

## NAME

SDFFileIO

## SYNOPSIS

```
use SDFFileIO;

use SDFFileIO qw(:all);
```

## DESCRIPTION

SDFFileIO class provides the following methods:

new, GenerateMoleculeString, IsSDFFile, ParseMoleculeString, ReadMolecule, ReadMoleculeString, WriteMolecule

The following methods can also be used as functions:

GenerateMoleculeString, IsSDFFile, ParseMoleculeString

Data specific to SDFFileIO class not directly used by Molecule, Atom and Bond objects - data label/value pairs, atom StereoParity and so on - is associated to and retrieved from appropriate objects using following methods:

```
SetMDL<PropertyName>
GetMDL<PropertyName>.
```

SD data label and values are attached to Molecule object as a reference to a hash using SetMDLDataFieldLabelAndValues and can be retrieved using GetMDLDataFieldLabelAndValues method.

SDFFileIO class is derive from *FileIO* class and uses its methods to support generic file related functionality.

## METHODS

new

```
$NewSDFFileIO = new SDFFileIO(%NamesAndValues);
```

Using specified *SDFFileIO* property names and values hash, new method creates a new object and returns a reference to newly created SDFFileIO object.

GenerateMoleculeString

```
$MoleculeString = $SDFFileIO->GenerateMoleculeString($Molecule);
$MoleculeString = SDFFileIO::GenerateMoleculeString($Molecule);
```

Returns a MoleculeString in SD format corresponding to *Molecule*

IsSDFFile

```
$Status = $SDFFileIO->IsSDFFile($FileName);
$Status = SDFFileIO::IsSDFFile($FileName);
```

Returns 1 or 0 based on whether *FileName* is a SD file

ParseMoleculeString

```
$Molecule = $SDFFileIO->ParseMoleculeString($MoleculeString);
$Molecule = SDFFileIO::ParseMoleculeString($MoleculeString);
```

Parses *MoleculeString* and returns a Molecule object. SD data field label and value pairs are associated to Molecule object as a reference to a hash using:

```
$Molecule->SetMDLDataFieldLabelAndValues(\%DataLabelsAndValues)
```

The reference to hash can be retrieved by:

```
$DataLabelsAndValues = $Molecule->GetMDLDataFieldLabelAndValues();
for $DataLabel (sort keys %{$DataLabelsAndValues}) {
    $DataValue = $DataLabelsAndValues->{$DataLabel};
}
```

**ReadMolecule**

```
$Molecule = $SDFFileIO->ReadMolecule($FileHandle);
```

Reads data for the next compound in a file using already opened *FileHandle*, creates, and returns a Molecule object

**ReadMoleculeString**

```
$MoleculeString = $SDFFileIO->ReadMoleculeString($FileHandle);
```

Reads data for the next compound in a file using already opened *FileHandle* and returns a MoleculeString corresponding to compound structure and other associated data

**WriteMolecule**

```
$SDFFileIO->WriteMolecule($Molecule);
```

Writes *Molecule* data to a file in MDLMol format and returns SDFFileIO

**AUTHOR**

Manish Sud <msud@san.rr.com>

**SEE ALSO**

MoleculeFileIO.pm, MDLMolFileIO.pm

**COPYRIGHT**

Copyright (C) 2004-2008 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.