

## NAME

MoleculeFileIO

## SYNOPSIS

```
use MoleculeFileIO;

use MoleculeFileIO qw(:all);
```

## DESCRIPTION

MoleculeFileIO class provides the following methods:

new, Close, IsSupportedMoleculeFileFormat, Open, ReadMolecule, ReadMoleculeString, WriteMolecule

The following methods can also be used as functions:

IsSupportedMoleculeFileFormat

## METHODS

new

```
$NewMoleculeFileIO = new MoleculeFileIO([%PropertyNameAndValues]);
```

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

```
Name = ""
Mode = ""
FileIORef = ""
```

Based on extension of specified file *Name*, an input class is automatically associated to provide molecule read and write methods.

Examples:

```
$Name = "Water.mol";
$Mode = "Read";
$MoleculeFileIO = new MoleculeFileIO('Name' => $Name,
                                         'Mode' => $Mode);

$MoleculeFileIO->Open();
$Molecule = $MoleculeFileIO->ReadMolecule();
$Molecule->DetectRings();
print "$Molecule\n";
$MoleculeFileIO->Close();

$MoleculeFileIO = new MoleculeFileIO('Name' => 'Sample1.sdf',
                                         'Mode' => 'Read');

$MoleculeFileIO->Open();
while ($Molecule = $MoleculeFileIO->ReadMolecule()) {
    $Molecule->DetectRings();
    print "$Molecule\n";

    $DataLabelsAndValuesRef =
        $Molecule->GetDataFieldLabelAndValues();
    for $DataLabel (sort keys %{ $DataLabelsAndValuesRef } ) {
        $DataValue = $DataLabelsAndValuesRef->{ $DataLabel };
        print "<DataLabel: $DataLabel; DataValue: $DataValue>; ";
    }
    print "\n";
}
$MoleculeFileIO->Close();
```

Close

```
$MoleculeFileIO->Close();
```

Closes an open file

IsSupportedMoleculeFileFormat

```
$Status = $MoleculeFileIO->IsSupportedMoleculeFileFormat($Name);
```

```
$Status = MoleculeFileIO::IsSupportedMoleculeFileFormat($Name);  
($Status, $FormatType, $IOClassName) =  
    $MoleculeFileIO::IsSupportedMoleculeFileFormat($Name);
```

Returns 1 or 0 based on whether input file *Name* format is supported. In list context, value of supported format type and name of associated IO class is also returned.

File extension is used to determine file format. Currently, following file extensions are supported:

```
FileExts - FormatType - AssociatedIOClassName  
  
.mol - MDLMOL - MDLMolFileIO  
.sdf, .sd - SDF - SDFFileIO
```

#### Open

```
$MoleculeFileIO->Open([$Mode]);
```

Opens a file in a specified *Mode*. Default mode value: *Read*. Supported mode values:

Read, Write, Append, <, >, >>, r, w, or a

#### ReadMolecule

```
$Molecule = $MoleculeFileIO->ReadMolecule();
```

Reads molecule data from the file and returns a *Molecule* object.

#### ReadMoleculeString

```
$MoleculeString = $MoleculeFileIO->ReadMoleculeString();
```

Reads molecule data from a file and returns a *Molecule* string.

#### WriteMolecule

```
$MoleculeFileIO->WriteMolecule();
```

Write molecule data to a file for a *Molecule*.

#### AUTHOR

Manish Sud <msud@san.rr.com>

#### SEE ALSO

FileIO.pm, MDLMolFileIO.pm, SDFFileIO.pm

#### COPYRIGHT

Copyright (C) 2024 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.