

## NAME

AtomTypes - AtomTypes class

## SYNOPSIS

```
use AtomTypes::AtomTypes;

use AtomTypes::AtomTypes qw(:all);
```

## DESCRIPTION

AtomTypes base class used to derive all other atom types classes provides the following methods:

new, GetAtomType, GetAtomTypes, IsAtomTypesAssignmentSuccessful, LoadAtomTypesData, SetAtomType, SetMolecule, SetType

AtomTypes class is derived from ObjectProperty base class which provides methods not explicitly defined in Fingerprints or ObjectProperty classes using Perl's AUTOLOAD functionality. These methods are generated on-the-fly for a specified object property:

```
Set<PropertyName>(<PropertyValue>);
$PropertyValue = Get<PropertyName>();
Delete<PropertyName>();
```

## METHODS

new

```
$NewAtomTypes = new AtomTypes::AtomTypes(%NamesAndValues);
```

Using specified *AtomTypes* property names and values hash, new method creates a new object and returns a reference to newly created AtomTypes object. By default, following properties are initialized:

```
Molecule = '';
Type = '';
IgnoreHydrogens = 0;
```

GetAtomType

```
$AtomType = $AtomTypes->GetAtomType($Atom);
```

Returns AtomType value string assigned to *Atom* by *AtomTypes* object.

GetAtomTypes

```
%AtomTypes = $AtomTypes->GetAtomTypes();
```

Returns atom types assigned to atoms by *AtomTypes* object as a hash with atom ID and atom types as key and value pairs.

IsAtomTypesAssignmentSuccessful

```
$Status = $AtomTypes->IsAtomTypesAssignmentSuccessful();
```

Returns 1 or 0 based on whether atom types assignment was successfully performed. For a successful atom types assignment, all atoms must have an atom type other than a string *None*.

LoadAtomTypesData

```
$AtomTypes->LoadAtomTypesData($AtomTypesDataMapRef);
AtomTypes::AtomTypes::LoadAtomTypesData($AtomTypesDataMapRef);
```

Loads atom types data from the specified CSV atom type file into the specified hash reference.

The lines starting with # are treated as comments and ignored. First line not starting with # must contain column labels and the number of columns in all other data rows must match the number of column labels.

The first column is assumed to contain atom types; all other columns contain data as indicated in their column labels.

In order to avoid dependence of data access on the specified column labels, the column data is loaded into hash with *DataColNum* and *AtomType* as hash keys; however, the data for the first column which is treated as AtomTypes is also loaded into an array with AtomTypes as hash key. The format of the data structure loaded into a specified hash reference is:

```
@{$AtomTypesDataMapRef->{AtomTypes}} - Array of all possible atom
                                     types for all atoms
@{$AtomTypesDataMapRef->{NonHydrogenAtomTypes}} - Array of all possible
                                                  atom types for non-hydrogen atoms
@{$AtomTypesDataMapRef->{ColLabels}} - Array of column labels
%{$AtomTypesDataMapRef->{DataCol<ColNum>}} - Hash keys pair:
                                     <DataCol<ColNum>, AtomType>
```

*ColNum* starts from 1. Column data for first column is not loaded into *DataColNum*, *AtomType* hash keys pairs.

SetAtomType

```
$AtomTypes->SetAtomType($Atom, $AtomType);
```

Assigns specific *AtomType* to *Atom* and returns *AtomTypes*.

#### SetMolecule

```
$AtomTypes->SetMolecule($Molecule);
```

Sets *Molecule* object for *AtomTypes* and returns *AtomTypes*.

#### SetType

```
$AtomTypes->SetType($Type);
```

Sets *Type* for *AtomTypes* object and returns *AtomTypes*.

#### AUTHOR

Manish Sud <msud@san.rr.com>

#### SEE ALSO

AtomicInvariantsAtomTypes.pm, DREIDINGAtomTypes.pm, EStateAtomTypes.pm, FunctionalClassAtomTypes.pm, MMFF94AtomTypes.pm, SLogPAtomTypes.pm, SYBYLAtomTypes.pm, TPSAAtomTypes.pm, UFFAtomTypes.pm

#### COPYRIGHT

Copyright (C) 2018 Manish Sud. All rights reserved.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.