Daylight Chemical Information Systems, Inc.
120 Vantis - Suite 550 - Aliso Viejo, CA 92656
tel +1 949-831-9990 - fax +1 949-831-9902 - www.daylight.com

Daylight Conversion Manual

Table of Contents

Daylight Conversion Package Manual

Daylight Version 4.9
Release Date 08/01/11

1. Introduction

The Daylight Conversion Package also know as "Convert" allows the interchange of data and structures between MDL chemical table-based file v2000 formats (molfile, SDfile, RGfile, rxnfile and RDfile only) and Daylight SMILES-based formats [SMILES (SMI), isomeric SMILES (ISM), SMARTS (SMA), SMIRKS (SMRK), THOR Data Tree (TDT), and SQL loader (SQLLDR) files]. The Convert program can be run either from the command-line or from within Oracle using the DayCart dayconvert function.

Users should be aware that the MDL representation is predicated on accurate knowledge of the bond orders in a particular valence bond representation. Implicit hydrogen atoms are added to make up a valid valence value. These data are stored in their proprietary P-table. Daylight relies on having accurate hydrogen-counts for most atom types and had a small list of built-in valences which can be overridden. On very rare occasions it is possible for the output valences to be incorrect during conversion. Users are encouraged to compare molecular formulae generated from MDL formatted files against those generated from the SMILES where large critical conversions are carried out.

In addition, it should be noted that many of the program options allow the user to customize the resulting conversion in such a way as to follow particular business rules. For example, a user-defined P-table can be substituted for the default table or one can require that the input file have explicit hydrogen atoms on stereocenters.

Lastly, it should be noted that Conversion Package versions of mol2smi and smi2mol are not intented to be direct replacements for the older contrib versions of programs with the same name. Both options and conversion behavior differs.

2. Prerequisites

2.1 Programming Experience

Users of this manual should possess general UNIX skills as well as general knowledge of Daylight software. DayCart users should have general knowledge of Oracle.

2.2 Software Requirements

Convert is included with the standard Daylight distribution (versions 4.91 or later). The standard distribution is available for download from Daylight's web site (http://www.daylight.com). In order to use the Convert

program, an appropriate program license from Daylight is required. Within Oracle, DayCart must be licensed to call the dayconvert function.

3. Data Conventions

Convert will interconvert data and structures between MDL chemical connection table v2000 formats (see CTFile Formats June 2005 document from the MDL website --- www.mdli.com for a detailed description) and Daylight SMILES-based formats.

Convert does not recognize XDfiles or Sgroup features. Lastly, the MDL v3000 format is not supported. Specific input and/or output formats are as follows:

- SMILES: Any legitimate molecule/reaction(read and write)
- Isomeric SMILES: Any legitimate molecule/reaction with stereochemistry and isotopic features as appropriate (read and write)
- SMARTS: Any legitimate molecule/reaction query (write only)
- SMIRKS: Any legitimate transformation (write only)
- TDT: Any legitimate molecule/reaction TDT rooted in $SMI and potentially branched on another identifier (read and write); Any legitimate non-structure, non-branched TDT rooted on a non-$SMI identifier (read and write); Any legitimate non-branched molecule/reaction query or transformation TDT rooted in a non-$SMI with SMA or SMIRK tags (write only)
- molfile/SDfile/RGfile: Any legitimate molecule or query that contains only those features that can be represented in both MDL and SMILES or SMARTS formats(read and write non-queries; read only queries)
- rxnfile/RDfile: Any legitimate molecule/reaction/molecule query/reaction query/transformation that contains only those features that can be represented in both MDL and SMILES, SMARTS or SMIRKS formats (read and write non-queries; read only queries and transformations)
- SQLLDR: Specifically formatted data and SMILES-based structure files for loading with SQL loader within Oracle

See the Appendix for examples of the different formats.

3.1 MDL to Daylight

The following conventions are used in converting from MDL format (molfile, rxnfile, SDfile, RGfile and RDfile) to Daylight format (SMILES, isomeric SMILES, SMARTS, SMIRKS, TDT, and SQLLDR):

- Unless otherwise specified, the information on the first line of the header block of each connection table for molecules or the $RIREG value for reactions is considered to be a unique ID. If there is no ID, then the actual isomeric SMILES, SMARTS or SMIRKS generated is used as the ID.
- When converting to SMILES, isomeric SMILES, SMARTS or SMIRKS formats, the unique ID is written after the SMILES, isomeric SMILES, SMARTS or SMIRKS, e.g, c1ccccc1 BENZENE or [C,Br][C;v4] QUERY1.
- For TDT output the unique ID is written as $NAM and for SQLLDR output the unique ID appears on the first line of each record.
- Data in SDfiles/RDfiles are converted for TDT and SQLLDR outputs using the input data identifiers. Legal characters for data tags are limited to: $, _, /, A-Z, a-z, and 0-9.
- A special $SMIG datatype containing the conversion program name and version is added to TDT output files.

- The SQLLDR format is only be available for the command line version of convert stores data in one file (.dat) and structural information in another (.str). If there is no data, no .dat file is written.
- An entry in an MDL file without structural information can be converted to a TDT rooted in $NAM but no entry is written to a SMILES, isomeric SMILES or SQLLDR .str output file. Note that output TDTs for SMARTS and SMIRKS are also rooted in $NAM.

Standard order for an output TDT:

$SMI - unique SMILES
2D or $D3D - 2D or 3D coordinates associated with SMILES
$NAM - name
LINE1 - data on line 1 of the CTAB (molecules only)
ISM/SMA/SMRK - isomeric SMILES, SMARTS or SMIRKS
CFLAG - chirality flag
VIS - visible atoms
VAL - valence
RXNR - reaction component type (reactions only)
RXNC - reaction component number (reactions only)
RXNM - reaction atom-atom mapping number (reactions only)
RXNE - reaction exact change flag (reactions only)
RXNI - reaction inversion/retention flag (reactions only)
ASYM - atom symbol
BST - bond style
2DI or 3DI - 2D or 3D coordinates associated with isomeric SMILES, SMARTS or SMIRKS
MIREG or RIREG - identifiers(RDfiles only)
MEREG or REREG - identifiers (RDfiles only)
PREFIX - prefix for $DTYPE (RDfiles only)
Data field identifier - additional data in the same order as the input
Standard SQLLDR .dat output format:

Unique ID; (RDfiles only)
MIREG or RIREG; (RDfiles only)
MIREG or RIREG value (RDfiles only)
Unique ID; (only included if prefix is specified, RDfiles only)
PREFIX; (only included if prefix is specified, RDfiles only)
PREFIX value (only included if prefix is specified, RDfiles only)
Unique ID;
Data field identifier;
Data value
Standard SQLLDR .str output format:

Unique ID;
Information on line1 of CTAB; (molecules only)
Unique SMILES;
2D coordinates associated with SMILES;
3D coordinates associated with SMILES;
Isomeric SMILES, SMARTS or SMIRKS;
Chirality flag;
Visible atoms;
Valence;
Reaction component type; (reactions only)

Reaction component number; reactions only)
Reaction atom-atom mapping number; (reactions only)
Reaction exact change flag; (reactions only)
Reaction inversion/retention flag; (reactions only)
Atom symbol;
Bond style;
2D coordinates associated with isomeric SMILES, SMARTS or SMIRKS;
3D coordinates associated with isomeric SMILES, SMARTS or SMIRKS
Note: Example scripts for loading SQLLDR output files can be found in
$DY_ROOT/contrib/src/oracle/dayconvert, e.g., load_conv_demos.sh, mol.ctl, and mol.par.

3.2 Daylight to MDL

The following conventions will be used in converting from Daylight format (SMILES and TDT) to MDL format (SDfile and RDfile):

- The space-delimited string following the SMILES in a SMILES input file is considered to be a unique ID. If there is no unique identifier supplied then the isomeric SMILES will be used as the ID. Note that if there is no structure, then the ID must be preceded by a space. The ID will be written to line 1 of the header block for all output molecules (SDfile and RDfile formats), to MEREG for output molecules (RDfile format only) and as REREG for reactions (RDfile format only).
- Unless otherwise specified, $NAM or the root identifier is used as the unique ID from an input TDT file. When converting a TDT containing a molecule, the information in LINE1, if available, is written to line 1 of the header block of the MDL output file otherwise the default ($NAM) or specified ID (see options) is used. In addition, the ID is used as the value for MEREG or REREG for RDfile output.
- Branched input TDTS rooted in $SMI is split on $NAM unless another field is specified (see options).
- Data in TDT files is converted for SDfile and RDfile output with the datatype name as the data identifier. However, the special meaning of '$' is lost.
- If the input file does not have coordinates, then 2D coordinates are generated using the Daylight depict algorithm.
- The format for MDL output files meets the standard described in the CTFile Formats June 2005 document with the following exception: convert can read but not write the MDL RAD lines to the properties block. When writing to MDL format, radical information is captured in the valence field of the atom block.
- Although the MDL standard does require the addition of an 'M END' line under certain circumstances, convert always adds this line to output MDL files.

3.3 P-table

Accurate conversion of MDL files relies on adding implicit hydrogen atoms to make up a valid valence value. Therefore, MDL to Daylight conversions require the use of a Daylight p-table that lists acceptable charge-valence pairs to be used in the calculation of hydrogen-counts. This table is populated with default information related to atom number, symbol, common isotope, and valence-charge pairs. An example table (ptable.dat) is provided in $DY_ROOT/data. The user can modify the default table as described in Sections 4.1 Command Line and 4.2 DayCart.

3.4 SMARTS and SMIRKS

The following are special handling conventions when converting to SMARTS or SMIRKS:

- Input transformations with query bonds cannot be converted to SMIRKS.
- Input transformations with query atoms that do not have the query atoms mapped cannot be converted to SMIRKS.
- Before a file containing SMIRKS is written out, the SMIRKS is checked to confirm that it can be compiled. Otherwise, an error message is given.

4. General Usage

Convert consists of a set of programs that encompasses the following set of functions:

- Convert molfiles/SDfiles containing molecules and data into SMILES, isomeric SMILES, TDT or SQLLDR files
- Convert molfiles/SDfiles containing molecule queries* and data into SMARTS, TDT or SQLLDR files
- Convert rxnfiles/RDfiles containing molecules, reactions and data into SMILES, isomeric SMILES, TDT or SQLLDR files.
- Convert rxnfiles/RDfiles containing reaction queries* and data into SMARTS, TDT or SQLLDR files
- Convert rxnfiles/RDfiles containing transformations* and data into SMIRKS, TDT or SQLLDR files
- Convert SMILES/TDT files containing molecules and data into SDfiles
- Convert SMILES/TDT files containing reactions and data into RDfiles

The goal of each program is to convert structures from one format to the other while retaining all data and depiction information if the output format permits. Atomic features such as mass difference, charge, radical state, valence, stereochemistry, implicit hydrogen atom counts and atom-atom mapping and bond features such as bond style, topology, and multiple bond types for queries are recognized and converted. For isomeric SMILES, SMARTS, SMIRKS, TDT and SQLLDR output from MDL input, double bond stereochemistry and tetrahedral chirality is inferred from the atom coordinates and bond style information in the input connection table. For SDfile and RDfile output, stereochemistry is inferred from the isomeric SMILES and appropriate MDL bond styles are set to reflect this. While these conversions follow MDL rules, some users may have conventions that allow a less rigorous depiction of chirality. The chirality flag is only be set in the output, if it is set in the input TDT.

MDL query features that are supported include: H-counts (atom block); bond types and bond topology (bond block); ring bond count, substitution count, unsaturated atom count and atom list (property block). In addition, although the MDL exact change flag (atom block) and reacting center status (bond block) features are not used in SMILES, the information is stored in TDT and SQLLDR output when possible for future back conversion. In contrast, the following MDL query features are not supported: stereo care box and inversion/retention flag (atom block) and link atom (property block).

In general, recognition of query features in an input SDfile or RDfile will be restricted to those functions (marked with * in the list above) that specifically handle query input. However, in order to facilitate ease of use certain unambiguous query features are allowed when calling the other MDL to Daylight functions, i.e., hydrogen count and aromatic bond type. Certain atom symbols (X, R, *) are also permitted in non-queries when converting from MDL to Daylight. In general, the algorithm permits atom symbol information to be carried in appropriate Daylight formats (TDT, SQLLDR). Attempting to read any other query features will result in an ERROR being generated.

4.1 Command Line

The command line version of convert consists of a set of case-insensitive specific programs as summarized below:

MOL2SMI	input = molfile/SDfile	output = SMI/ISM/TDT/SQLLDR
RD2SMI	input = rxnfile/RDfile	output = SMI/ISM/TDT/SQLLDR
SD2SMARTS	input = molfile/SDfile/RGfile	output = SMA/TDT/SQLLDR
RD2SMARTS	input = rxnfile/RDfile	output = SMA/TDT/SQLLDR
RD2SMIRKS	input = rxnfile/RDfile	output = SMRK/TDT/SQLLDR
SMI2MOL	input = SMI or TDT	output = SDfile
SMI2RD	input = SMI/TDT	output = RDfile

Note: Output from SMI2MOL and SMI2RD will always be in SDfile or RDfile format rather than molfile or rxnfile format even if there is no associated data. In addition, the input files for SD2SMARTS may be molecules or molecule queries, for RD2SMARTS may be reactions or reaction queries and for RD2SMIRKS may be reactions or transforms.

The general usage is as follows where program_name is MOL2SMI, SMI2MOL, etc.:

$ **`program_name [options] [infile [outfile]]`** - where program_name is mol2smi, smi2mol, etc.

Default input is to stdin and default output to stdout. If the output format is to be SQLLDR, then a value for the outfile is required. This will be used as the rootname for the .dat and .str files.

If a file argument is provided, it will be used as input. If two arguments are provided, they will be used as input and output. Note: if an output file is to be specified, but input from stdin is desired, the input file must be designated as '-' (dash), e.g.,

```
$ gzip -cd infile.gz | program_name [options] [infile [outfile]]
$ cat infile | program_name [options] [- [outfile]]
```

Note: The user will not have access to the default p-table but can call a user-defined p-table as an option. The user can edit a copy of the example ptable.dat located in $DY_ROOT/data by uncommenting one or more lines with desired changes and editing the valence-charge pair values. For example, changing from #6 C 12 3,-1,4,0,3,1 to #6 C 12 4,0 would remove the ability to recognize charge (-1 or +1) carbons when the user-provided ptable is called.

Default values for a series of general and specific options are available (see 5.1 General Options and 5.2 Function Specific Options) and will be modifiable on a per conversion basis from the command line by providing the option name and new_value. Permanent modification of default values will be allowed by editing the $DY_ROOT/etc/unix/convert.dat file.

4.2 DayCart

Convert is accessible through DayCart by using the case-insensitive dayconvert function or operator either directly using SQLPLUS or by running an SQL file with the appropriate information (see $DY_ROOT/contrib/src/oracle/dayconvert for examples). Data input is an actual string such as 'CC(=C)CC' or a COLUMN_NAME from a specific TABLE_NAME.

SQL **SELECT DDPACKAGE.FDAYCONVERT(COLUMN_NAME,'IFMT','OFMT',TYPE,CLASS) FROM TABLE_NAME;**

SQL **SELECT DAYCONVERT('STRING','IFMT','OFMT',TYPE,CLASS) FROM DUAL;**

The IFMT and OFMT parameters are used to designate the input and output formats, respectively. Both parameters need to be identified by a particular letter sequence (SMI, ISM, SMAK, SMRK, TDT, TDTSMA, TDTSMRK, MOL, SDF OR RDF). Valid combinations of input and output formats for conversion are as follows where TDTSMA and TDTSMRK are the TDT versions with SMARTS and SMIRKS, respectively:

- SMI ---> SDF/RDF
- TDT ---> SDF/RDF
- SDF ---> SMI/ISM/SMA/TDT/TDTSMA
- RDF ---> SMI/ISM/SMA/SMRK/TDT/TDTSMA/TDTSMRK

Note: MOL is a valid input value that is interchangeable with SDF regardless of the actual input format. In addition, MOL is a valid output format value if the input value is SMI or TDT. However, the output will always be written in SDF format even if there is no associated data. Lastly, rxnfile format is not recognized as a separate format. Either MOL or SDF is used for RGfile input and RDF is used for both rxnfiles and RDfiles.

The 'type' parameter has been included for backwards compatibility and no longer controls the inclusion of isomeric information in the conversion output. Conversion output from a smiles form to an MDL form will always include the isomeric information while conversion from an MDL form to SMILES is controlled by the inclusion of ism versus smi for ofmt. However, inclusion of a value for the type parameter (0 or 1) is required for versions 4.92 and earlier.

SQL **SELECT DDPACKAGE.FDAYCONVERT('CCCO','SMI','SDF',1) FROM DUAL;**

In contrast for versions 4.93 and later, a value for the type parameter is only required if a ptable_class value is given.

SQL **SELECT DDPACKAGE.FDAYCONVERT('CCCO','SMI','SDF') FROM DUAL;**

OR

SQL **SELECT DDPACKAGE.FDAYCONVERT('CCCO','SMI','SDF',1, 10) FROM DUAL;**

The PTABLE-CLASS parameter is optional. When DayCart is installed, an empty table named PTABLE is created with the following columns:

AT_NO NUMBER
SYMBOL VARCHAR2(8)
AT_MASS NUMBER --- common isotope
VALENCE_CHARGE_LIST VARCHAR2(4000) --- valence-charge pairs

CLASS NUMBER This table can be filled with specific user-defined valence and charge information, e.g,

SQL **INSERT INTO PTABLE VALUES(7,'N',14,'2,-1,3,0,4,1',1);**

If the PTABLE_CLASS parameter is provided, it will indicate that the valence and charge information in the PTABLE table is to be used instead of the information provided in the default p-table. The specific information to be used is based upon the class number supplied. Note: See the example ptable (ptable.dat) located at $DY_ROOT/data for default values.

A set of dayconvert options with default values will be available (See 5.2 Function Specific Options). The user will be able to find the current value and reset it by using standard DayCart functions on a per session basis. To find the current value of an option named conv_option one would run:

SQL> **SELECT DDPACKAGE.FINFO('CONV_OPTION') FROM DUAL;**

To reset the value for conv_option one would run:

SQL> **SELECT DDPACKAGE.FSETINFO('CONV_OPTION=new_value') FROM DUAL;**

In addition, information about a failed operation will be provided through the standard DayCart fgeterrors function which will provide the error string from the toolkit error queue for the level specified (0 or 1 = something of interest but not an error, 2 = abnormal finding that may require attention, 3 = requested operation cannot be carried out, and 4 = serious error, the program cannot continue):

SQL> **SELECT DDPACKAGE.FGETERRORS(LEVEL) FROM DUAL;**

5. Options

There are a set of general options as well as a set specific to certain conversion functions. These options are case insensitive and cannot be abbreviated. Available options are detailed in the following sections.

5.1 General Options

The following general options are available for all of the convert commands for only the command line version.

`-h`
`-HELP`

Description: Print the help message. `-VERSION`

Description: Print the program version. `-DO_RECORDS [INTEGER]`

Description: Convert only a specified number of records. The default converts all records in the input file. `-SKIP_RECORDS [INTEGER]`

Description: Start conversion with a specific record in the input file. The default is to start conversion with record 1. `-ERROR_LOG [FILE_NAME]`

Description: Create an error log with all error messages for the level set and the records in which the errors occurred. The default is for no error log to be created. `-REJECT_LOG [FILE_NAME]`

Description: Create a reject log with all of the rejected records. The default is for no reject log to be created.
`-SHOW_ERROR_LEVEL [ERROR_LEVEL]`

Description: Change the error level displayed where ERROR_LEVEL is one of the following with the default level being ERROR:

ALL or NOTE --- something of interest but not an error
WARNING --- abnormal finding that may require attention
ERROR --- requested operation cannot be carried out
FATAL --- serious error, the program cannot continue

5.2 Function Specific Options

Unless otherwise indicated, the following options are available for a specific set of functions for both versions of convert. In the following list the command line option is given first followed by the DayCart option.

`-ADD_2D [TRUE|FALSE]`	MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS and RD2SMIRKS
`-ADD_3D [TRUE|FALSE]`	MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS and RD2SMIRKS
`CONV_ADD_2D [TRUE|FALSE]`	SDF/RDF ---> TDT/TDTSMA/TDTSMRK
`CONV_ADD_3D [TRUE|FALSE]`	SDF/RDF ---> TDT/TDTSMA/TDTSMRK

Description: Adds 2D or 3D coordinates to the output. These data are taken from the actual coordinates in the input atom block and stored as a comma-separated list of values in unique SMILES/SMARTS order. The default is TRUE for both. If non-zero coordinates are found in the atom block, then either 2D or 3D coordinates are written to the output file depending on which are available. Setting one of these values to FALSE eliminates the value for that set of coordinates.

Example Input:

```
3D
  smi2mol

11 11  0     0  0            999 V2000
    0.0021   -0.0041    0.0020 S   0  0  0  0  0
   -0.0196    1.5948    0.0108 C   0  0  0  0  0
    1.0517    2.4471    0.0052 N   0  0  0  0  0
   -1.4522    2.6722    0.0224 S   0  0  0  0  0
    0.7753    3.8483   -0.3526 C   0  0  0  0  0
   -0.5428    4.2414    0.4092 C   0  0  0  0  0
    1.9462    2.1437    0.2257 H   0  0  0  0  0
    1.5993    4.4865   -0.0336 H   0  0  0  0  0
    0.6251    3.9382   -1.4285 H   0  0  0  0  0
   -1.0245    5.1138   -0.0324 H   0  0  0  0  0
   -0.3795    4.3674    1.4795 H   0  0  0  0  0
  1  2  2  0  0  0
  2  3  1  0  0  0
```

```
  2  4  1  0  0  0
  3  5  1  0  0  0
  3  7  1  0  0  0
  4  6  1  0  0  0
  5  6  1  0  0  0
  5  8  1  0  0  0
  5  9  1  0  0  0
  6 10  1  0  0  0
  6 11  1  0  0  0
M  END
$$$$
```

Example Output (default):

```
$SMI<S=C1NCCS1>
$D3D<0.0;3D;0.0021,-0.0041,0.0020,-0.0196,1.5948,0.0108,1.0517,
2.4471,0.0052,0.7753,3.8483,-0.3526,-0.5428,4.2414,0.4092,-1.4522,
2.6722,0.0224,1.9462,2.1437,0.2257,1.5993,4.4865,-0.0336,0.6251,
3.9382,-1.4285,-1.0245,5.1138,-0.0324,-0.3795,4.3674,1.4795;n/a>
$NAM<3D>
LINE1<3D>
ISM<S=C1NCCS1>
CFLAG<0>
VIS<1,1,1,1,1,1,1,1,1,1>
VAL<,,,,,,,,,,>
ASYM<,,,,,,,,,,>
BST<,,,,,,,,,,>
3DI<0.0;3D;0.0021,-0.0041,0.0020,-0.0196,1.5948,0.0108,1.0517,2.4471,
0.0052,0.7753,3.8483,-0.3526,-0.5428,4.2414,0.4092,-1.4522,2.6722,
0.0224,1.9462,2.1437,0.2257,1.5993,4.4865,-0.0336,0.6251,3.9382,
-1.4285,-1.0245,5.1138,-0.0324,-0.3795,4.3674,1.4795;n/a>
|
```

Example Output (3D option set to FALSE):

```
$SMI<S=C1NCCS1>
2D<0.0021,-0.0041,-0.0196,1.5948,1.0517,2.4471,0.7753,3.8483,
-0.5428,4.2414,-1.4522,2.6722>
$NAM
```

`-CHI_EXPLICIT_H [TRUE|FALSE]` MOL2SMI and RD2SMI
`CONV_CHI_EXPLICIT_H [TRUE|FALSE]` SDF/RDF ---> ISM/TDT

Description: Determines whether chiral atoms in the input file must have explicit hydrogens. The default is FALSE. Setting this option as TRUE requires that chiral atoms have all hydrogens explicitly indicated in order to generate isomeric SMILES

Example Input:

```
chiral_without_hydrogens
 smi2mol

  6  5  0  0  0  0  0  0  0  0  1
  238.0000  191.0000    0.0000 C   0  0  0  0  0
  240.4705  158.8519    0.0000 N   0  0  0  0  0
```

```
  213.0000  191.0000    0.0000 Br  0  0  0  0  0
  269.0000  189.0000    0.0000 C   0  0  0  0  0
  280.6777  173.3223    0.0000 O   0  0  0  0  0
  286.6777  206.6777    0.0000 O   0  0  0  0  0
  1  2  1  1  0  0
  1  3  1  6  0  0
  1  4  1  6  0  0
  4  5  1  0  0  0
  4  6  2  0  0  0
M  END
$$$$
chiral_with_hydrogens
  smi2mol

  7  6  0  0  0  0  0  0  0  0  1
  238.0000  177.0000    0.0000 C   0  0  0  0  0
  244.4705  201.1481    0.0000 N   0  0  0  0  0
  244.4705  152.8519    0.0000 H   0  0  0  0  0
  213.0000  177.0000    0.0000 Br  0  0  0  0  0
  263.0000  177.0000    0.0000 C   0  0  0  0  0
  280.6777  159.3223    0.0000 O   0  0  0  0  0
  280.6777  194.6777    0.0000 O   0  0  0  0  0
  1  2  1  6  0  0
  1  3  1  6  0  0
  1  4  1  1  0  0
  1  5  1  1  0  0
  5  6  2  0  0  0
  5  7  1  0  0  0
M  END
$$$$
```

Example Output (default):

```
N[C@H](Br)C(=O)O chiral_without_hydrogens
N[C@H](Br)C(=O)O chiral_with_hydrogens
```

Example Output (option set to TRUE):

```
NC(Br)C(=O)O chiral_without_hydrogens
N[C@H](Br)C(=O)O chiral_with_hydrogens
```

`-DAYLIGHT_LIKE [TRUE|FALSE]` `---`	SD2SMARTS and RD2SMARTS No DayCart equivalent

Description: Sets all DAYLIGHT options (DAYLIGHT_HCOUNT, DAYLIGHT_STEREO, and DAYLIGHT_CHI_H) to TRUE or FALSE. The default is TRUE.

Examples: See examples for individual DAYLIGHT options

`-DAYLIGHT_CHI_H [TRUE|FALSE]` `CONV_DAY_CHIH [TRUE|FALSE]`	SD2SMARTS, RD2SMARTS and RD2SMIRKS SDF/RDF ---> SMA/SMRK/TDTSMA/TDTSMRK

Description: Determines whether chiral atoms in the input file must have explicit hydrogens. The default is TRUE. Setting this option to FALSE removes the requirement that chiral atoms have all hydrogens explicitly indicated.

Example Input (query):

```
chiral_without_hydrogens
 DCIS

  6  5  0  0  0  0  0  0  0  0  1
  238.0000  191.0000    0.0000 C   0  0  0  0  0
  240.4705  158.8519    0.0000 N   0  0  0  0  0
  213.0000  191.0000    0.0000 L   0  0  0  0  0
  269.0000  189.0000    0.0000 C   0  0  0  0  0
  280.6777  173.3223    0.0000 O   0  0  0  0  0
  286.6777  206.6777    0.0000 O   0  0  0  0  0
  1  2  1  1  0  0
  1  3  1  6  0  0
  1  4  1  6  0  0
  4  5  1  0  0  0
  4  6  2  0  0  0
M  ALS   3  2 F    F   I
M  END
$$$$
chiral_with_hydrogens
  DCIS

  7  6  0  0  0  0  0  0  0  0  1
  238.0000  177.0000    0.0000 C   0  0  0  0  0
  244.4705  201.1481    0.0000 N   0  0  0  0  0
  244.4705  152.8519    0.0000 H   0  0  0  0  0
  213.0000  177.0000    0.0000 L   0  0  0  0  0
  263.0000  177.0000    0.0000 C   0  0  0  0  0
  280.6777  159.3223    0.0000 O   0  0  0  0  0
  280.6777  194.6777    0.0000 O   0  0  0  0  0
  1  2  1  6  0  0
  1  3  1  6  0  0
  1  4  1  1  0  0
  1  5  1  1  0  0
  5  6  2  0  0  0
  5  7  1  0  0  0
M  ALS   4  2 F    F   I
M  END
$$$$
```

Example Output (default):

```
[#9,#53]C([N,n])C(=[O,o])[O,o] chiral_without_hydrogens
[#9,#53][C;@@;!H0]([N,n])[C] chiral_with_hydrogens
```

Example Output (option set to FALSE):

```
[#9,#53][C;@@;!H0]([N,n])[C](=[O,o])[O,o] chiral_without_hydrogens
[#9,#53][C;@@;!H0]([N,n])[C](=[O,o])[O,o] chiral_with_hydrogens
```

`-DAYLIGHT_HCOUNT [TRUE|FALSE]`
`CONV_DAY_HCOUNT [TRUE|FALSE]`

SD2SMARTS and RD2SMARTS
MOL/SDF/RDF --->
SMA/TDTSMA

Description: Determines whether both explicit H and H-count field are used. The default is TRUE. If this option is set to FALSE then the H-count field is not used.

Example Input (query):

```
explicitH_Hcount
  DCIS

  7  6  0  0  0  0  0  0  0  0999 V2000
   -2.0958    0.5917    0.0000 N   0  0  0  3  0  0  0  0  0  0  0  0
   -1.3000    0.8083    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5875    0.3958    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.2083    0.6125    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3750   -0.4000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -1.7167    1.5250    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
    0.4208   -0.6125    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
  3  4  5  0  0  0  0
  3  5  1  0  0  0  0
  2  3  1  0  0  0  0
  2  6  1  6  0  0  0
  1  2  1  0  0  0  0
  5  7  1  0  0  0  0
M  END
$$$$
```

Example Output (default):

```
[N,n;!H0;!H1][C;!H0]C([O;!H0])-,=[O,o] explicitH_Hcount
```

Example Output (option set to FALSE):

```
[O;!H0]C(-,=[O,o])[C;!H0][N,n] explicitH_Hcount
```

`-DAYLIGHT_STEREO [TRUE|FALSE]`
`CONV_DAY_STEREO [TRUE|FALSE]`

SD2SMARTS and RD2SMIRKS
SDF/RDF --->
SMA/SMRK/TDTSMA/TDTSMRK

Description: Determines whether only specified stereochemistry is used. The default is TRUE. When this option is set to FALSE then specified and unspecified stereochemistry is used.

Example Input (query):

```
specified_unspecified_stereo
  DCIS

  5  4  0  0  0  0  0  0  0  0999 V2000
   -3.1875   -0.4333    0.0000 *   0  0  0  0  0  0  0  0  0  0  0  0
   -2.4750   -0.0167    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.7625   -0.4292    0.0000 F   0  0  0  0  0  0  0  0  0  0  0  0
   -2.0667    0.7000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
```

```
   -3.0625    0.5708    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
  2  3  1  0  0  0  0
  1  2  1  0  0  0  0
  2  4  1  1  0  0  0
  2  5  1  6  0  0  0
M  END
$$$$
```

Example Output (default):

```
[*][C;@]([C,c])([N,n])[F] specified_unspecified_stereo
```

Example Output (option set to FALSE):

```
[*][C;@?]([C,c])([N,n])[F] specified_unspecified_stereo
```

`-DB_EXPLICIT_H [TRUE|FALSE]`
`CONV_DB_EXPLICIT_H [TRUE|FALSE]`

MOL2SMI and RD2SMI
SDF/RDF ---> ISM/TDT

Description: Determines whether double bonds in the input file must have explicit hydrogens. The default is FALSE. Setting this option as TRUE requires that double bonds have all hydrogens explicitly indicated in order to generate isomeric SMILES.

Example Input:

```
DB_without_hydrogens
 smi2mol

  6  5  0  0  0  0  0  0  0  0  1
  170.0000  206.0000    0.0000 C   0  0  0  0  0
  195.0000  206.0000    0.0000 C   0  0  0  0  0
  212.6777  223.6777    0.0000 C   0  0  0  0  0
  157.5000  184.3494    0.0000 Br  0  0  0  0  0
  236.8258  217.2072    0.0000 C   0  0  0  0  0
  258.4765  229.7072    0.0000 C   0  0  0  0  0
  1  2  2  0  0  0
  1  4  1  0  0  0
  2  3  1  0  0  0
  3  5  1  0  0  0
  5  6  1  0  0  0
M  END
$$$$
DB_with_hydrogens
 smi2mol

  8  7  0  0  0  0  0  0  0  0  1
  170.0000  206.0000    0.0000 C   0  0  0  0  0
  195.0000  206.0000    0.0000 C   0  0  0  0  0
  212.6777  223.6777    0.0000 C   0  0  0  0  0
  212.6777  188.3223    0.0000 H   0  0  0  0  0
  152.3223  223.6777    0.0000 H   0  0  0  0  0
  157.5000  184.3494    0.0000 Br  0  0  0  0  0
  236.8258  217.2072    0.0000 C   0  0  0  0  0
  258.4765  229.7072    0.0000 C   0  0  0  0  0
  1  2  2  0  0  0
```

```
  1  5  1  0  0  0
  1  6  1  0  0  0
  2  3  1  0  0  0
  2  4  1  0  0  0
  3  7  1  0  0  0
  7  8  1  0  0  0
M  END
$$$$
```

Example Output (default):

```
CCC/C=C/Br DB_without_hydrogens
CCC/C=C/Br DB_with_hydrogens
```

Example Output (option set to TRUE):

```
CCCC=CBr DB_without_hydrogens
CCC/C=C/Br DB_with_hydrogens
```

`-DB_RING_CISTRANS [TRUE|FALSE]`
`CONV_RING_CISTRANS [TRUE|FALSE]`

MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS AND RD2SMIRKS
SDF/RDF ---> ISM/TDT/SMA/SMRK/TDTSMA/TDTSMRK

Description: Toggles whether stereochemistry for ring double bonds is indicated. Default is FALSE. Setting this option as TRUE, marks the cis/trans stereochemistry for all ring double bonds when generating isomeric SMILES, SMARTS or SMIRKS.

Example Input:

```
ring_cis_trans
  smi2mol

  9  9  0  0  0  0  0  0  0  0  1
  152.6517  225.9992    0.0000 C   0  0  0  0  0
  174.3018  213.4992    0.0000 C   0  0  0  0  0
  174.3018  188.4991    0.0000 C   0  0  0  0  0
  152.6517  175.9991    0.0000 C   0  0  0  0  0
  131.0006  188.4991    0.0000 C   0  0  0  0  0
  131.0006  213.4992    0.0000 C   0  0  0  0  0
  195.9524  225.9992    0.0000 Br  0  0  0  0  0
  195.9524  175.9991    0.0000 F   0  0  0  0  0
  109.3500  175.9991    0.0000 C   0  0  0  0  0
  1  2  1  0  0  0
  6  1  1  0  0  0
  2  3  2  0  0  0
  2  7  1  0  0  0
  3  4  1  0  0  0
  3  8  1  0  0  0
  4  5  1  0  0  0
  5  6  1  0  0  0
  5  9  1  0  0  0
M  END
$$$$
```

Example Output (default):

```
CC1CCC(=C(F)C1)Br ring_cis_trans
```

Example Output (option set to TRUE):

```
CC1CC/C(=C(/F)\C1)/Br ring_cis_trans
```

`-FIX_RADICAL_RINGS [TRUE|FALSE]`
`CONV_FIX_RADICAL_RINGS [TRUE|FALSE]`

MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS AND RD2SMIRKS
SDF/RDF ---> SMI/ISM/SMA/SMRK/TDT/TDTSMA/TDTSMRK

Description: Converts radical rings to aromatic. The default is TRUE which allows for the certain types of five, six, and seven-membered radical rings to be converted to aromatic. Changing this option to FALSE, keeps the rings as specified in the input file. In order for a ring to be converted, all atoms in the ring must be carbon and designated as doublet radicals. In addition, no atom in the ring may have a charge.

Example Input:

```
radical_ring
  smi2rd

  5  5  0  0  0  0  0  0  0  0999 V2000
    0.5050    4.0800    0.0000 C   0  4  0  0  0  0  0  0  0  0  0  0
   -0.5050    4.0800    0.0000 C   0  4  0  0  0  0  0  0  0  0  0  0
   -0.8150    3.1300    0.0000 C   0  4  0  0  0  0  0  0  0  0  0  0
   -0.0050    2.5400    0.0000 C   0  4  0  0  0  0  0  0  0  0  0  0
    0.8250    3.1200    0.0000 C   0  4  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  1  5  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  5  4  1  0  0  0  0
M  RAD  1   2   2   2   3   2   4   2   5   2
M  END
$$$$
```

Example Output (default):

```
c1cc[cH-]c1 CC34 radical_ring
```

Example Output (option set to FALSE):

```
[CH]1[CH][CH][CH][CH]1 radical_ring
```

`-ID_FIELD [FIELD_NAME]`
`CONV_ID_FIELD [FIELD_NAME]`

MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS AND RD2SMIRKS
SDF/RDF ---> SMI/ISM/SMA/SMRK/TDT/TDTSMA/TDTSMRK

Description: Sets the data field identifier to be used as a unique ID. For molfile/SDfile input, the

default for ID name is the first line of each header block. If there is no ID on line 1, the isomeric SMILES/SMARTS is used. Alternatively, designating a data field identifier as the ID_FIELD causes the data in that field to be used as the ID. For RDfile input, the default ID name for a molecule is the first line of each header block. Designating all or part of a data field identifier including MIREG or MEREG as the named ID_FIELD causes the data in that field to be used as the ID. For reactions, the value for $RIREG or this value followed by '_' (underscore) and the RXN:VARIATION number if specified as the -PREFIX is used as the default. Designating all or part of a datatype including REREG as the ID_FIELD causes the data in that field to be recognized as the ID. Note one may need to place the data field identifier in quotes and use `\` before '$'. Input records not containing information in the designated field are rejected.

Example Input:

```
000095
  smi2mol

  9  8  0  0  0  0  0  0  0  0999 V2000
    7.1962    1.2500    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    6.3301    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.4641    0.2500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    4.5981    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.7321    0.2500    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    2.8660    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.0000    0.2500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.8660   -1.2500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.7321   -0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  3  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  4  5  1  0  0  0  0
  5  6  1  0  0  0  0
  6  7  1  0  0  0  0
  8  9  1  0  0  0  0
  5  9  1  0  0  0  0
M  END
> <EXTREG>
95
$$$$
```

Example Output (default):

```
CCN(CC)CCC#N 000095
```

Example Output (option set to EXTREG):

```
CCN(CC)CCC#N 95
```

`-IFMT [SMI|TDT]` `---`	SMI2MOL and SMI2RD See Section 4.2 DayCart for equivalent option

Description: Controls whether the input file is in SMILES or TDT format. Default is SMI.

Example Input (TDT):

```
$SMI<Clc1ccc(NC(=O)CBr)cc1>
2D<-2.0193,-0.7955,-1.1583,-0.2975,-1.1583,0.4375,-0.5213,0.8055,0.1156,
0.4375,0.7496,0.8035,1.3904,0.4334,2.0313,0.8034,1.3904,-0.3065,0.7495,
-0.6764,0.1156,-0.2975,-0.5213,-0.6655,,,,,,,,,,,,>
$NAM<481>
CODEX<AAFG>
ASSAY1<0.988>
ASSAY2<0.993>
|
```

Example Output:

```
481
  smi2mol

 12 12  0  0  0  0  0  0  0  0999 V2000
   -2.0193   -0.7955    0.0000 Cl  0  0  0  0  0  0  0  0  0  0  0  0
   -1.1583   -0.2975    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.1583    0.4375    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5213    0.8055    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.1156    0.4375    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7496    0.8035    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    1.3904    0.4334    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.0313    0.8034    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    1.3904   -0.3065    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7495   -0.6764    0.0000 Br  0  0  0  0  0  0  0  0  0  0  0  0
    0.1156   -0.2975    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5213   -0.6655    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  3  2  1  0  0  0  0
 12  2  2  0  0  0  0
  3  4  2  0  0  0  0
  4  5  1  0  0  0  0
  6  5  1  0  0  0  0
 11  5  2  0  0  0  0
  6  7  1  0  0  0  0
  8  7  2  0  0  0  0
  9  7  1  0  0  0  0
 10  9  1  0  0  0  0
 12 11  1  0  0  0  0
M  END
> <CODEX>
AAFG
> <ASSAY1>
0.988
> <ASSAY2>
0.993
$$$$
```

-IMPLICIT_CHIRALITY [TRUE|FALSE] CONV_IMPLICIT_CHIRALITY [TRUE|FALSE]	MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS AND RD2SMIRKS SDF/RDF ---> ISM/SMA/SMRK/TDT/TDTSMA/TDTSMRK

Description: Alters the way in which chirality is determined in order to detect implicit chiral centers. This is useful for some natural products. The default is FALSE. Setting -IMPLICIT_CHIRALITY to TRUE allows both ends of chiral bonds to be used in the determination of chiral centers when generating isomeric SMILES, SMARTS or SMIRKS. For a bond A-hash-B, the interpretation is that B is below A from the perspective of A and A is above B from the perspective of B.

Example Input:

```
implicit_chirality
 smi2mol

  8  7  0  0  0  0  0  0  0  0  1
  138.0000  216.0000    0.0000 N   0  0  0  0  0
  159.6506  228.5000    0.0000 C   0  0  0  0  0
  183.7988  222.0295    0.0000 C   0  0  0  0  0
  205.4494  234.5295    0.0000 C   0  0  0  0  0
  153.1802  252.6481    0.0000 Br  0  0  0  0  0
  159.6506  203.5000    0.0000 F   0  0  0  0  0
  205.4494  209.5295    0.0000 Br  0  0  0  0  0
  177.3283  197.8814    0.0000 S   0  0  0  0  0
  2  1  1  0  0  0
  3  2  1  1  0  0
  5  2  1  0  0  0
  6  2  1  0  0  0
  3  4  1  1  0  0
  3  7  1  6  0  0
  3  8  1  6  0  0
M  END
$$$$
```

Example Output (default):

```
C[C@](S)(Br)C(N)(F)Br implicit_chirality
```

Example Output (option set to TRUE):

```
C[C@](S)(Br)[C@@](N)(F)Br implicit_chirality
```

`-M__ISO_ARE_DEFECTS [TRUE|FALSE]`
`CONV_ISO_ARE_DEFECTS`
`[TRUE|FALSE]`

MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS AND RD2SMIRKS
SDF/RDF ---> ISM/SMA/SMRK/TDT/TDTSMA/TDTSMRK

Description: Indicates whether the values in the M ISO line of the property block are mass defects or actual masses for the isotopes listed. Default is FALSE. When -M__ISO_ARE_DEFECTS is set as TRUE, values in the line are treated as defects when generating isomeric SMILES, SMARTS, or SMIRKS

Example Input:

```
M_ISO_is_actual_mass
   smi2mol          2D

  7  6  0  0  0  0  0  0  0  0999 V2000
```

```
    1.8300    1.6000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.1100    2.3200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.1300    2.0600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5900    2.7800    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.5700    2.5100    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -1.8300    1.5300    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3200    3.7600    0.0000 I  -2  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  4  5  2  0  0  0  0
  4  7  1  0  0  0  0
  5  6  1  0  0  0  0
M  ISO  1   7 125
M  END
$$$$
M_ISO_is_mass_default
   smi2mol          2D

  7  6  0  0  0  0  0  0  0  0999 V2000
    1.8300    1.6000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.1100    2.3200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.1300    2.0600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5900    2.7800    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.5700    2.5100    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -1.8300    1.5300    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3200    3.7600    0.0000 I  -2  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  4  5  2  0  0  0  0
  4  7  1  0  0  0  0
  5  6  1  0  0  0  0
M  ISO  1   7  -2
M  END
$$$$
```

Example Output (default):

```
CCC/C(=N\N)/[125I] M_ISO_is_actual_mass
CCC/C(=N\N)/[-2I] M_ISO_is_mass_default
```

Example Output (option set to TRUE):

```
CCC/C(=N\N)/[252I] M_ISO_is_actual_mass
CCC/C(=N\N)/[125I] M_ISO_is_mass_default
```

`-NAME_DATATAG [DATATAG_NAME]`
`CONV_NAMETAG [DATATAG_NAME]`

SMI2MOL and SMI2RD TDT ---> SDF/RDF

Description: Designates the data tag to be used as the unique ID. For SMI input, the space-delimited ID after the SMILES is used. For TDT input, the LINE1 value, if available, is used. Otherwise the default tag is $NAM. In the absence of LINE1, the ID is place on the first line of the header block of the connection for SDfile output. For molecules converted to RDfile output, the ID is save to the

$MEREG field and written to the first line of the header block unless there is a LINE1 datatype. For reactions, the ID is placed in the $REREG field. In addition, multi-branched TDTs are split using the default or designated data tag for both SDfile and RDfile output. Note: One may need to place the tag name in quotes on the command line and use '\' before a '$' if it is an identifier in the TDT.

Example Input:

```
$SMI<CC(N)C(=O)O>
2D<-0.375,3.940,-0.375,2.930,-1.255,2.420,0.505,2.410,1.005,
1.530,1.385,2.910>
$NAM<name>
EXTREG<extreg>
|
```

Example Output (default):

```
name
  smi2mol

  6  5  0  0  0  0  0  0  0  0999 V2000
    -0.375     3.940     0.000 C   0  0  0  0  0  0  0  0  0  0  0  0
    -0.375     2.930     0.000 C   0  4  0  0  0  3  0  0  0  0  0  0
    -1.255     2.420     0.000 N   0  0  0  0  0  1  0  0  0  0  0  0
     0.505     2.410     0.000 C   0  4  0  0  0  3  0  0  0  0  0  0
     1.005     1.530     0.000 O   0  4  0  0  0  1  0  0  0  0  0  0
     1.385     2.910     0.000 O   0  4  0  0  0  1  0  0  0  0  0  0
  1  2  1  0  0  0  0
  3  2  1  0  0  0  0
  2  4  1  0  0  0  0
  5  4  2  0  0  0  0
  6  4  1  0  0  0  0
M  END
> <EXTREG>
extreg
$$$$
```

Example Output (option set to EXTREG):

```
extreg
  smi2mol

  6  5  0  0  0  0  0  0  0  0999 V2000
    -0.375     3.940     0.000 C   0  0  0  0  0  0  0  0  0  0  0  0
    -0.375     2.930     0.000 C   0  4  0  0  0  3  0  0  0  0  0  0
    -1.255     2.420     0.000 N   0  0  0  0  0  1  0  0  0  0  0  0
     0.505     2.410     0.000 C   0  4  0  0  0  3  0  0  0  0  0  0
     1.005     1.530     0.000 O   0  4  0  0  0  1  0  0  0  0  0  0
     1.385     2.910     0.000 O   0  4  0  0  0  1  0  0  0  0  0  0
  1  2  1  0  0  0  0
  3  2  1  0  0  0  0
  2  4  1  0  0  0  0
  5  4  2  0  0  0  0
  6  4  1  0  0  0  0
M  END
$$$$
```

`-OFMT [SMI|ISM|SMA|SMRK|TDT|SQLLDR]`	MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS and RD2SMIRKS
`-OUTPUT_FORMAT [SMI|ISM|SMA|SMRK|TDT|SQLLDR]` `---`	MOL2SMI, RD2SMI, SD2SMARTS, RD2SMARTS and RD2SMIRKS See Section 4.2 DayCart for equivalent option

Description: Controls the output format. The default is SMI (mol2smi and rd2smi) or SMA (sd2smarts and rd2smarts). For the TDT and SQLLDR formats, information on the first line of each input header block (molecules or queries only) and any non-standard atom labels in the input file are stored as LINE1 and as an atom-tuple in the ASYM datatype, respectively. The original atom is designated by '*' in the SMILES. For TDT output, a special $SMIG datatype is written containing data about the conversion program name and version.

Example Input:

```
1
  smi2mol

  7  6  0  0  0  0  0  0  0  0999 V2000
    1.2150    3.2300    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.4850    2.5100    0.0000 C   0  0  1  0  0  0  0  0  0  0  0  0
    0.2250    1.5300    0.0000 Br  0  0  0  0  0  0  0  0  0  0  0  0
   -0.4950    2.7700    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2150    2.0500    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2150    3.4900    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    1.4650    2.2500    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
  2  1  1  1  0  0  0
  3  2  1  0  0  0  0
  2  4  1  0  0  0  0
  5  4  2  0  0  0  0
  6  4  1  0  0  0  0
  2  7  1  6  0  0  0
M  END
>  <ASSAY>
77.6
$$$$
```

Example Output (default):

```
NC(Br)C(=O)O 1
```

Example Output (option set to SQLLDR --- .str file):

```
1;
1;
NC(Br)C(=O)O;
1.215,3.230,0.485,2.510,0.225,1.530,-0.495,2.770,-1.215,2.050,-1.215,3.490;
;
N[C@@H](Br)C(=O)O;
0;
1,1,1,1,1,1,,,1,;
,,,,,,,,,;
,,,,,,,,,;
-1,,,,,,,2,;
1.215,3.230,0.485,2.510,0.225,1.530,-0.495,2.770,-1.215,2.050,-1.215,
3.490,,,,,1.465,2.250,,;
```

Example Output (option set to SQLLDR --- .dat file):

```
;
ASSAY;
~~77.6~~
```

-PREFIX [PREFIX_NAME] CONV_PREFIX [PREFIX_NAME]	RD2SMI, RD2SMARTS and RD2SMIRKS RDF ---> TDT/TDTSMA/TDTSMRK

Description: Parses the designated prefix from data field identifiers. The default is to use the full $DTYPE name. Specifying a prefix removes the defined string from the output $DTYPE name and stores the prefix in the TDT/SQLLDR output files. For molecules, any string can be used. For reactions, the string must cover from the beginning of the name to the first '('. For example RXN:VARIATION can be stripped from the following $DTYPE names: RXN:VARIATION(1) and RXN:VARIATION(2). Also note that in this case, the reaction is split on RXN:VARIATION number with the ID identified by the $RIREG value followed by '_' (underscore) and the RXN:VARIATION number.

Example Input:

```
RFMT $RIREG 1 $REREG
$RXN

  SMI2RD REACTION

  1  1
$MOL

    smi2rd          2D

  4  3  0  0  0  0  0  0  0  0999 V2000
    0.2700    1.3400    0.0000 Br  0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.3600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.7200   -0.3600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.4600   -1.3400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  2  0  0  0  0
M  END
$MOL

    smi2rd          2D

  4  3  0  0  0  0  0  0  0  0999 V2000
    4.8400    1.3400    0.0000 I   0  0  0  0  0  0  0  0  0  0  0  0
    4.5700    0.3600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.2900   -0.3600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.0300   -1.3400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  2  0  0  0  0
M  END
$DTYPE RXN:VARIATION(1):DOC
```

```
$DATUM 75-3121
$DTYPE RXN:VARIATION(2):DOC
$DATUM 75-6562
```

Example Output (default):

```
$SMI<"BrCC=C>>ICC=C">
2D<0.27,1.34,0.00,0.36,0.72,-0.36,0.46,-1.34,4.84,1.34,4.57,0.36,
5.29,-0.36,5.03,-1.34>
$NAM<1>
ISM<"BrCC=C>>ICC=C">
CFLAG<0>
VIS<1,1,1,1,,,,,1,1,1,1,,,,>
VAL<,,,,,,,,,,,,,,,,>
RXNR<,,,,,,,,,,,,,,,,>
RXNC<,,,,,,,,,,,,,,,,>
RXNM<,,,,,,,,,,,,,,,,>
RXNE<,,,,,,,,,,,,,,,,>
RXNI<,,,,,,,,,,,,,,,,>
ASYM<,,,,,,,,,,,,,,,,>
BST<,,,,,,,,,,,,,,,>
2DI<0.27,1.34,0.00,0.36,0.72,-0.36,0.46,-1.34,,,,,,,,,,4.84,1.34,
4.57,0.36,5.29,-0.36,5.03,-1.34,,,,,,,,,>
RIREG<1>
DATA<RXN:VARIATION(1):DOC;"75-3121">
DATA<RXN:VARIATION(2):DOC;"75-6562">
|
```

Example Output (option set to RXN:VARIATION):

```
$SMI<"BrCC=C>>ICC=C">
2D<0.27,1.34,0.00,0.36,0.72,-0.36,0.46,-1.34,4.84,1.34,4.57,0.36,
5.29,-0.36,5.03,-1.34>
$NAM<1>
ISM<"BrCC=C>>ICC=C">
CFLAG<0>
VIS<1,1,1,1,,,,,1,1,1,1,,,,>
VAL<,,,,,,,,,,,,,,,,>
RXNR<,,,,,,,,,,,,,,,,>
RXNC<,,,,,,,,,,,,,,,,>
RXNM<,,,,,,,,,,,,,,,,>
RXNE<,,,,,,,,,,,,,,,,>
RXNI<,,,,,,,,,,,,,,,,>
ASYM<,,,,,,,,,,,,,,,,>
BST<,,,,,,,,,,,,,,,>
2DI<0.27,1.34,0.00,0.36,0.72,-0.36,0.46,-1.34,,,,,,,,,,4.84,1.34,
4.57,0.36,5.29,-0.36,5.03,-1.34,,,,,,,,,>
RIREG<1>
PREFIX<"RXN:VARIATION">
DATA<DOC;"75-3121">
|
$SMI<"BrCC=C>>ICC=C">
2D<0.27,1.34,0.00,0.36,0.72,-0.36,0.46,-1.34,4.84,1.34,4.57,0.36,
5.29,-0.36,5.03,-1.34>
$NAM<1>
ISM<"BrCC=C>>ICC=C">
CFLAG<0>
VIS<1,1,1,1,,,,,1,1,1,1,,,,>
VAL<,,,,,,,,,,,,,,,,>
RXNR<,,,,,,,,,,,,,,,,>
```

```
RXNC<,,,,,,,,,,,,,,,,>
RXNM<,,,,,,,,,,,,,,,,>
RXNE<,,,,,,,,,,,,,,,,>
RXNI<,,,,,,,,,,,,,,,,>
ASYM<,,,,,,,,,,,,,,,,>
BST<,,,,,,,,,,,,,,,>
2DI<0.27,1.34,0.00,0.36,0.72,-0.36,0.46,-1.34,,,,,,,,,,4.84,1.34,
4.57,0.36,5.29,-0.36,5.03,-1.34,,,,,,,,,>
RIREG<1>
PREFIX<"RXN:VARIATION">
DATA<DOC;"75-6562">
|
```

`-PTABLE [PTABLE_NAME]`
`---`

MOL2SMI and RD2SMI
See Section 4.2 DayCart for equivalent option

Description: Provides location of user-defined periodic table. Setting this option with a name of a user-defined P-table causes the atoms in the default table to be overwritten by the atoms present in the user P-table. An example table is located in $DY_ROOT/data. Uncommenting and editing a specific line of this file changes the valence and charge used for this atom.

Example Input:

```
charged_carbon
  smi2mol

  5  5  0  0  0  0  0  0  0  0999 V2000
    0.5050    4.0800    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5050    4.0800    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.8150    3.1300    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.0050    2.5400    0.0000 C   0  5  0  0  0  0  0  0  0  0  0  0
    0.8250    3.1200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  1  5  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  5  4  1  0  0  0  0
M  END
$$$$
```

P-table Change:

```
From "#6 C 12 3,-1,4,0,3,1" to "6 C 12 4,0"
```

Example Output (default):

```
C1CC[CH-]C1 charged_carbon
```

Example Output (option set to reference user p-table):

```
ERROR: No PTable entry for: C with 2 bonds and charge -1 (get_Hcount)
ERROR: Valence error (dy_ctab2smi)
```

-SMI_COMMENT [TRUE|FALSE] CONV_COMMENT_SMI [TRUE|FALSE]	SMI2MOL and SMI2RD SMI/TDT ---> SDF/RDF

Description: Determines whether the SMILES is placed in the comment line. Default is FALSE. Designating -SMI_COMMENT as TRUE writes the SMILES to the comment line (line 3) of the header block in each connection table. Note: The comment line is limited to 80 characters.

Example Input:

```
CCCC(=NN)Br comment
```

Example Output (default):

```
comment
   smi2mol          2D

  7  6  0  0  0  0  0  0  0  0999 V2000
    1.8300    1.6000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.1100    2.3200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.1300    2.0600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5900    2.7800    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.5700    2.5100    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -1.8300    1.5300    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3200    3.7600    0.0000 Br  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  4  5  2  3  0  0  0
  4  7  1  0  0  0  0
  5  6  1  0  0  0  0
M  END
$$$$
```

Example Output (option set to TRUE):

```
comment
   smi2mol          2D
CCCC(=NN)Br
  7  6  0  0  0  0  0  0  0  0999 V2000
    1.8300    1.6000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.1100    2.3200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.1300    2.0600    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5900    2.7800    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.5700    2.5100    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -1.8300    1.5300    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3200    3.7600    0.0000 Br  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  4  5  2  3  0  0  0
  4  7  1  0  0  0  0
  5  6  1  0  0  0  0
M  END
$$$$
```

-SMI_IS_ISM [TRUE|FALSE] ----	MOL2SMI and RD2SMI CONV_SMI_IS_ISM [TRUE|FALSE]

Description: Replaces SMILES with isomeric SMILES in the output. Some programs such as rubicon require the SMILES datatype carry isomeric information. The default value for -SMI_IS_ISM is FALSE. Setting this option to be TRUE allows isomeric information to be stored in the SMILES data type.

Example Input:

```
stereo
   smi2mol          2D

  7  6  0  0  0  0  0  0  0  0999 V2000
   -0.6300    3.4900    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3600    2.5100    0.0000 C   0  0  2  0  0  0  0  0  0  0  0  0
   -0.1000    1.5300    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.6200    2.7700    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.8800    3.7500    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    1.3400    2.0500    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -1.3400    2.2500    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
  2  1  1  1  0  0  0
  2  3  1  1  0  0  0
  2  4  1  6  0  0  0
  4  5  2  0  0  0  0
  4  6  1  0  0  0  0
  2  7  1  6  0  0  0
M  END
$$$$
```

Example Output (default):

```
$SMI<CC(N)C(=O)O>
2D<-0.63,3.49,-0.36,2.51,-0.10,1.53,0.62,2.77,0.88,3.75,1.34,2.05>
$NAM<stereo>
LINE1<stereo>
ISM<C[C@H](N)C(=O)O>
CFLAG<0>
VIS<1,1,1,1,1,1,,,,1,,,>
VAL<,,,,,,,,,,,,>
ASYM<,,,,,,,,,,,,>
BST<-1,1,2,,,,,,2,,,>
2DI<-0.63,3.49,-0.36,2.51,-0.10,1.53,0.62,2.77,0.88,3.75,1.34,
2.05,,,,,,,-1.34,2.25,,,,,,>
|
```

Example Output (option set to TRUE):

```
$SMI<C[C@H](N)C(=O)O>
2D<-0.63,3.49,-0.36,2.51,-0.10,1.53,0.62,2.77,0.88,3.75,1.34,2.05>
$NAM<stereo>
LINE1<stereo>
ISM<C[C@H](N)C(=O)O>
CFLAG<0>
VIS<1,1,1,1,1,1,,,,1,,,>
```

```
VAL<,,,,,,,,,,,>>
ASYM<,,,,,,,,,,,>
BST<-1,1,2,,,,,,2,,,>
2DI<-0.63,3.49,-0.36,2.51,-0.10,1.53,0.62,2.77,0.88,3.75,1.34,
2.05,,,,,,,-1.34,2.25,,,,,,>
|
```

`-SMI_WITH_TUPLES [TRUE|FALSE]`
`(CONV_SMI_TUPLES [TRUE|FALSE])`

SMI2MOL and
SMI2RD
TDT ---> SDF/RDF

Description: Determines whether output tuple information is associated with SMILES or isomeric SMILES. Default is TRUE so that tuples associated with $SMI (2D or $D3D) are saved in the output file. Setting this option to FALSE outputs the tuple information associated with the ISM (2DI or 3DI).

Example Input:

```
$SMI<CCC=CBr>
2D<2.47,-5.38,1.76,-4.97,1.05,-5.38,0.25,-5.17,-0.46,-5.59;>
$NAM<tuple_with_smiles>
ISM<[2H]/C(=C(/[3H])\CC)/Br>
|
$SMI<CCC=CCl>
$NAM<tuple_with_isomeric_smiles>
ISM<[2H]/C(=C(/[3H])\CC)/Cl>
2DI<-0.16,-4.45,0.25,-5.17,1.05,-5.38,1.26,-6.17,
1.76,-4.96,2.47,-5.37,-0.45,-5.58,,,,,,,,,>
|
```

Example Output (default - generates coords of tuple_with_isomeric_smiles):

```
tuple_with_smiles
   smi2mol          2D

  5  4  0  0  0  0  0  0  0  0999 V2000
      2.47     -5.38      0.00 C   0  0  0  0  0  0  0  0  0  0  0  0
      1.76     -4.97      0.00 C   0  0  0  0  0  0  0  0  0  0  0  0
      1.05     -5.38      0.00 C   0  0  0  0  0  0  0  0  0  0  0  0
      0.25     -5.17      0.00 C   0  0  0  0  0  0  0  0  0  0  0  0
     -0.46     -5.59      0.00 Br  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  2  3  0  0  0
  4  5  1  0  0  0  0
M  END
$$$$
tuple_with_isomeric_smiles
   smi2mol          2D

  5  4  0  0  0  0  0  0  0  0999 V2000
    1.7600    1.5300    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.8800    2.0400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    1.5300    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.8800    2.0400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.7600    1.5300    0.0000 Cl  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
```

```
  2  3  1  0  0  0  0
  3  4  2  3  0  0  0
  4  5  1  0  0  0  0
M  END
$$$$
```

Example Output (option set to FALSE - generates coords for tuple_with_smiles):

```
tuple_with_smiles
   smi2mol          2D

  7  6  0  0  0  0  0  0  0  0999 V2000
   -0.8900    1.5300    0.0000 D   0  0  0  0  0  0  0  0  0  0  0  0
   -0.8800    2.5400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    3.0400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    4.0500    0.0000 T   0  0  0  0  0  0  0  0  0  0  0  0
    0.8800    2.5200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.7600    3.0200    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.7600    3.0500    0.0000 Br  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
  2  7  1  0  0  0  0
  3  4  1  0  0  0  0
  3  5  1  0  0  0  0
  5  6  1  0  0  0  0
M  ISO  2   1   2   4   3
M  END
$$$$
tuple_with_isomeric_smiles
   smi2mol          2D

  7  6  0  0  0  0  0  0  0  0999 V2000
   -0.16     -4.45      0.00   D   0  0  0  0  0  0  0  0  0  0  0  0
    0.25     -5.17      0.00   C   0  0  0  0  0  0  0  0  0  0  0  0
    1.05     -5.38      0.00   C   0  0  0  0  0  0  0  0  0  0  0  0
    1.26     -6.17      0.00   T   0  0  0  0  0  0  0  0  0  0  0  0
    1.76     -4.96      0.00   C   0  0  0  0  0  0  0  0  0  0  0  0
    2.47     -5.37      0.00   C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.45     -5.58      0.00   Cl  0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
  2  7  1  0  0  0  0
  3  4  1  0  0  0  0
  3  5  1  0  0  0  0
  5  6  1  0  0  0  0
M  ISO  2   1   2   4   3
M  END
$$$$
```

`-SPLIT_FIELDS [TRUE|FALSE]` `CONV_SPLIT_FIELDS [TRUE|FALSE]`	MOL2SMI, SMI2MOL and SD2SMARTS SDF ---> TDT/TDTSMA or TDT ---> SDF

Description: Splits data that is spread across multiple lines in an input into separate entries. The default is FALSE so that multiple lines are considered as a single value. Setting SPLIT_FIELDS to TRUE allows each line of a multi-line field to be considered as a separate value with the same data field identifier or data tag.

Example Input:

```
multiple_lines
   smi2mol          2D

  4  3  0  0  0  0  0  0  0  0999 V2000
   -1.3400    1.8000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.3600    1.5300    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.3600    2.2500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.3400    1.9900    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  2  0  0  0  0
M  END
$$$$> <TEST>
3X
NN5
32
$$$$
```

Example Output (default):

```
$SMI<CCC=O>
2D<-1.34,1.80,-0.36,1.53,0.36,2.25,1.34,1.99>
$NAM<CCC=O>
LINE1<>
ISM<CCC=O>
CFLAG<0>
VIS<1,1,1,1,,,,,,>
VAL<,,,,,,,,,>
ASYM<,,,,,,,,,>
BST<,,,,,,,,>
2DI<-1.34,1.80,-0.36,1.53,0.36,2.25,1.34,1.99,,,,,,,,,,,>
TEST<3X;NN5;32;>
|
```

Example Output (option set to TRUE):

```
$SMI<CCC=O>
2D<-1.34,1.80,-0.36,1.53,0.36,2.25,1.34,1.99>
$NAM<CCC=O>
LINE1<>
ISM<CCC=O>
CFLAG<0>
VIS<1,1,1,1,,,,,,>
VAL<,,,,,,,,,>
ASYM<,,,,,,,,,>
BST<,,,,,,,,>
2DI<-1.34,1.80,-0.36,1.53,0.36,2.25,1.34,1.99,,,,,,,,,,,>
TEST<3X>
TEST<NN5>
TEST<32>
|
```

-USE_3D [TRUE|FALSE] CONV_USE3D [TRUE|FALSE]	SMI2MOL and SMI2RD TDT ---> SDF/RDF

Description: Designates whether 3D coordinates are included in the output. Default is FALSE. If -USE_3D is set to TRUE and the input TDT file contains 3D coordinates, then 3D coordinates are included in the output file.

Example Input:

```
$SMI<C=CCl>
$D3D<0.0;3D;2.47,-5.38,0.25,1.76,-4.97,0.33,1.05,-5.38,0.55;n/a>
$NAM<3D>
|
```

Example Output (default):

```
3D
   smi2mol          2D

  3  2  0  0  0  0  0  0  0  0999 V2000
   -0.8800    1.5300    0.0000 Cl  0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    2.0400    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.8800    1.5300    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
M  END
$$$$
```

Example Output (option set to TRUE):

```
3D
   smi2mol          3D

  3  2  0  0  1  0  0  0  0  0999 V2000
      2.47     -5.38      0.25 Cl  0  0  0  0  0  0  0  0  0  0  0  0
      1.76     -4.97      0.33 C   0  0  0  0  0  0  0  0  0  0  0  0
      1.05     -5.38      0.55 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  2  3  2  0  0  0  0
M  END
$$$$
```

Daylight Conversion Package Appendix

Example Files

Example input and output files are shown below:

SMILES:

```
CCCCOCCOCCOC(=O)CCCCCCCCC(=O)O
ClCCN(Cl)(C)(C)C 3334
C=CCOc1ccc(cc1)C(C)Cc2ccc(cc2)OCC=C 56XC
 678C2
```

Isomeric SMILES:

```
Cc1ccc(NC(=O)C(=O)N/N=C/C=C/c2ccccc2)cc1 S46282
C[C@@]12CC[C@H]3[C@@H](CC[C@@H]4C[C@@H](O)CC[C@@]43C)[C@]2(O)CC[C@H]1c5ccc(=O)oc5 7743
CN1Cc2c(ncn2-c3ccc(F)cc3C1=O)C(=O)OCC[18F]
C[Si](N[Si](C)(C)C)(C)C 4413
```

SMARTS:

```
[Cl,Br][C;v4]([C][C]([C])[C][C])[N][C]
[C][c]:1:[c]:[c]:[c]([C]):[c]:[c]1 LST12
[c,n]:[14c]:[!c;!o;H0]:[c;!H0] XCD123
```

SMIRKS:

```
[!#6;!#1:1][C@:5]([C,c:2])([F:3])[Cl:4]>>[!S;!N:1][C@:5]([C,c:2])([F:3])[Cl:4] 19
[C]C1=[C][N][C]=C1[N]>>[C]C1=[C][N][C]=C1[N]
[C]c1[c][n]([H])[c]c1[N]>>[C]c1[c][n]([H])[c]c1[N] transform2
```

molfile:

```
33456
  DCIS

  7  5  0     0  0              1 V2000
    0.4552    0.3828    0.0000 C   0  0  0  0  0  0  0  0  0
    1.1552   -0.0448    0.0000 C   0  0  0  0  0  0  0  0  0
   -0.2759   -0.0103    0.0000 C   0  0  0  0  0  0  0  0  0
    1.1379   -0.8724    0.0000 O   0  0  0  0  0  0  0  0  0
    1.8862    0.3517    0.0000 O   0  0  0  0  0  0  0  0  0
   -0.9759    0.4172    0.0000 C   0  0  0  0  0  0  0  0  0
   -3.3828   -0.2241    0.0000 Na  0  3  0  0  0  0  0  0  0
  1  2  1  0  0  0
  1  3  1  0  0  0
  2  4  1  0  0  0
  2  5  2  0  0  0
  3  6  1  0  0  0
M  CHG  1   7   1
M  END
```

molfile (query):

```
785
  DCIS

 10  9  1  0  0  0  0  0  0  0999 V2000
    3.5875   -6.7958    0.0000 C   0  0  3  0  0  0  0  0  0  0  0  0
    4.3000   -6.3792    0.0000 L   0  0  0  0  0  0  0  0  0  0  0  0
    2.8708   -6.3833    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    2.8708   -7.2042    0.0000 H   0  0  0  0  0  0  0  0  0  0  0  0
    3.9958   -7.5083    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.5811   -8.2215    0.0000 C   0  0  3  0  0  0  0  0  0  0  0  0
    3.9913   -8.9373    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.7561   -8.2188    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.8697   -5.5583    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    4.8083   -8.9333    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  3  1  0  0  0  0
  5  6  1  0  0  0  0
  1  2  1  0  0  0  0
  6  7  1  0  0  0  0
  1  4  1  0  0  0  0
  6  8  1  0  0  0  0
  3  9  1  0  0  0  0
  7 10  1  0  0  0  0
  1  5  1  0  0  0  0
M  ALS   2  4 F F   Cl  Br  I
M  END
```

SDfile:

```
90003-X
  smi2mol

  7  7  0     0  0              1 V2000
    0.0000   -0.6828    0.0000 N   0  3  0  0  0  0  0  0  0
    1.2379    0.0483    0.0000 C   0  0  0  0  0  0  0  0  0
   -1.2379    0.0483    0.0000 C   0  0  0  0  0  0  0  0  0
    0.0000   -2.1103    0.0000 O   0  5  0  0  0  0  0  0  0
    1.2379    1.4724    0.0000 C   0  0  0  0  0  0  0  0  0
   -1.2379    1.4724    0.0000 C   0  0  0  0  0  0  0  0  0
    0.0000    2.2034    0.0000 N   0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0
  1  3  2  0  0  0
  1  4  1  0  0  0
  2  5  2  0  0  0
  3  6  1  0  0  0
  5  7  1  0  0  0
  6  7  2  0  0  0
M  END
> <MOLNAME>
90003-X
> <MOL.WEIGHT>
96.088
$$$$
```

RGfile:

```
$MDL  REV  1 0107081157
$MOL
$HDR
1R-23
  -ISIS-  01070811572D
```

```
$END HDR
$CTAB
  7  7  0  0  0  0  0  0  0  0999 V2000
    4.5000    0.0208    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    4.5000   -0.8042    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.2120   -1.2125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.9240   -0.8042    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.9240    0.0208    0.0000 C   0  0  3  0  0  0  0  0  0  0  0  0
    5.2120    0.4375    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    6.6397    0.4312    0.0000 R#  0  0  0  0  0  0  0  0  0  0  0  0
  5  7  1  0  0  0  0
  1  2  1  0  0  0  0
  1  6  1  0  0  0  0
  2  3  1  0  0  0  0
  3  4  1  0  0  0  0
  4  5  1  0  0  0  0
  5  6  1  0  0  0  0
M  RGP  1   7   1
M  LOG  1   1   0   0   >0
M  END
$END CTAB
$RGP
   1
TAB
  3  2  0  0  0  0  0  0  0  0999 V2000
   -0.5544   -4.0375    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.1600   -3.6250    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.5439   -4.8542    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
  1  3  1  0  0  0  0
M  APO  1   3   1
M  END
$END CTAB
$CTAB
  3  2  0  0  0  0  0  0  0  0999 V2000
    1.4873   -4.0500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.2017   -3.6375    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    1.4873   -4.8750    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  2  0  0  0  0
  1  3  1  0  0  0  0
M  APO  1   3   1
M  END
$END CTAB
$CTAB
  2  1  0  0  0  0  0  0  0  0999 V2000
    3.5542   -5.0666    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    4.2686   -4.6541    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0  0
M  APO  1   1   1
M  END
$END CTAB
$END RGP
$END MOL
```

RDfile (molecule):

```
$RDFILE 1
$DATM    12/05/05 22:51
$MFMT $MIREG 1
```

```
  smi2rd

  7  7  0     0  0            999 V2000
    0.4483    0.6655    0.0000 C   0  0  0  0  0  0  0  0  0
   -0.8552    1.4172    0.0000 C   0  0  0  0  0  0  0  0  0
    0.4483   -0.8414    0.0000 C   0  0  0  0  0  0  0  0  0
    1.7483    1.4172    0.0000 Se  0  0  0  0  0  0  0  0  0
   -2.1586    0.6655    0.0000 C   0  0  0  0  0  0  0  0  0
   -0.8552   -1.5931    0.0000 C   0  0  0  0  0  0  0  0  0
   -2.1586   -0.8414    0.0000 C   0  0  0  0  0  0  0  0  0
  1  2  1  0  0  0
  1  3  2  0  0  0
  1  4  1  0  0  0
  2  5  2  0  0  0
  3  6  1  0  0  0
  5  7  1  0  0  0
  6  7  2  0  0  0
M  END
$DTYPE MOL:NAME
$DATUM SECC4
$DTYPE MOL:EXTREG
$DATUM 112
$DTYPE MOL:MOLECULAR.WEIGHT
$DATUM 157.073
```

RDfile (reaction):

```
$RDFILE 1
$DATM    12/06/05 21:51
$RFMT $RIREG 1
$RXN

  SMI2RD REACTION

  1  2
$MOL

  smi2rd

  6  6  0  0  0  0              1 V2000
    0.0000    1.5422    0.0000 C   0  0  0  0  0  0  0  0  0  1  0  0
   -1.3335    0.7681    0.0000 C   0  0  0  0  0  0  0  0  0  2  0  0
    1.3335    0.7681    0.0000 C   0  0  0  0  0  0  0  0  0  3  0  0
   -1.3335   -0.7681    0.0000 C   0  0  0  0  0  0  0  0  0  4  0  0
    1.3335   -0.7681    0.0000 C   0  0  0  0  0  0  0  0  0  5  0  0
    0.0000   -1.5422    0.0000 C   0  0  0  0  0  0  0  0  0  6  0  0
  1  2  1  0  0  0  0
  1  3  1  0  0  0  0
  2  4  1  0  0  0  0
  3  5  1  0  0  0  0
  4  6  1  0  0  0  0
  5  6  1  0  0  0  0
M  END
$MOL

  smi2rd

  7  7  0  0  0  0              1 V2000
   10.0047    0.6541    0.0000 C   0  0  3  0  0  0  0  0  0  1  0  0
   -1.3315   -0.1191    0.0000 C   0  0  0  0  0  0  0  0  0  2  0  0
   11.3315   -0.1191    0.0000 C   0  0  0  0  0  0  0  0  0  3  0  0
```

```
   10.0070    2.1935    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -1.3315   -1.6585    0.0000 C   0  0  0  0  0  0  0  0  0  4  0  0
   11.3315   -1.6585    0.0000 C   0  0  0  0  0  0  0  0  0  5  0  0
   10.0047   -2.4341    0.0000 C   0  0  0  0  0  0  0  0  0  6  0  0
  1  2  1  0  0  0  0
  1  3  1  0  0  0  0
  1  4  1  0  0  0  0
  2  5  1  0  0  0  0
  3  6  1  0  0  0  0
  5  7  1  0  0  0  0
  6  7  1  0  0  0  0
M  END
$MOL

  smi2rd

  7  7  0  0  0  0              1 V2000
    0.0000    0.6445    0.0000 C   0  0  0  0  0  0  0  0  0  1  0  0
    1.3312   -0.1148    0.0000 C   0  0  0  0  0  0  0  0  0  2  0  0
   -1.3312   -0.1148    0.0000 C   0  0  0  0  0  0  0  0  0  3  0  0
    0.0000    2.2006    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    1.3312   -1.6569    0.0000 C   0  0  0  0  0  0  0  0  0  4  0  0
   -1.3312   -1.6569    0.0000 C   0  0  0  0  0  0  0  0  0  5  0  0
    0.0000   -2.4420    0.0000 C   0  0  0  0  0  0  0  0  0  6  0  0
  1  2  1  0  0  0  0
  1  3  1  0  0  0  0
  1  4  2  0  0  0  0
  2  5  1  0  0  0  0
  3  6  1  0  0  0  0
  5  7  1  0  0  0  0
  6  7  1  0  0  0  0
M  END
$DTYPE RXN:VARIATION(1):NAME
$DATUM DD45
$DTYPE RXN:VARIATION(1):CONC
$DATUM (0.25 mmol)
$DTYPE RXN:VARIATION(1):CATALYST(1):REGNO
$DATUM $MFMT $MIREG 432

  smi2rd

  3  2  0  0  0  0              1 V2000
    1.1893   -0.1551    0.0000 C   0  0  0  0  0  0
   -0.1395   -0.9334    0.0000 C   0  0  0  0  0  0
    1.1893    1.3753    0.0000 O   0  0  0  0  0  0
  1  2  1  0  0  0
  1  3  2  0  0  0
M  END
```

TDT (structure branched on $NAM):

```
$SMI<Ic1cccc(C=O)c1>
2D<-1.26,5.02,-0.75,4.14,-1.27,3.29,-0.76,2.41,0.25,2.41,0.76,3.27,1.77,3.27,
2.28,2.40,0.25,4.14,,,,,,,,,>
$NAM<N1a>
CAT<Supplier1;500026271;Fine;62.10/G USD>
$NAM<N1b>
CAT<Supplier2;800045;Fine;65.55/G USD per 5 G>
|
```

TDT (non-structure):

```
$NAM<BB2948>
CFLAG<0>
VIS<>
BST<>
EXTREG<XR32D>
RN<CR01050>
COMMENT<Structure unknown>
|
```

SQLLDR .dat:

```
XX12;
CODE;
36A
XX12;
Assay1;
45.23
```

SQLLDR .str:

```
general_line1;
general_line1;
NC(Br)C(=O)O;
1.215,3.230,0.485,2.510,0.225,1.530,-0.495,2.770,-1.215,2.050,-1.215,3.490;
;
N[C@@H](Br)C(=O)O;
0;
1,1,1,1,1,1,,,1,;
,,,,,,,,;
,,,,,,,,;
-1,,,,,,,2,;
1.215,3.230,0.485,2.510,0.225,1.530,-0.495,2.770,-1.215,2.050,-1.215,3.490
,,,,,1.465,2.250,,;
```