

NAME

EStateAtomTypes

SYNOPSIS

```
use AtomTypes::EStateAtomTypes;

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

DESCRIPTION

EStateAtomTypes class provides the following methods:

new, AssignAtomTypes, GetAllPossibleEStateAtomTypes, GetAllPossibleEStateNonHydrogenAtomTypes, GetEStateAtomTypesData, StringifyEStateAtomTypes

The following functions are available:

GetAllPossibleEStateAtomTypes, GetAllPossibleEStateNonHydrogenAtomTypes, GetEStateAtomTypesData

EStateAtomTypes is derived from AtomTypes class which in turn is derived from ObjectProperty base class that provides methods not explicitly defined in EStateAtomTypes, AtomTypes 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>();
```

The data file EStateAtomTypes.csv distributed with MayaChemTools release contains all possible electrotopological state (E-state) [Ref 75-78] atom types.

E-state atom types for various different atom groups [Appendix Table 1 in Ref 76, Appendix III in Ref 77] are defined using central atom environments indicating its topological and valence state along with bonded hydrogens.

The current release of MayaChemTools implements an extended E-state atom assignment methodology which is able to assign atom types to any valid non-hydrogen atom in any atom group instead of a fixed set of E-state atom types [Ref 77].

Let:

As = Atom symbol corresponding to element symbol

H<n> = Number of implicit and explicit hydrogens for atom

s = Single bond to non-hydrogen atoms attached to atom

s<x> = Symbol s repeated x times to indicate multiple single bonds

d = Double bond to non-hydrogen atoms attached to atom

d<x> = Symbol d repeated x times to indicate multiple double bonds

t = Triple bond to non-hydrogen atoms attached to atom

t<x> = Symbol t repeated x times to indicate multiple triple bonds

a = Aromatic to bond non-hydrogen atoms attached to atom

t<x> = Symbol a repeated x times to indicate multiple aromatic bonds

p = Plus or positive formal charge

m = Minus or negative formal charge

Then, E-state atom type specification for non-hydrogen or heavy atoms corresponds to:

t<x>d<x>a<x>s<x>AsH<n>p or t<x>d<x>a<x>s<x>AsH<n>m

Notes:

o p and n with values of 0 are not shown.

o s, d, t, and a bond symbol with values of zero are not shown.

o s and d bonds which are also aromatic don't contribute to the count of single and double bonds; instead, aromatic bond count reflect these bonds.

Hydrogen E-state [Ref 76-77] atom type definitions are:

HGroup AtomType

```
-OH      HsOH
-SH      HsSH
```

-NH2	HsNH2	
>NH	HssNH	
=NH	HdNH	
:NH:	HaaNH	
-NH3+	HsNH3p	
>NH2+	HssNH2p	
>NH-+	HsssNHp	
#CH	HtCH	
=CH2	HdCH2	- H attached to a terminal vinyl group
=CH-	HdsCH	- H attached a non-terminal vinyl group
:CH:	HaaCH	
>CHF	HCHF	
-CH2F	HCH2F	
>CHCl	HCHCl	
-CH2Cl	HCH2Cl	
CHn (saturated)	HCsats	- H attached to sp3 carbon attached to saturated carbon(s)
CHn (unsatd.)	HCsatu	- H attached to sp3 carbon attached to unsaturated carbon(s)
CHn (aromatic)	Havin	- H attached to a non-terminal vinyl group, =CH-, attached to an aromatic carbon
CHn	Hother	- H attached to any other type of C, N, O or S
AHn	Hmisc	- H not attached to C, N, O or S

Notes:

- o - : Single bond; = : Double bond; # : Triple bond
- o Hother atom type capture Hydrogen atom groups not explicitly defined.
- o HGroup doesn't explicitly corresponds to functional groups
- o -OH group could be a hydroxyl group or part of carboxylic acid group and so on
- o -NH2 group could be primary amine or part of an amide group and so on

Examples of E-state atom types for non-hydrogen or heavy atoms:

sCH3, dCH2, dsCH, ddC, aasC, sNH2 and so on

METHODS

new

```
$NewEStateAtomTypes = new AtomTypes::EStateAtomTypes(%NamesAndValues);
```

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

```
Molecule = ''
Type = 'EState'
IgnoreHydrogens = 0
```

Examples:

```
$EStateAtomTypes = new AtomTypes::EStateAtomTypes(
    'Molecule' => $Molecule,
    'IgnoreHydrogens' => 0);
```

AssignAtomTypes

```
$EStateAtomTypes->AssignAtomTypes();
```

Assigns E-state atom types to all the atoms in a molecule and returns *EStateAtomTypes*.

GetAllPossibleEStateAtomTypes

```
$AllAtomTypesDataRef = $EStateAtomTypes->
    GetAllPossibleEStateAtomTypes();
$AllAtomTypesDataRef = AtomTypes::EStateAtomTypes::
    GetAllPossibleEStateAtomTypes();
```

Returns all possible EState atom types corresponding to hydrogen and non-hydrogen atoms as an array reference.

GetAllPossibleEStateNonHydrogenAtomTypes

```
$AtomTypesDataRef = $EStateAtomTypes->
```

```

        GetAllPossibleEStateNonHydrogenAtomTypes();
$AtomTypesDataMapRef = AtomTypes::EStateAtomTypes::
        GetAllPossibleEStateNonHydrogenAtomTypes();

```

Returns all possible EState atom types corresponding to non-hydrogen atoms as an array reference.

GetEStateAtomTypesData

```

$AtomTypesDataMapRef = $EStateAtomTypes->GetEStateAtomTypesData();
$AtomTypesDataMapRef = AtomTypes::EStateAtomTypes::
        GetEStateAtomTypesData();

```

Returns EState atom types and associated data loaded from EState data file as a reference to hash with the following hash data format:

```

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

```

StringifyEStateAtomTypes

```

$string = $EStateAtomTypes->StringifyEStateAtomTypes();

```

Returns a string containing information about *EStateAtomTypes* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

AtomTypes.pm, AtomicInvariantsAtomTypes.pm, DREIDINGAtomTypes.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.