NAME

FunctionalClassAtomTypes

SYNOPSIS

use AtomTypes::FunctionalClassAtomTypes;

use AtomTypes::FunctionalClassAtomTypes qw(:all);

DESCRIPTION

FunctionalClassAtomTypes class provides the following methods:

new, AssignAtomTypes, GetAvailableFunctionalClasses, GetFunctionalClassesOrder, IsFunctionalClassAvailable, SetFunctionalClassesToUse, StringifyFunctionalClassAtomTypes

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

Possible values for functional clas atom types are: *Ar, CA, H, HBA, HBD, Hal, NI, PI, RA*. Default value: *HBD, HBA, PI, NI, Ar, Hal*.

The functional calss atom types abbreviations correspond to:

```
HBD: HydrogenBondDonor
HBA: HydrogenBondAcceptor
PI :  PositivelyIonizable
NI : NegativelyIonizable
Ar : Aromatic
Hal : Halogen
H : Hydrophobic
RA : RingAtom
CA : ChainAtom
```

FunctionalAtomTypes are assigned using the following definitions [Ref 60-61, Ref 65-66]:

```
HydrogenBondDonor: NH, NH2, OH
HydrogenBondAcceptor: N[!H], O
PositivelyIonizable: +, NH2
NegativelyIonizable: -, C(=O)OH, S(=O)OH, P(=O)OH
```

Notes:

```
o Final functional class atom type shows only those functional
  classes to which an atom belongs; others are not shown.
o A null string is assigned as final atom type to those atom which
  don't belong to any of the specified functional classes.
```

```
 Examples of functional class atom types:
```

```
HBD.HBA - Hydrogen bond donor and acceptor
HBD.RA - Hydrogen bond donor in a ring
```

METHODS

new

```
$NewFunctionalClassAtomTypes = new AtomTypes::FunctionalClassAtomTypes(
                                                %NamesAndValues);
```

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

```
Molecule = ''
Type = 'FunctionalClass'
IgnoreHydrogens = 0
FunctionalClassesToUse = HBD, HBA, PI, NI, Ar, Hal
```

Examples:

```
$FunctionalClassAtomTypes = new AtomTypes::FunctionalClassAtomTypes(
```

```
                                        'Molecule' => $Molecule,
                                        'IgnoreHydrogens' => 0,
                                        'FunctionalClassesToUse' =>
                                                  ['HBD', 'HBA', 'PI', 'NI', 'Ar', 'Hal']);
```

AssignAtomTypes

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

Assigns functional class atom types to all the atoms in a molecule and returns *FunctionalClassAtomTypes*.

GetAvailableFunctionalClasses

```
    %AvailableFunctionalClasses = $FunctionalClassAtomTypes->
                                  GetAvailableFunctionalClasses();
```

Returns available functional classes as a hash containing available functional classes and their description as key/value pairs.

GetFunctionalClassesOrder

```
    @FunctionalClassesOrder = $FunctionalClassAtomTypes->
                              GetFunctionalClassesOrder();
```

Returns an array obtaining order of functional classes used to generate atom types.

IsAtomTypesAssignmentSuccessful

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

Returns 1 or 0 based on whether atom types assignment was successfully performed. This method overrides the same method available in the base class AtomTypes.pm used to derived this class.

IsFunctionalClassAvailable

```
    $Status = $FunctionalClassAtomTypes->
              IsFunctionalClassAvailable($FunctionalClass);
    $Status = AtomTypes::FunctionalClassAtomTypes::
              IsFunctionalClassAvailable($FunctionalClass);
```

Returns 1 or 0 based on whether *FunctionalClass* is valid.

SetFunctionalClassesToUse

```
    $FunctionalClassAtomTypes->SetFunctionalClassesToUse($ValuesRef);
    $FunctionalClassAtomTypes->SetFunctionalClassesToUse(@Values);
```

Set functional classes to use for generating and assigning atom types and returns *FunctionalClassAtomTypes*.

StringifyFunctionalClassAtomTypes

```
    $String = $FunctionalClassAtomTypes->StringifyFunctionalClassAtomTypes();
```

Returns a string containing information about *FunctionalClassAtomTypes* object.

AUTHOR

Manish Sud <msud@san.rr.com>

SEE ALSO

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

COPYRIGHT