

## FingerprintsStringUtil

## SYNOPSIS

```
use Fingerprints::FingerprintsStringUtil;

use Fingerprints::FingerprintsStringUtil qw(:all);
```

## DESCRIPTION

FingerprintsStringUtil module provides the following functions:

AreFingerprintsStringValueValid, GenerateFingerprintsBitVectorString, GenerateFingerprintsString, GenerateFingerprintsVectorString, GetDefaultBitStringFormat, GetDefaultBitsOrder, GetDefaultVectorStringFormat, GetFingerprintsStringDelimiter, GetFingerprintsStringTypeAndDescription, GetFingerprintsStringValues, ParseFingerprintsBitVectorString, ParseFingerprintsString, ParseFingerprintsVectorString

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:ArbitrarySize;10;NumericalValues;IDsAndValuesString;C.X1.B01.H3 C.X2.B02.H2 C.X2.B03.H1 C.X3.B03.H1 C.X3.B04 F.X1.B01 N.X2.B02.H1 N.X3.B03 O.X1.B01.H1 O.X1.B02;2 4 14 3 10 1 1 1 3 2
```

```
FingerprintsVector;AtomTypesCount:SLogPAtomTypes:ArbitrarySize;16;Num  
ericalValues;IDsAndValuesString;Cl 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
```

[illegible]

```
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
```

```
FingerprintsVector;EStateIndicies:FixedSize;87;OrderedNumericalValues;
ValuesString;0 0 0 0 0 0 0 3.975 0 -0.073 0 0 24.778 -2.270 0 0 -1.435
4.387 0 0 0 0 0 0 3.024 0 0 0 0 0 0 0 1.993 0 29.759 25.023 0 0 0 0 1
4.006 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
FingerprintsVector;ExtendedConnectivity:AtomicInvariantsAtomTypes:Radi
us2;60;AlphaNumericalValues:ValuesString;73555770 333564680 352413391
666191900 1001270906 1371674323 1481469939 1977749791 2006158649 21414
08799 49532520 64643108 79385615 96062769 273726379 564556671 85514103
5 906706094 988546669 1018231313 1032696425 1197507444 1331250018 1338
532734 1455473691 1607485225 1609687129 1631614296 1670251330 17303...
```

```
FingerprintsVector;ExtendedConnectivityCount;AtomicInvariantsAtomTypes
:Radius2;60;NumericalValues;IDsAndValuesString;73555770 333564680 3524
13391 666191900 1001270906 1371674323 1481469939 1977749791 2006158649
2141408799 49532520 64643108 79385615 96062769 273726379 564565671...;
3 3 2 1 1 14 1 2 10 4 3 1 1 1 2 1 2 1 1 2 3 1 1 2 1 3 3 8 2 2 2 6 2
1 2 1 1 2 1 1 1 2 1 2 1 2 1 1 1 1 1 1 1 1 1 2 1 1
```

```
FingerprintsBitVector;ExtendedConnectivityBits;AtomicInvariantsAtomTypes;Radius2;1024;BinaryString;Ascending;00000000000000000000000000000000  
0000000001010000000110000011000000000000100000000000000000000000000000000000000000000
```

```
FingerprintsVector;TopologicalAtomPairs:AtomicInvariantsAtomTypes:MinDistance1:MaxDistance10;223;NumericalValues;IDsAndValuesString;C.X1.B01.H3-D1-C.X3.B03.H1 C.X2.B02.H2-D1-C.X2.B02.H2 C.X2.B02.H2-D1-C.X3.B03.H1 C.X2.B02.H2-D1-C.X3.B04 C.X2.B02.H2-D1-N.X3.B03 C.X2.B03.H1-D1-...;
```

```
2 1 4 1 1 10 8 1 2 6 1 2 2 1 2 1 2 2 1 2 1 5 1 10 12 2 2 1 2 1 9 1 3 1
1 1 2 2 1 3 6 1 6 14 2 2 2 3 1 3 1 8 2 2 1 3 2 6 1 2 2 5 1 3 1 23 1...
```

```
FingerprintsVector;TopologicalAtomPairs:FunctionalClassAtomTypes:MinDistance:MaxDistance10;144;NumericalValues;IDsAndValuesString;Ar-D1-Ar
Ar-D1-Ar.HBA Ar-D1-HBD Ar-D1-Hal Ar-D1-None Ar.HBA-D1-None HBA-D1-NI H
BA-D1-None HBA.HBD-D1-NI HBA.HBD-D1-None HBD-D1-None NI-D1-None No...;
23 2 1 1 2 1 1 1 1 2 1 1 7 28 3 1 3 2 8 2 1 1 1 5 1 5 24 3 3 4 2 13 4
1 1 4 1 5 22 4 4 3 1 19 1 1 1 1 1 2 2 3 1 1 8 25 4 5 2 3 1 26 1 4 1 ...
```

```
FingerprintsVector;TopologicalAtomTorsions:AtomicInvariantsAtomTypes;3
3;NumericalValues;IDsAndValuesString;C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-
C.X3.BO4 C.X1.BO1.H3-C.X3.BO3.H1-C.X3.BO4-N.X3.BO3 C.X2.BO2.H2-C.X2.BO
2.H2-C.X3.BO3.H1-C.X2.BO2.H2 C.X2.BO2.H2-C.X2.BO2.H2-C.X3.BO3.H1-O...;
2 2 1 1 2 2 1 1 3 4 4 8 4 2 2 6 2 2 1 2 1 1 2 1 1 2 6 2 4 2 1 3 1
```

```
FingerprintsVector;TopologicalAtomTorsions:EStateAtomTypes;36;NumericalValues;IDsAndValuesString;aaCH-aaCH-aaCH-aaCH aaCH-aaCH-aaCH-aasC aaC
H-aaCH-aasC-aaCH aaCH-aaCH-aasC-aasC aaCH-aaCH-aasC-sF aaCH-aaCH-aasC-
ssNH aaCH-aasC-aasC-aasC aaCH-aasC-aasC-aasN aaCH-aasC-ssNH-dssC a...;
4 4 8 4 2 2 6 2 2 2 4 3 2 1 3 3 2 2 2 1 2 1 1 1 2 1 1 1 1 1 1 2 1 1 2
```

```
FingerprintsVector;TopologicalAtomTriplets:AtomicInvariantsAtomTypes:M
inDistance:MaxDistance10;3096;NumericalValues;IDsAndValuesString;C.X1
.BO1.H3-D1-C.X1.BO1.H3-D1-C.X3.BO3.H1-D2 C.X1.BO1.H3-D1-C.X2.BO2.H2-D1
0-C.X3.BO4-D9 C.X1.BO1.H3-D1-C.X2.BO2.H2-D3-N.X3.BO3-D4 C.X1.BO1.H3-D1
-C.X2.BO2.H2-D4-C.X2.BO2.H2-D5 C.X1.BO1.H3-D1-C.X2.BO2.H2-D6-C.X3...;
1 2 2 2 2 2 2 8 4 8 4 4 2 2 2 2 4 2 2 2 4 2 2 2 2 1 2 2 4 4 4 2 2
2 4 4 4 8 4 4 2 4 4 4 2 4 4 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 8...
```

```
FingerprintsVector;TopologicalAtomTriplets:SYBYLAtomTypes:MinDistance:MaxDistance10;2332;NumericalValues;IDsAndValuesString;C.2-D1-C.2-D9-C
.3-D10 C.2-D1-C.2-D9-C.ar-D10 C.2-D1-C.3-D1-C.3-D2 C.2-D1-C.3-D10-C.3-
D9 C.2-D1-C.3-D2-C.3-D3 C.2-D1-C.3-D2-C.ar-D3 C.2-D1-C.3-D3-C.3-D4 C.2
-D1-C.3-D3-N.ar-D4 C.2-D1-C.3-D3-O.3-D2 C.2-D1-C.3-D4-C.3-D5 C.2-D1-C.
3-D5-C.3-D6 C.2-D1-C.3-D5-O.3-D4 C.2-D1-C.3-D6-C.3-D7 C.2-D1-C.3-D7...
```

```
FingerprintsVector;TopologicalPharmacophoreAtomPairs:ArbitrarySize:MinDistance: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:MinDistance: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 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:MinDistance: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 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:MinDistance: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 0 100 101 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 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 1 1 1 0 0 5 ...
```

## FUNCTIONS

## AreFingerprintsStringValuesValid

```
$Status = AreFingerprintsStringValuesValid($FPString);
```

Returns 0 or 1 based on whether *FingerprintsString* contains valid values.

## GetDefaultBitStringFormat

```
$BitStringFormat = GetDefaultBitStringFormat();
```

Returns default BitStringFormat for fingerprints bit-vector strings.

## GetDefaultBitsOrder

```
$BitsOrder = GetDefaultBitsOrder();
```

Returns default BitsOrder for fingerprints bit-vector fingerprints.

## GetDefaultVectorStringFormat

```
$StringFormat = GetDefaultVectorStringFormat();
```

Returns default VectorStringFormat for fingerprints vector strings.

## GetFingerprintsStringDelimiter

```
$Delimiter = GetFingerprintsStringDelimiter();
```

Returns string Delimiter used to generate fingerprints bit-vector and vector strings.

## GenerateFingerprintsBitVectorString

```
$FPString = GenerateFingerprintsBitVectorString($FPBitVectorObject,
[$BitStringFormat, $BitsOrder]);
```

Returns a FingerprintsString generated using *FingerprintsBitVectorObject* and optionally specified *BitStringFormat* and *BitsOrder* values.

Possible *BitStringFormat* values: *BinaryString*, *Binary*, *Bin*, *HexadecimalString*, *Hexadecimal*, or *Hex*. Default *BitStringFormat* value: *BinaryString*.

Possible *BitsOrder* values: *Ascending* or *Descending*. Default *BitsOrder* value: *Ascending*.

## GenerateFingerprintsVectorString

```
$FPString = GenerateFingerprintsVectorString($FPVectorObject,
[$VectorStringFormat]);
```

Returns a FingerprintsString generated using *FingerprintsVectorObject* and optionally specified *VectorStringFormat*.

Possible *VectorStringFormat* values: *IDsAndValuesString*, *IDsAndValues*, *IDsAndValuesPairsString*, *IDsAndValuesPairs*, *ValuesAndIDsString*, *ValuesAndIDs*, *ValuesAndIDsPairsString*, *ValuesAndIDsPairs*, *ValuesString*, *Values*.

Default *VectorStringFormat* value: for *NumericalValues FPVectorType* - *IDsAndValuesString*; for all other *FPVectorTypes* - *ValuesString*.

## GenerateFingerprintsString

```
$FPString = GenerateFingerprintsBitVectorString($FPBitVectorObject,
[$BitStringFormat, $BitsOrder]);
```

```
$FPString = GenerateFingerprintsVectorString($FPVectorObject,
[$VectorStringFormat]);
```

Returns a FingerprintsString generated using *FingerprintsBitVectorObject* or *FingerprintsVectorObject* and optionally specified parameters.

## GetFingerprintsStringTypeAndDescription

```
($FPType, $FPDescription) = GetFingerprintsStringTypeAndDescription(
$FPString);
```

Returns FingerprintsStringType and *FingerprintsStringDescription* strings for FingerprintsString corresponding to FingerprintsBitVectorObject or FingerprintsVectorObject.

## GetFingerprintsStringValues

```
@FPStringValues = GetFingerprintsStringValues($FPString);
```

Parses FingerprintsString corresponding to FingerprintsBitVectorObject or FingerprintsVectorObject and returns its individual component values as an array.

### ParseFingerprintsBitVectorString

```
$FPBitVectorObject = ParseFingerprintsBitVectorString($FPBitVectorString,  
[ $ValidateValues ] );
```

Returns FingerprintsBitVectorObject generated by parsing *FingerprintsBitVectorString* with optional validation of its component values.

### ParseFingerprintsString

```
$FPBitVectorObject = ParseFingerprintsBitVectorString($FPBitVectorString,  
[ $ValidateValues ] );
```

```
$FPVectorObject = ParseFingerprintsVectorString($FPVectorString,  
[ $ValidateValues ] );
```

Returns FingerprintsBitVectorObject or ***FingerprintsVectorObject*** generated by parsing *FingerprintsBitVectorString* or *FingerprintsVectorString* with optional validation of its component values.

### ParseFingerprintsVectorString

```
$FPVectorObject = ParseFingerprintsVectorString($FPVectorString,  
[ $ValidateValues ] );
```

Returns FingerprintsVectorObject generated by parsing *FingerprintsVectorString* with optional validation of its component values.

### AUTHOR

Manish Sud <msud@san.rr.com>

### SEE ALSO

BitVector.pm, FingerprintsBitVector.pm, FingerprintsVector.pm, Vector.pm

### COPYRIGHT

Copyright (C) 2015 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.