ctools Documentation

Release a

Chris Lee & Christoph Klein

May 12, 2011

CONTENTS

Contents:

CHAPTER

ONE

EXAMPLE

```
>>> Driver.initSystem("Solvated GMX")
>>> # Reading
>>> Driver.loadctools("system_GMX.top")
>>> Driver.loadStructure("system_GMX.gro")
>>> # Writing
>>> Driver.writeStructure("system_GMX_out.gro")
>>> Driver.writectools("system_GMX_out.top")
```

CHAPTER

TWO

SYSTEM

class `ctools.System.`**`System`**(*name=None*)

Bases: `object`

Initialize a new System object. This must be run before the system can be used.

Args: name (str): The name of the system

```
>>> print __init__(name='sysname')
```

`addMolecule`(*molecule*)

Append a molecule into the System.

Args: molecule (Molecule): The molecule object to be appended

`getBoxVector`()

Get the box vector coordinates

`removeMoleculeType`(*molecule*)

Remove a molecule from the System.

Args: molecule (Molecule): The molecule object to be removed

`setBoxVector`(*v1x, v2x, v3x, v1y, v2y, v3y, v1z, v2z, v3z*)

Sets the boxvector for the system. Assumes the box vector is in the correct form. [[v1x,v2x,v3x],[v1y,v2y,v3y],[v1z,v2z,v3z]]

CHAPTER

THREE

MOLECULE

class `ctools.Molecule.`**`Molecule`**`(`*name=None*`)`

Bases: `object`

Initialize the molecule

Args: name (str): name of the molecule

`addAtom``(`*atom*`)`

Add and atom

Args: atom (atom): the atom to add into the molecule

`getAtoms``()`

Return an orderedset of atoms

`removeAtom``(`*atom*`)`

Remove Atom

Args: atom (atom): the atom to remove from the molecule

CHAPTER

FOUR

MOLECULETYPE

class `ctools.MoleculeType.`**`MoleculeType`**(*name*)

Bases: `object`

Initialize the MoleculeType container

Args: name (str): the name of the moleculetype to add

`addForce`(*force*)

Add a force to the moleculeType

Args: force (AbstractForce): Add a force or contraint to the moleculeType

`addMolecule`(*molecule*)

Add a molecule into the moleculetype container

Args: molecule (Molecule): the molecule to append

`getForce`(*force*)

Get a force from the moleculeType

Args: force (AbstractForce): Retrieve a force from the moleculeType

`getMolecule`(*molecule*)

Get a molecule from the system

Args: molecule (Molecule): retrieve an equivalent molecule from the moleculetype

`getNrexcl`()

Gets the nrexcl

`removeForce`(*force*)

Remove a force from the moleculeType

Args: force (AbstractForce): Remove a force from the moleculeType

`removeMolecule`(*molecule*)

Remove a molecule from the system.

Args: molecule (Molecule): remove a molecule from the moleculeType

`setNrexcl`(*nrexcl*)

Set the nrexcl

Args: nrexcl (int): the value for nrexcl

CHAPTER

FIVE

ATOM

class `ctools.Atom.`**`Atom`**(*atomIndex*, *atomName=None*, *residueIndex=-1*, *residueName=None*)

Bases: `object`

Create an Atom object

Args: atomIndex (int): index of atom in the molecule atomName (str): name of the atom (eg., N, C, H, O) residueIndex (int): index of residue in the molecule residueName (str): name of the residue (eg., THR, CYS)

`getAtomType`(*index=None*)

Gets the atomtype

Args: index (str): the value corresponding with type precedence (A Type, B Type)

Returns: atomtype (list, str): Returns the atomtype list or the value at index if index is specified

`getCgnr`(*index=None*)

Gets the Cgnr

Args: index (int): the index to retrieve, defaults to None

Returns: cngr (dict, int): returns the index or the dictionary depending on if index is set

`getCharge`(*index=None*)

Gets the charge of the atom

Args: index (int): index of the charge to retrieve defaults to None

Returns: charge (float): Charge of the atom

`getEpsilon`(*index=None*)

`getForce`()

Gets the force of the atom

Returns: Tuple [fx, fy, fz]

`getMass`(*index=None*)

Gets the mass of the atom

Returns: mass (float): mass of the atom index (str): index to retrieve

`getPosition`()

Gets the position fo the atom

Returns: Tuple [x, y, z]

`getSigma`(*index=None*)

`getVelocity``()`

Gets the velocity of the atom

Returns: Tuple [vx, vy, vz]

`setAtomType``(`*index*, *atomtype*`)`

Sets the atomtype

Args: atomtype (str): the atomtype of the atom index (str): the value corresponding with type precedence (A Type, B Type)

`setCgnr``(`*index*, *cgnr*`)`

Sets the Cgnr

Args: cgnr (int): The charge group number index (int): the value corresponding with cgnr precedence

`setCharge``(`*index*, *charge*`)`

Sets the charge of the atom

Args: charge (float): Charge of the atom index (int): the index corresponding with charge precedence

`setEpsilon``(`*index*, *epsilon*`)`

Sets the epsilon

Args: epsilon (float): epsilon of the atom index(int): index corresponding to epsilon

`setForce``(`*fx*, *fy*, *fz*`)`

Sets the force of the atom

Args: fx (float): x force fy (float): y force fz (float): z force

`setMass``(`*index*, *mass*`)`

Sets the mass of the atom

Args: mass (float): mass of the atom index (str): the index corresponding with mass precedence (A Mass, B Mass)

`setPosition``(`*x*, *y*, *z*`)`

Sets the position of the atom

Args: x (float): x position y (float): y position z (float): z position

`setSigma``(`*index*, *sigma*`)`

Sets the sigma

Args: sigma (float): sigma of the atom index (int): index to insert at

`setVelocity``(`*vx*, *vy*, *vz*`)`

Sets the velocity of the atom

Args: vx (float): x velocity vy (float): y velocity vz (float): z velocity

CHAPTER

SIX

HASHMAP

class `ctools.HashMap.`**`HashMap`**

Bases: `object`

Initializes the HashMap class

`add``(`*key*`)`

Add a value to the container

Args: key: the key to add

`get``(`*key*`)`

Retrieve a key from the container

Args: key: key to retrieve

`itervalues``()`

Return a list of values

`remove``(`*key*`)`

Remove a key from the container

Args: key: key to remove

CHAPTER

SEVEN

ORDEREDSET

class `ctools.OrderedSet.`**`OrderedSet`**

Bases: `object`

Initialize the orderedSet. Essentially a map coupled with a list.

`add`(*key*)

Add a key to the container

Args: key: key to add

`get`(*key*)

Get a key from the container

Args: key: key to retrieve from the container

`pop`(*last=True*)

Pop a value from the container

Args: last (boolean): if true it pops the last element otherwise the first

`remove`(*key*)

Remove a value from the container

Args: key: key to remove:w

CHAPTER

EIGHT

CONVERTER

`ctools.Converter.`**`convert_units`**(*arg*, *unit*)

Checks compatibility and converts units using simtk.units package

Args: arg (Quantity): quantity to be converted unit (Unit): Unit to be converted to

Returns: arg (Quantity): Quantity scaled to the new unit

CHAPTER

NINE

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*

PYTHON MODULE INDEX

INDEX