#), and [NullTest::NullTest](#).

Definition at line 153 of file BaseTest.py.

#### 3.1.2.5 `def BaseTest::BaseTest::XMLWriter ( self, resultsdict )`

This method packages the results dictionary into our standard XML Format. The layout is roughly as follows

```
<test id='TestName'>
  <config>
<potential> PotentialName </potential>
<element> Element Symbol </element>
  </config>
  <results>
<FirstResultKey> FirstResultValue </FirstResultKey>
<SecondResultKey> SecondResultValue </SecondResultKey>
  ...
  </results>
</test>
```

Definition at line 69 of file BaseTest.py.

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

- tests/BaseTest.py

## 3.2 BCCLattice::BCCLattice Class Reference

Inheritance diagram for BCCLattice::BCCLattice:

### Public Member Functions

- def [\\_\\_init\\_\\_](#)
- def [BCCEnergy](#)
- def [TestResults](#)
- def [Verify](#)

### 3.2.1 Detailed Description

BCCLattice test returns the optimal bcc lattice constant and energy per atom

Definition at line 16 of file BCCLattice.py.

### 3.2.2 Member Function Documentation

#### 3.2.2.1 def BCCLattice::BCCLattice::\_\_init\_\_( self, potentialname, element, TestDependencies = [], args, kwargs )

Pass the initialization arguments to the BaseTest initializer

Definition at line 19 of file BCCLattice.py.

#### 3.2.2.2 def BCCLattice::BCCLattice::BCCEnergy( self, a )

This function computes the energy of the crystal formation given a certain lattice constant

It uses the ase helper function bulk to create a 1 atom periodic boundary condition crystal with a specific structure

Definition at line 24 of file BCCLattice.py.

#### 3.2.2.3 def BCCLattice::BCCLattice::TestResults( self )

BCC Lattice Test Result

uses scipy fmin (a simplex method minimization tool), to find the optimal lattice constant, and corresponding energy per atom

Reimplemented from [BaseTest::BaseTest](#).

Definition at line 39 of file BCCLattice.py.

### 3.2.2.4 `def BCCLattice::BCCLattice::Verify ( self )`

Simple verification script. Creates a plot that shows the crystal energy in the neighborhood of the computed minimum, along with the computed minimum, as a check

Reimplemented from [BaseTest::BaseTest](#).

Definition at line 58 of file BCCLattice.py.

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

- tests/BCCLattice.py

## 3.3 ElasticModulus::ElasticModulus Class Reference

Inheritance diagram for ElasticModulus::ElasticModulus:

### Public Member Functions

- `def \_\_init\_\_`
- `def FCCEnergy`
- `def FCCTensileLoaded`
- `def getFCCLattice`
- `def TestResults`
- `def Verify`

### Public Attributes

- **volume**
- **a**
- **eps**

### 3.3.1 Detailed Description

ElasticModulus test returns the elastic moduli

Definition at line 16 of file ElasticModulus.py.

### 3.3.2 Member Function Documentation

#### 3.3.2.1 `def ElasticModulus::ElasticModulus::__init__ ( self, potentialname, element, TestDependencies = [], args, kwargs )`

Pass the initialization arguments to the BaseTest initializer

Definition at line 19 of file ElasticModulus.py.

**3.3.2.2 def ElasticModulus::ElasticModulus::FCCEnergy ( *self*, *a* )**

This function computes the energy of the crystal formation given a certain lattice constant

It uses the ase helper function bulk to create a 1 atom periodic boundary condition crystal with a specific structure

Definition at line 24 of file ElasticModulus.py.

**3.3.2.3 def ElasticModulus::ElasticModulus::getFCCLattice ( *self* )**

```
import FCCLattice
fcclattice = FCCLattice.FCCLattice(self.potentialname,self.element)
results = fcclattice.TestResults()
```

Definition at line 50 of file ElasticModulus.py.

**3.3.2.4 def ElasticModulus::ElasticModulus::TestResults ( *self* )**

FCC Lattice Test Result

uses scipy fmin (a simplex method minimization tool), to find the optimal lattice constant, and corresponding energy per atom

Reimplemented from [BaseTest::BaseTest](#).

Definition at line 61 of file ElasticModulus.py.

**3.3.2.5 def ElasticModulus::ElasticModulus::Verify ( *self* )**

Optional verify method, creates an easy to check visual verification of test results

Reimplemented from [BaseTest::BaseTest](#).

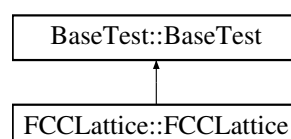
Definition at line 97 of file ElasticModulus.py.

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

- tests/ElasticModulus.py

**3.4 FCCLattice::FCCLattice Class Reference**

Inheritance diagram for FCCLattice::FCCLattice:



## Public Member Functions

- def [\\_\\_init\\_\\_](#)
- def [FCCEnergy](#)
- def [TestResults](#)
- def [Verify](#)

### 3.4.1 Detailed Description

FCCLattice test returns the optimal fcc lattice constant and energy per atom

Definition at line 16 of file FCCLattice.py.

### 3.4.2 Member Function Documentation

#### 3.4.2.1 def FCCLattice::FCCLattice::\_\_init\_\_( *self*, *potentialname*, *element*, *TestDependencies* = [], *args*, *kwargs* )

Pass the initialization arguments to the BaseTest initializer

Definition at line 19 of file FCCLattice.py.

#### 3.4.2.2 def FCCLattice::FCCLattice::FCCEnergy( *self*, *a* )

This function computes the energy of the crystal formation given a certain lattice constant

It uses the ase helper function bulk to create a 1 atom periodic boundary condition crystal with a specific structure

Definition at line 24 of file FCCLattice.py.

#### 3.4.2.3 def FCCLattice::FCCLattice::TestResults( *self* )

FCC Lattice Test Result

uses scipy fmin (a simplex method minimization tool), to find the optimal lattice constant, and corresponding energy per atom

Reimplemented from [BaseTest::BaseTest](#).

Definition at line 39 of file FCCLattice.py.

#### 3.4.2.4 def FCCLattice::FCCLattice::Verify( *self* )

Simple verification script. Creates a plot that shows the crystal energy in the neighborhood of the computed minimum, along with the computed minimum, as a check

Reimplemented from [BaseTest::BaseTest](#).

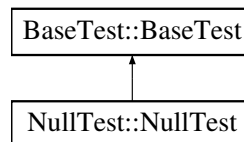
Definition at line 58 of file FCCLattice.py.

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

- tests/FCCLattice.py

## 3.5 NullTest::NullTest Class Reference

Inheritance diagram for NullTest::NullTest:



### Public Member Functions

- def `__init__`
- def `TestResults`
- def `Verify`

### 3.5.1 Detailed Description

NullTest does nothing, but serves as an example test.

It inherits its functionality from BaseTest, and serves as a template for future tests.

Simply copy NullTest.py, and rename the file, and class name, and rewrite TestResults

Definition at line 11 of file NullTest.py.

### 3.5.2 Member Function Documentation

#### 3.5.2.1 def NullTest::NullTest::TestResults ( *self* )

Required module, the TestResults Module returns a dictionary of result.

of the form { 'NameOfValue' : value, 'NameOfSecondValue' : secondvalue }

This is where your test code goes. Feel free to write other methods if necessary.

Reimplemented from [BaseTest::BaseTest](#).

Definition at line 25 of file NullTest.py.

#### 3.5.2.2 def NullTest::NullTest::Verify ( *self* )

Optional verify script to be used to generate a visual output for quick check that everything is going alright

Reimplemented from [BaseTest::BaseTest](#).

Definition at line 36 of file NullTest.py.

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

- tests/NullTest.py