

## NAME

FingerprintsTextFileIO

## SYNOPSIS

```
use FileIO::FingerprintsTextFileIO;

use FileIO::FingerprintsTextFileIO qw(:all);
```

## DESCRIPTION

FingerprintsTextFileIO class provides the following methods:

new, GetDataColLabels, GetDataLineWords, GetFingerprints, GetFingerprintsString, IsFingerprintsDataValid, IsFingerprintsFileDataValid, IsFingerprintsTextFile, Next, Read, SetBitStringFormat, SetBitsOrder, SetColMode, SetDataColLabels, SetDataLineWords, SetDetailLevel, SetFingerprints, SetFingerprintsString, SetFingerprintsStringMode, SetInDelim, SetOutDelim, SetVectorStringFormat, WriteFingerprints, WriteFingerprintsString

The following methods can also be used as functions:

IsFingerprintsTextFile

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

The fingerprints CSV/TSV text file format with .csv or .tsv file extensions supports two types of fingerprints string data: fingerprints bit-vectors and fingerprints vector strings. The fingerprints string data is treated as column value in a text file.

Example of text file format containing fingerprints string data:

```
"CompoundID", "PathLengthFingerprints"
"Cmpd1", "FingerprintsBitVector;PathLengthBits:AtomicInvariantsAtomTypes
:MinLength1:MaxLength8;1024;HexadecimalString;Ascending;9c8460989ec8a4
9913991a6603130b0a19e8051c89184414953800cc2151082844a20104280013086030
8e8204d402800831048940e44281c00060449a5000ac80c894114e006321264401..."
... ..
... ..
```

The current release of MayaChemTools supports the following types of fingerprint bit-vector and vector strings:

```
FingerprintsVector;AtomNeighborhoods:AtomicInvariantsAtomTypes:MinRadi
us0:MaxRadius2;41;AlphaNumericalValues;ValuesString;NR0-C.X1.BO1.H3-AT
C1:NR1-C.X3.BO3.H1-ATC1:NR2-C.X1.BO1.H3-ATC1:NR2-C.X3.BO4-ATC1 NR0-C.X
1.BO1.H3-ATC1:NR1-C.X3.BO3.H1-ATC1:NR2-C.X1.BO1.H3-ATC1:NR2-C.X3.BO4-A
TC1 NR0-C.X2.BO2.H2-ATC1:NR1-C.X2.BO2.H2-ATC1:NR1-C.X3.BO3.H1-ATC1:NR2
-C.X2.BO2.H2-ATC1:NR2-N.X3.BO3-ATC1:NR2-O.X1.BO1.H1-ATC1 NR0-C.X2.B...
```

```
FingerprintsVector;AtomTypesCount:AtomicInvariantsAtomTypes:ArbitraryS
ize;10;NumericalValues;IDsAndValuesString;C.X1.BO1.H3 C.X2.BO2.H2 C.X2
.BO3.H1 C.X3.BO3.H1 C.X3.BO4 F.X1.BO1 N.X2.BO2.H1 N.X3.BO3 O.X1.BO1.H1
O.X1.BO2;2 4 14 3 10 1 1 1 3 2
```

```
FingerprintsVector;AtomTypesCount:SLogPAtomTypes:ArbitrarySize;16;Nume
ricalValues;IDsAndValuesString;C1 C10 C11 C14 C18 C20 C21 C22 C5 CS F
N11 N4 O10 O2 O9;5 1 1 1 14 4 2 1 2 2 1 1 1 1 3 1
```

```
FingerprintsVector;AtomTypesCount:SLogPAtomTypes:FixedSize;67;OrderedN
umericalValues;IDsAndValuesString;C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 C11 C
12 C13 C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 CS N1 N
2 N3 N4 N5 N6 N7 N8 N9 N10 N11 N12 N13 N14 NS O1 O2 O3 O4 O5 O6 O7 O8
O9 O10 O11 O12 OS F Cl Br I Hal P S1 S2 S3 Me1 Me2;5 0 0 0 2 0 0 0 0 1
1 0 0 1 0 0 0 14 0 4 2 1 0 0 0 0 0 2 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0...
```

```
FingerprintsVector;EStateIndicies:ArbitrarySize;11;NumericalValues;IDs
AndValuesString;SaaCH SaasC SaasN SdO SdssC SsCH3 SsF SsOH SssCH2 SssN
H SsssCH;24.778 4.387 1.993 25.023 -1.435 3.975 14.006 29.759 -0.073 3
.024 -2.270
```





```

Distance1:MaxDistance10;54;NumericalValues;IDsAndValuesString;H-D1-H H
-D1-NI HBA-D1-NI HBD-D1-NI H-D2-H H-D2-HBA H-D2-HBD HBA-D2-HBA HBA-D2-
HBD H-D3-H H-D3-HBA H-D3-HBD H-D3-NI HBA-D3-NI HBD-D3-NI H-D4-H H-D4-H
BA H-D4-HBD HBA-D4-HBA HBA-D4-HBD HBD-D4-HBD H-D5-H H-D5-HBA H-D5-...;
18 1 2 1 22 12 8 1 2 18 6 3 1 1 1 22 13 6 5 7 2 28 9 5 1 1 1 36 16 10
3 4 1 37 10 8 1 35 10 9 3 3 1 28 7 7 4 18 16 12 5 1 2 1

```

```

FingerprintsVector;TopologicalPharmacophoreAtomPairs:FixedSize:MinDist
ance1:MaxDistance10;150;OrderedNumericalValues;ValuesString;18 0 0 1 0
0 0 2 0 0 1 0 0 0 0 22 12 8 0 0 1 2 0 0 0 0 0 0 0 0 0 18 6 3 1 0 0 0 1
0 0 1 0 0 0 0 22 13 6 0 0 5 7 0 0 2 0 0 0 0 0 28 9 5 1 0 0 0 1 0 0 1 0
0 0 0 36 16 10 0 0 3 4 0 0 1 0 0 0 0 0 37 10 8 0 0 0 0 1 0 0 0 0 0 0
0 35 10 9 0 0 3 3 0 0 1 0 0 0 0 0 28 7 7 4 0 0 0 0 0 0 0 0 0 0 0 18...

```

```

FingerprintsVector;TopologicalPharmacophoreAtomTriplets:ArbitrarySize:
MinDistance1:MaxDistance10;696;NumericalValues;IDsAndValuesString;Ar1-
Ar1-Ar1 Ar1-Ar1-H1 Ar1-Ar1-HBA1 Ar1-Ar1-HBD1 Ar1-H1-H1 Ar1-H1-HBA1 Ar1-
-H1-HBD1 Ar1-HBA1-HBD1 H1-H1-H1 H1-H1-HBA1 H1-H1-HBD1 H1-HBA1-HBA1 H1-
HBA1-HBD1 H1-HBA1-NI1 H1-HBD1-NI1 HBA1-HBA1-NI1 HBA1-HBD1-NI1 Ar1-...;
46 106 8 3 83 11 4 1 21 5 3 1 2 2 1 1 1 100 101 18 11 145 132 26 14 23
28 3 3 5 4 61 45 10 4 16 20 7 5 1 3 4 5 3 1 1 1 1 5 4 2 1 2 2 2 1 1 1
119 123 24 15 185 202 41 25 22 17 3 5 85 95 18 11 23 17 3 1 1 6 4 ...

```

```

FingerprintsVector;TopologicalPharmacophoreAtomTriplets:FixedSize:MinD
istance1:MaxDistance10;2692;OrderedNumericalValues;ValuesString;46 106
8 3 0 0 83 11 4 0 0 0 1 0 0 0 0 0 0 0 0 21 5 3 0 0 1 2 2 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 100 101 18 11 0 0 145 132 26
14 0 0 23 28 3 3 0 0 5 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 61 45 10 4 0
0 16 20 7 5 1 0 3 4 5 3 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 5 ...

```

## METHODS

new

```
$NewFingerprintsTextFileIO = new FileIO::FingerprintsTextFileIO(%IOParameters);
```

Using specified *IOParameters* names and values hash, new method creates a new object and returns a reference to a newly created FingerprintsTextFileIO object. By default, the following properties are initialized during *Read* mode:

```

Name = '';
Mode = 'Read';
Status = 0;
FingerprintsStringMode = 'AutoDetect';
FingerprintsCol = 'AutoDetect';
ColMode = 'ColNum';
CompoundIDCol = 'AutoDetect';
CompoundIDPrefix = 'Cmpd';
InDelim = 'Comma';
ValidateData = 1;
DetailLevel = 1;

```

During *Write* mode, the following properties get initialize by default:

```

FingerprintsStringMode = undef;

BitStringFormat = HexadecimalString;
BitsOrder = Ascending;

VectorStringFormat = NumericalValuesString or ValuesString;
OutDelim = 'Comma';
OutQuote = 1;

```

Examples:

```

$NewFingerprintsTextFileIO = new FileIO::FingerprintsTextFileIO(
    'Name' => 'Sample.csv',
    'Mode' => 'Read');

$NewFingerprintsTextFileIO = new FileIO::FingerprintsTextFileIO(

```

```

        'Name' => 'Sample.csv',
        'Mode' => 'Read',;
        'FingerprintsStringMode' =>
            'AutoDetect',
        'ColMode' => 'ColLabel',
        'FingerprintsCol' => 'Fingerprints',
        'CompoundIDCol' => 'CompoundID',
        'InDelim' => 'Comma');

    $NewFingerprintsTextFileIO = new FileIO::FingerprintsTextFileIO(
        'Name' => 'Sample.csv',
        'Mode' => 'Write',
        'FingerprintsStringMode' =>
            'FingerprintsBitVectorString',
        'Overwrite' => 1,
        'BitStringFormat' => 'HexadecimalString',
        'BitsOrder' => 'Ascending');

    $NewFingerprintsTextFileIO = new FileIO::FingerprintsTextFileIO(
        'Name' => 'Sample.tsv',
        'Mode' => 'Write',
        'FingerprintsStringMode' =>
            'FingerprintsVectorString',
        'Overwrite' => 1,
        'VectorStringFormat' => 'IDsAndValuesString',
        'OutDelim' => 'Tab',
        'OutQuote' => 0);

```

#### GetDataColLabels

```

    @ColLabels = $FingerprintsTextFileIO->GetDataColLabels();
    $NumOfColLabels = $FingerprintsTextFileIO->GetDataColLabels();

```

Returns an array of ColLabels from first line in text file. In scalar context, it returns number of column labels.

#### GetDataLineWords

```

    @DataWords = $FingerprintsTextFileIO->GetDataLineWords();
    $NumOfDataWords = $FingerprintsTextFileIO->GetDataLineWords();

```

Returns an array of DataWords in current data line. In scalar context, it returns number of data words.

#### GetFingerprints

```

    $FingerprintsObject = $FingerprintsTextFileIO->GetFingerprints();

```

Returns FingerprintsObject generated for current data line using fingerprints bit-vector or vector string data. The fingerprints object corresponds to any of the supported fingerprints such as PathLengthFingerprints, ExtendedConnectivity, and so on.

#### GetFingerprintsString

```

    $FingerprintsString = $FingerprintsTextFileIO->GetFingerprintsString();

```

Returns FingerprintsString for current data line.

#### IsFingerprintsDataValid

```

    $Status = $FingerprintsTextFileIO->IsFingerprintsDataValid();

```

Returns 1 or 0 based on whether FingerprintsObject is valid.

#### IsFingerprintsFileDataValid

```

    $Status = $FingerprintsTextFileIO->IsFingerprintsFileDataValid();

```

Returns 1 or 0 based on whether text file contains valid fingerprints data.

#### IsFingerprintsTextFile

```

    $Status = $FingerprintsTextFileIO->IsFingerprintsTextFile($FileName);
    $Status = FileIO::FingerprintsTextFileIO::IsFingerprintsTextFile($FileName);

```

Returns 1 or 0 based on whether *FileName* is a fingerprints text file.

#### Next or Read

```
$FingerprintsTextFileIO = $FingerprintsTextFileIO->Next();
$FingerprintsTextFileIO = $FingerprintsTextFileIO->Read();
```

Reads next available fingerprints line in text file, processes the data, generates appropriate fingerprints object, and returns FingerprintsTextFileIO. The generated fingerprints object is available using method GetFingerprints.

#### SetBitStringFormat

```
$FingerprintsTextFileIO->SetBitStringFormat($Format);
```

Sets bit string *Format* for fingerprints bit-vector string data in a text file and returns FingerprintsTextFileIO. Possible values for BitStringFormat: *BinaryString* or *HexadecimalString*.

#### SetBitsOrder

```
$FingerprintsTextFileIO->SetBitsOrder($BitsOrder);
```

Sets *BitsOrder* for fingerprints bit-vector string data in a text file and returns FingerprintsTextFileIO. Possible values for BitsOrder: *Ascending* or *Descending*.

#### SetColMode

```
$FingerprintsTextFileIO->SetColMode($ColMode);
```

Sets *ColMode* for a text file and returns FingerprintsTextFileIO. Possible values for ColMode: *ColNum* or *ColLabel*.

#### SetDataColLabels

```
$FingerprintsTextFileIO->SetDataColLabels(@ColLabels);
$FingerprintsTextFileIO->SetDataColLabels(\@ColLabels);
```

Sets *ColLabels* for a text file using an array or a reference to an array containing column labels and returns FingerprintsTextFileIO.

#### SetDataLineWords

```
$FingerprintsTextFileIO->SetDataLineWords(@LineWords);
$FingerprintsTextFileIO->SetDataLineWords(\@LineWords);
```

Sets *DataLineWords* for a text file using an array or a reference to an array containing data words and returns FingerprintsTextFileIO.

#### SetDetailLevel

```
$FingerprintsTextFileIO->SetDetailLevel($Level);
```

Sets details *Level* for generating diagnostics messages during text file processing and returns FingerprintsTextFileIO. Possible values: *Positive integers*.

#### SetFingerprints

```
$FingerprintsTextFileIO->SetFingerprints($FingerprintsObject);
```

Sets *FingerprintsObject* for current data line and returns FingerprintsTextFileIO.

#### SetFingerprintsString

```
$FingerprintsTextFileIO->SetFingerprintsString($FingerprintsString);
```

Sets *FingerprintsString* for current data line and returns FingerprintsTextFileIO.

#### SetFingerprintsStringMode

```
$FingerprintsTextFileIO->SetFingerprintsStringMode($Mode);
```

Sets *FingerprintsStringMode* for text file and returns FingerprintsTextFileIO. Possible values: *AutoDetect*, *FingerprintsBitVectorString*, or *FingerprintsVectorString*

#### SetInDelim

```
$FingerprintsTextFileIO->SetInDelim($InDelim);
```

---

Sets *InDelim* for text file and returns FingerprintsTextFileIO. Possible values: *comma, semicolon, tab*.

#### SetOutDelim

```
$FingerprintsTextFileIO->SetOutDelim($OutDelim);
```

Sets *OutDelim* for text file and returns FingerprintsTextFileIO. Possible values: *comma, semicolon, tab*.

#### SetVectorStringFormat

```
$FingerprintsTextFileIO->SetVectorStringFormat($Format);
```

Sets *VectorStringFormat* for text file and returns FingerprintsTextFileIO. Possible values: *IDsAndValuesString, IDsAndValuesPairsString, ValuesAndIDsString, ValuesAndIDsPairsString*.

#### WriteFingerprints

```
$FingerprintsTextFileIO->WriteFingerprints($FingerprintsObject,  
                                           @DataColValues);
```

Writes fingerprints string generated from *FingerprintsObject* object and other data including *DataColValues* to text file and returns FingerprintsTextFileIO.

#### WriteFingerprintsString

```
$FingerprintsSDFFileIO->WriteFingerprints($FingerprintsString,  
                                           @DataColValues);
```

Writes *FingerprintsString* and other data including *DataColValues* to text file and returns FingerprintsTextFileIO.

#### Caveats:

- o *FingerprintsStringMode, BitStringFormat, BitsOrder, VectorStringFormat* values are ignored during writing of fingerprints and it's written to the file as it is.

#### AUTHOR

Manish Sud <msud@san.rr.com>

#### SEE ALSO

FingerprintsSDFFileIO.pm, FingerprintsFPFileIO.pm

#### COPYRIGHT

Copyright (C) 2025 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.