Symyx

CTfile Formats

June 2010

For the latest version:

This documentation is available at www.symyx.com. Log in and navigate to the download site for this product.

June 2010

Contents

About Symyx CTfile formats

Connection Table [CTAB]

Molfile

RGfiles (Rgroup file)

Rxnfile

V2000 Connection Table [CTAB]

V2000 Molfile

V2000 Rxnfile

V2000 RGfiles (Rgroup file)

SDfiles (multiple structures and optional data)

RDfiles (multiple reactions and optional data)

Appendix A: Stereo Notes

Index

About Symyx CTfile formats

Overview

Symyx applications support various formats of chemical table files formats (CTfile formats) for representing and communicating chemical information. To find out which file formats a specific Symyx product supports, see the documentation for that product.

Note: Symyx products were formerly MDL products.

The intended audience for this document is any software programmer who is coding an application that parses files written in one or more of the CTfile formats.

Change Log

The following are the changes in this document:

Change	Page(s)
June 2010	
V3000 molfile description updated for template-based expansion supporting Biologics and other large structures	See "Template block" on page 30.
V3000 is now preferred over V2000, so the chapters have been reordered accordingly.	
Removed many references to products that are no longer supported. (Some references remain to support legacy.)	
The contents of the Atom Limits Enhancements chapter has been moved inside of the V2000 Molfile chapter	
Removed the detailed XDfile chapter and provided cross-reference to where to get the same information.	See XDfiles in "CTfiles - Summary of Each Format" on page 7
November 2007	
Product and company rebranding changes (Symyx)	
June 2005	
Fixed example in 3D feature type -5 in "The Connection Table [CTAB] (V2000)"	See "The Properties Block for 3D Features [3D]" on page 59.
Added "Automatic V3000 Output" section.	See "Automatic V3000 output" on page 7.
Product and company rebranding changes (Elsevier MDL)	
October 2003	
Added new chapter on XDfile format	
Added XDfile in this Introduction	
August 2002	
Deleted chapter on moving CTfiles on/off the Clipboard	
Minor corrections	
Added new information on enhanced stereochemistry features	

Added new chapter on Extended Reaction File format	See "Rxnfile" on page 41.
May, 2001	
Added section describing advantages of V3000 file format	See "Molfile in V3000 Format - Advantages Over V2000" on page 6..
Added section on V3000 Collection Block	
Minor corrections	
December, 1999	
Updated entries in "Atom List"	
December, 1998	
Updated "Example of an SDfile"	
August, 1998	
Added STBOX field	
June, 1997	
Added Atom Attachment Order	
Added new ATTCHORD field	
October, 1996	
Minor corrections	
Enhanced description of connection table properties block	
Added Sgroup bracket style	

Molfile in V3000 Format - Advantages Over V2000

The V3000 format is intended to be the primary means for communication of future enhancements to Symyx chemical representation features. The preferred molfile format (V3000) offers advantages over the legacy V2000 format:

- Provides better support for new chemical properties or objects, and supports enhanced stereochemistry
- Removes fixed field widths to support large structures. (The fixed limits and distributed property information in the V2000 format make V2000 less than ideal for enhancing chemical representation.)
- Supports the use of templates in a template block, which is useful for representing large structures, such as biological molecules. See "Template block" on page 30. Consolidates property information for chemical objects.
- Uses free format and tagging of information for easier parsing
- Provides better backward compatibility through BEGIN/END blocks

Automatic V3000 output

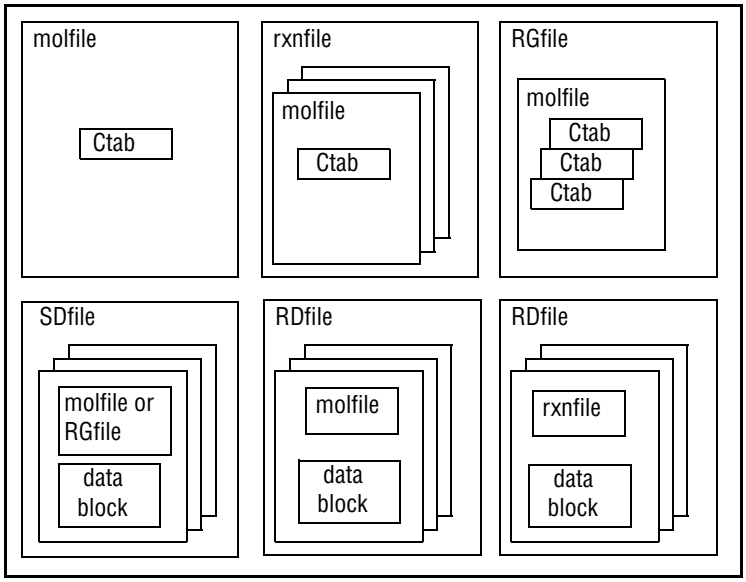
Current Symyx products support reading and writing of both V2000 and V3000 formats. These products continue to default to writing V2000 molfiles to maximize interoperability with third party applications. Future product versions might default to output of the preferred V3000 format.

Because of the limitations imposed by the V2000 format, there are situations when the V3000 format must be used:

- Structure highlighting - The V3000 format is required for molecule or collection highlighting. For information about structure highlighting, see the "Collection block" on page 22.
- Enhanced stereochemistry features - The V3000 format is required when using the enhanced stereochemical representations. See the "Collection block" on page 22. For a complete discussion of Symyx enhanced chemical representation, see *Symyx Chemical Representation*.
- Long fields - If any of the fields with fixed widths for any connection table properties are exceeded, the V3000 format is used. For example, if the number of atoms (or bonds) exceeds 999. This is because the number of atoms (or bonds) on the V2000 counts line cannot exceed 3 columns (see "The Counts Line" on page 45). For more details about the fixed field widths in the V2000 format, see "V2000 Connection Table [CTAB]" on page 43.
- Template block - useful for representing large structures, such as biological molecules. See "Template block" on page 30.
- New properties supported only in V3000 format - such as atom CLASS and SEQID. See "Meaning of values in the atom block" on page 14

CTfiles - Summary of Each Format

The following diagram illustrates the relationship between the various file formats. See "Description of the formats" on page 8.



Description of the formats

molfiles Molecule files: Each molfile describes a single molecular structure which can contain disjoint fragments. See "Molfile" on page 33.

RGfiles Rgroup files: An RGfile describes a single molecular query with Rgroups. Each RGfile is a combination of Ctabs defining the root molecule and each member of each Rgroup in the query. See "RGfiles (Rgroup file)" on page 37.

rxnfiles Reaction files: Each rxnfile contains the structural information for the reactants and products of a single reaction. See "Rxnfile" on page 41.

SDfiles Structure-data files: An SDfile contains structures and data for any number of molecules. Together with RDfiles, SDfiles are the primary format for large-scale data transfer between Symyx databases. See "SDfiles (multiple structures and optional data)" on page 77.

RDfiles Reaction-data files: Similar to SDfiles in concept, the RDfile is a more general format that can include reactions as well as molecules, together with their associated data. See "RDfiles (multiple reactions and optional data)" on page 81.

XDfiles XML-data files: XML-based data format for transferring recordsets of structure or reaction information with associated data. An XDfile can contain structures or reactions that use any of the CTfile formats, Chime strings, or SMILES strings.

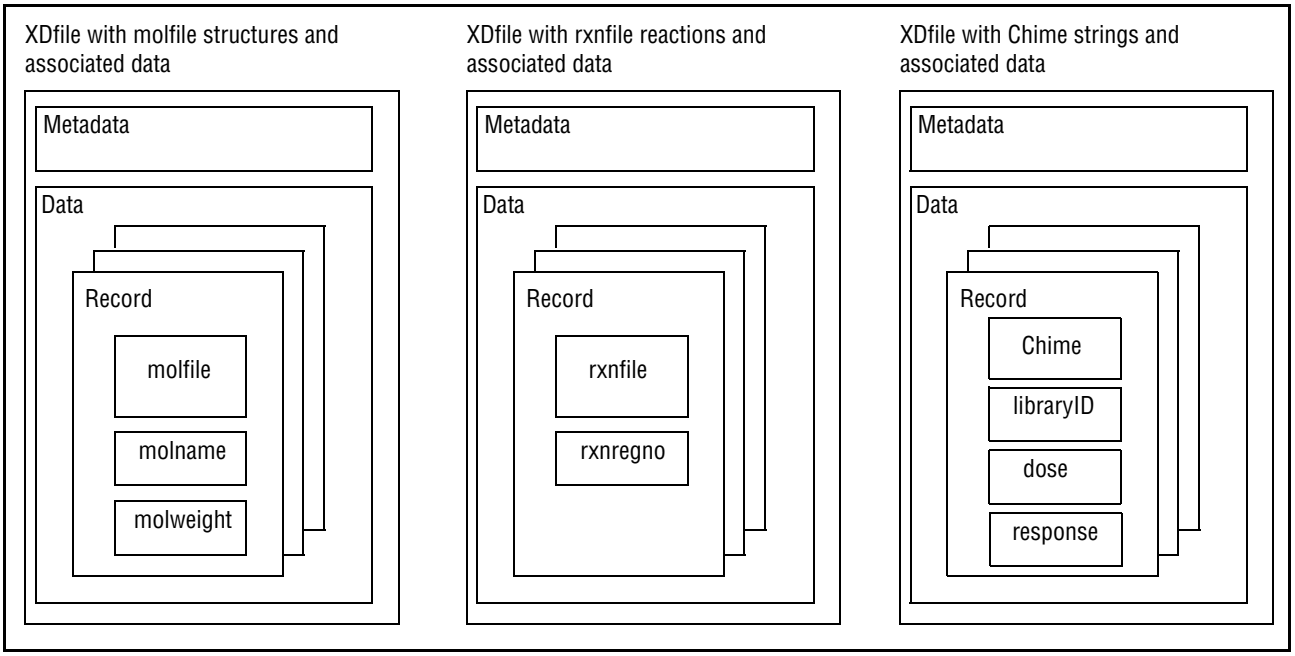
Chime is a compressed, encrypted format used to render structures and reactions on a Web page.

SMILES (Simplified Molecule Input Line Entry System) is a line notation format that uses character strings and SMILES syntax to represent a structure.

For detailed documentation of this file format, see the *XML Reference* of Symyx Isentris Developer Documentation.

XDfile types

The following diagram illustrates three examples of XDfiles:



What various CTfile types support

Some of the structural and query properties described in this document are generic in their applicability, while others are peculiar to certain CTfile types. The applicability of each property is identified in subsequent chapters by the bracketed terms shown in the following table.

Figure 1 Properties applicable to various CTfile types

Property	molfile	RGfile	SDfile	rxnfile	RDfile	XDfile
[Generic]	+	+	+	+	+	+ (mol/rxn)
[Sgroup]	+	+	+	[+]	+	+ (mol/[rxn])
[Rgroup]	+	+	+			+ (mol)
[3D]	+	+	+			+ (mol)
[Reaction]				+	+	+ (rxn)
[Query]	+	+	+	+	+	+ (mol/rxn)

Note: The XDfile inherits the functionality of the format of the embedded structure or reaction. In addition to the molfile and rxnfile formats, the XDfile supports Chime and SMILES strings.

Conventions

The format conventions used in this document are as follows:

UPPERCASE	Literal text, to be entered as shown. Only the position of "M V30 " is significant. White space can be added anywhere else to improve readability. Both lower- and uppercase characters, or any combination of them, are acceptable for literals. They are shown here in uppercase for readability.
lowercase	A token, which is defined elsewhere.
[ ]	An optional item. Do not include the brackets.
[ ]*	An optional item, where there can be zero, one, two, or more of the item.
|	Separates two or more options, only one of which is valid.
/	Separates two or more items. Either or both can appear in any order.
{}	Braces are used for grouping. They indicate indefinite or definite repeat.

Connection Table [CTAB]

Overview

A connection table (Ctab) contains information describing the structural relationships and properties of a collection of atoms. The atoms can be wholly or partially connected by bonds. (An atom can also be an unconnected fragment.) Such collections might, for example, describe molecules, molecular fragments, substructures, substituent groups, polymers, alloys, formulations, mixtures, and unconnected atoms.

The connection table is fundamental to all Symyx CTfile formats. The Ctab is included in a molfile, and multiple Ctabs can be included in an rxnfile, RGfile, RDfile, or SDfile.

This chapter provides an overview of the connection table (CTAB) for the preferred V3000 format. For information on the legacy format, see "V2000 Connection Table [CTAB]" on page 43.

Ctab block format

The format for a Ctab block is:

Counts line:	Specifies the number of atoms, bonds, Sgroups, 3D constituents, as well as the chiral flag setting, and the regno. For details, see "Counts Line" on page 13.
Atom block:	Specifies the atomic symbol and any mass difference, charge, stereochemistry, and associated hydrogens for each atom. For details, see "Atom Block" on page 13.
Bond block:	Specifies the two atoms connected by the bond, the bond type, and any bond stereochemistry and topology (chain or ring properties) for each bond. For details, see "Bond block" on page 16.
Link atom line:	Supports the representation of certain kinds of Sgroups with repeating units, such as link nodes. For details about Sgroups, see *Symyx Chemical Representation*. See "Link atom line" on page 17.
Sgroup block:	"Sgroup block" on page 18. For details about Sgroups, see *Symyx Chemical Representation*.
Collection block:	A collection block specifies all collection information for objects in the current connection table context. See "Collection block" on page 22.
3D block:	Contains values for x, y, and z dimensions, and supports three-dimensional concepts, such as plane and exclusion sphere. See "3D block" on page 25.
Rgroup block:	Supports Markush representation. See "Rgroup logic lines" on page 29. For details about Markush structures (Rgroups), see *Symyx Chemical Representation*.
Template block	Supports the representation of large molecules, such as biologics. See "Template block" on page 30.

General Syntax of Entries

The general syntax of an entry is:

```
M  V30 key posval posval ... [keyword=value] [keyword=value] ...
```

or

```
M  V30 BEGIN key [blockname]
M  V30 posval posval ... keyword=value keyword=value ...
...
M  V30 END key
```

Each line must begin with "`M  V30 `" with the two blank spaces after `M` and one blank space after `30`. Following this is a list of zero or more required positional values (posval). Optional values can follow that use a '`KEYWORD=value`' format. Items are separated by white space. There can also be white space preceding the first item. Trailing white space is ignored.

The value of a keyword can be a list containing two or more values:

```
KEYWORD=(N val1 val2 ... valN)
```

where `N` specifies the number of values that follow.

Values (`posval`, `value`, or `val1`, and so forth) can be strings. Strings that contain blank spaces or start with left parenthesis or double quote, must be surrounded by double quotes. A double quote can be entered literally by doubling it.

Each entry is one line of no more than 80 characters. To allow continuation when the 80-character line is too short, use a dash (-) as the last character. When read, the line is concatenated with the next line by removing the dash and stripping the initial "`M  V30`" from the following line. For example:

```
M  V30 10 20 30 "abc-
M  V30   def"
```

is read as:

```
M  V30 10 20 30 "abc  def"
```

Generally, each section of the molfile is enclosed in a *block* that consists of lines such as:

```
M  V30 BEGIN key [blockname]
...
M  V30 END key
```

The 'key' value defines the kind of block, for example, `CTAB`, `ATOM`, or `BOND`. Depending upon the type of block, there might or might not be values on the `BEGIN` line.

The Connection Table

The connection table contains core information used in all the CTfile types.

CTAB Block

A Ctab block defines the basic connection table:

```
M  V30 BEGIN CTAB [ctabname]
counts-line
atom-block
[bond-block]
[sgroup-block]
[3d-block]
[link-line]*
M  V30 END CTAB
[collection-block]
[rgroup-block]*
[template-block]
```

The atom block and counts line are required. The counts line, atom block, and bond block must appear in the order indicated. The Sgroup block, 3D block, and link lines can occur in any order after the atom and bond blocks.

Counts Line

A counts line is required, and must be first. It specifies the number of atoms, bonds, 3D objects, and Sgroups. It also specifies whether or not the CHIRAL flag is set. Optionally, the counts line can specify a regno (a number that could be used for molecule registration). The regno specification on the counts line is only used when the regno exceeds 999999 (the limit of the format in the molfile header line). The format of the counts line is:

```
M  V30 COUNTS na nb nsg n3d chiral [REGNO=regno]
```

where:

na	number of atoms
nb	number of bonds
nsg	number of Sgroups
n3d	number of 3D constraints
chiral	1 if molecule is chiral, 0 if not
regno	molecule or model regno

Atom Block

An atom block specifies all node information for the connection table. It must precede the bond block. It has the following format:

```
M  V30 BEGIN ATOM
M  V30 index type x y z aamap -
M  V30 [CHG=val] [RAD=val] [CFG=val] [MASS=val] -
M  V30 [VAL=val] -
M  V30 [HCOUNT=val] [STBOX=val] [INVRET=val] [EXACHG=val] -
M  V30 [SUBST=val] [UNSAT=val] [RBCNT=val] -
M  V30 [ATTCHPT=val] -
M  V30 [RGROUPS=(nvals val [val ...])] -
M  V30 [ATTCHORD=(nvals nbr1 val1 [nbr2 val2 ...])] -
M  V30 [CLASS=template_class] -
M  V30 [SEQID=sequence_id] -
. . .
M  V30 END ATOM
```

Meaning of values in the atom block

Field	Meaning	Values	Notes
index	Atom index	Integer > 0	Identifies atoms. The actual value of the index does not matter as long as each index is unique to each atom.
type	Atom type	Type = reserved atom *or* atom *or* [NOT] '['atom, atom,...']' where reserved atom = R# = Rgroup A = "any" atom Q = any atom but C or H * = "star" atom Atom = character string (For example, 'C' or 'Cl')	A character string. If the string contains white space, it must be quoted. It can be a single atom or an atom list enclosed in square brackets with an optional preceding NOT.
x y z	Atom coordinates		
aamap	Atom-atom mapping	0 = no mapping > 0 = mapped atom	Reaction property
CHG	Atom charge	Integer 0 = none (default)	-15 to +15. Default of 0 = uncharged atom.
RAD	Atom radical	0 = none (default) 1 = singlet 2 = doublet 3 = triplet	
CFG	Stereo configuration	0 = none (default) 1 = odd parity 2 = even parity 3 = either parity	
MASS	Atomic weight		Default = natural abundance A specified value indicates the absolute atomic weight of the designated atom.
VAL	Valence	Integer > 0 *or* 0 = none (default) -1 = zero	Abnormal valence, -1, 0, 1-14
HCOUNT	Query hydrogen count	Integer > 0 *or* 0 = not specified (default) -1 = zero	1 = H0, 2 = H1, 3 = H2, 4 = H3, 5 = H4 **[Query]** H0 means no H atoms allowed unless explicitly drawn. Hn means atom must have n or more H's in excess of explicit H's.
STBOX	Stereo box	0 = ignore the configuration of this double bond atom (default) 1 = consider the stereo configuration of this double bond atom	Both atoms of a double bond must be marked to search double bond stereochemistry. Alternatively, the STBOX bond property can be used.
INVRET	Configuration inversion	0 = none (default) 1 = configuration inverts 2 = configuration retained	Reaction property

EXACHG	Exact change	0 = property not applied (default) 1 = exact change as displayed in the reaction	Reaction property
SUBST	Query substitution count	Integer > 0 *or* 0 = not specified (default) -1 = none	Number of substitutions allowed: default of 0 = off, -1 = no substitution (s0),-2 = as drawn (s*); 1, 2, 3, 4, 5 = (s1) through (s5), 6 or more = (s6).
UNSAT	Query unsaturation flag	0 = not specified (default) 1 = unsaturated	
RBCNT	Query ring bond count	Integer > 0 *or* 0 = not specified (default) -1 = none	Number of ring bonds allowed: default of 0 = off, -1 = no ring bonds (r0),-2 = as drawn (r*); 2 = (r2), 3 = (r3), 4 or more = (r4).
ATTCHPT	Rgroup member attachment points	Attachment points on member: -1 = first and second site 1 = first site only 2 = second site only	
RGROUPS	nvals is the number of Rgroups that comprise this R# atom. val is the Rgroup number.	Integer > 0 Integer > 0	
ATTCHORD	nvals is the number of values that follow on the ATTCHORD line nbr1 is atom neighbor index #1, nbr2 is index #2 val1 is the attachment order for the nbr1 attachment.	Integer > 0	A list of atom neighbor index and atom neighbor value pairs that identify the attachment order information at an R# atom or template atom. This propery allows textual attachment ids. Only Rgroup atoms and collapsed template atoms may provide ATTACHORD information. It is an error to provide this information on explicit atom types. Rgroup atoms support only integer ATTACHORD values. Collapsed template atoms suppport text values (for example, Al,Br).
CLASS	This property provides the class information for a collapsed template atom.		Example: for a collapsed alanine template atom (AA/Ala), its type=Ala and CLASS=AA
SEQID	This property supports a positive integer value to capture residue sequence id information.		This property supports a positive integer value to capture residue sequence id information for a template.

Bond block

A bond block specifies all edge information for the connection table. It must precede the Sgroup or 3D blocks. Its format is:

```
M  V30 BEGIN BOND
M  V30 index type atom1 atom2 [CFG=val] [TOPO=val] [RXCTR=val]
[STBOX=val]
...
M  V30 END BOND
```

Meaning of values in the bond block

Field	Meaning	Values	Notes
index	Bond index	Integer > 0	The actual value of the index does not matter as long as all are unique.
type	Bond type	Integer: 1 = single 2 = double 3 = triple 4 = aromatic 5 = single or double 6 = single or aromatic 7 = double or aromatic 8 = any	Types 4 through 8 are for queries only.
atom1,**atom2**	Atom indexes	Integer > 0	Atom1 and Atom2 are bond end points.
CFG	Bond configuration	0 = none (default) 1 = up 2 = either 3 = down	
TOPO	Query property	0 = not specified (default) 1 = ring 2 = chain	
RXCTR	Reacting center status	0 = unmarked (default) -1 = not a reacting center 1 = generic reacting center Additional: 2 = no change 4 = bond made or broken 8 = bond order changes 12 =(4 + 8) bond made or broken and changes 5 = (4 + 1), 9 = (8 + 1), and 13 =(12 + 1) are also possible	

STBOX	Stereo box	0 = ignore the configuration of this double bond (default) 1 = consider the stereo configuration of this double bond	A double bond must be marked to search double bond stereochemistry

Link atom line

There is one link atom line for each link atom in the Ctab. A link atom line has the format:

```
M  V30 LINKNODE minrep maxrep nbonds inatom outatom [inatom
outatom...]
```

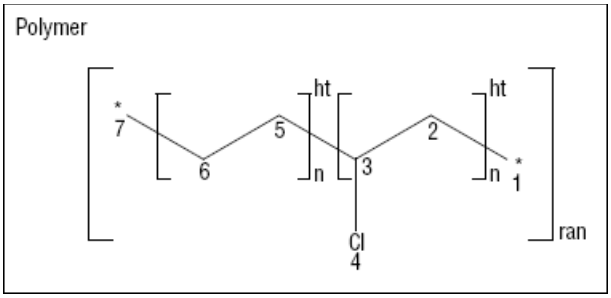
Meaning of values in link lines

Field	Meaning	Values	Notes
minrep	Minimum number of group repetitions.	1	For future expansion. Not currently used.
maxrep	Maximum number of group repetitions.	Integer > 0	
nbonds	Number of directed bonds defining the group.	nbonds = # of pairs of inatom-outatom tuples	The number of tuples is usually two (a pair), but can be one for link nodes with an attachment point.
inatom	Atom index of atom in the repeating group.	Integer > 0	
outatom	Atom index of atom bonded to inatom, but outside of repeating group.	Integer > 0	

Sgroup block

The Sgroup block contains general Sgroup information and information on each Sgroup structure as shown here. For the V2000 version, see "V2000 Molfile" on page 69.

Connection Table Organization of an Sgroup Structure




```
Polymer
          07189510252D 1   0.00374     0.00000     0
Figure 5, J. Chem. Inf. Comput. Sci., Vol 32, No. 3., 1992
  0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 7 6 3 0 0
M  V30 BEGIN ATOM
M  V30 1 * 2.9463 0.3489 0 0
M  V30 2 C 1.6126 1.1189 0 0
M  V30 3 C 0.2789 0.3489 0 0 CFG=3
M  V30 4 Cl 0.2789 -1.1911 0 0
M  V30 5 C -1.0548 1.119 0 0
M  V30 6 C -2.3885 0.349 0 0
M  V30 7 * -3.9246 1.147 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 1 2
M  V30 2 1 2 3
M  V30 3 1 3 4
M  V30 4 1 5 6
M  V30 5 1 5 3
M  V30 6 1 7 6
M  V30 END BOND
M  V30 BEGIN SGROUP
M  V30 1 SRU 5 ATOMS=(2 5 6) XBONDS=(2 5 6) BRKXYZ=(9 -0.6103 1.2969 0 -0.6103 -
M  V30 0.171 0 0 0 0) BRKXYZ=(9 -3.1565 0.185 0 -3.1565 1.311 0 0 0 0) -
M  V30 CONNECT=HT
M  V30 2 SRU 6 ATOMS=(3 2 3 4) XBONDS=(2 1 5) BRKXYZ=(9 2.2794 1.2969 0 2.2794 -
M  V30 0.1709 0 0 0 0) BRKXYZ=(9 -0.1657 0.171 0 -0.1657 1.2969 0 0 0 0) -
M  V30 CONNECT=HT
M  V30 3 COP 7 ATOMS=(7 1 2 3 4 5 6 7) BRKXYZ=(9 3.6382 1.6391 0 3.6382 -
M  V30 -1.7685 0 0 0 0) BRKXYZ=(9 -4.707 -1.7685 0 -4.707 1.6391 0 0 0 0) -
M  V30 SUBTYPE=RAN
M  V30 END SGROUP
M  V30 END CTAB
M  END
```

Header Block
Comment Line
Counts Line
Atom Block
Connection Table (Ctab)
Bond Block
Sgroup Block
Sgroup 1 Properties
Sgroup 2 Properties
Sgroup 3 Properties

For information on the comment line, see "V3000 Header" on page 35.

An Sgroup block defines all Sgroups in the molecule, including any `abbreviation Sgroup` (formerly called `abbrev` or `superatom`). The format of the Sgroup block is:

```
M  V30 BEGIN SGROUP
[M  V30 DEFAULT [CLASS=class] -]
M  V30 index type extindex -
M  V30 [ATOMS=(natoms atom [atom ...])] -
M  V30 [XBONDS=(nxbonds xbond [xbond ...])] -
M  V30 [CBONDS=(ncbonds cbond [cbond ...])] -
M  V30 [PATOMS=(npatoms patom [patom ...])] -
M  V30 [SUBTYPE=subtype] [MULT=mult] -
M  V30 [CONNECT=connect] [PARENT=parent] [COMPNO=compno] -
M  V30 [XBHEAD=(nxbonds xbond [xbond ...])] -
M  V30 [XBCORR=(nxbpairs xb1 xb2 [xb1 xb2 ...])] -
M  V30 [LABEL=label] -
M  V30 [BRKXYZ=(9 bx1 by1 bz1 bx2 by2 bz2 bx3 by3 bz3])* -
M  V30 [ESTATE=estate] [CSTATE=(4 xbond cbvx cbvy cbvz)]* -
M  V30 [FIELDNAME=fieldname] [FIELDINFO=fieldinfo] -
M  V30 [FIELDDISP=fielddisp] -
M  V30 [QUERYTYPE=querytype] [QUERYOP=queryop] -
M  V30 [FIELDDATA=fielddata] ... -
M  V30 [CLASS=class] -
M  V30 [SAP=(3 aidx lvidx id)]* -
M  V30 [BRKTYP=bracketType] -
...
M V30 [SEQID=sequence_id] -
M  V30 END SGROUP
```

The `DEFAULT` field provides a way to specify default values for keyword options. The same keyword options and values as defined in the following table.

Meaning of values in the Sgroup block

Field	Meaning	Values	Notes
index	Sgroup index	Integer > 0	The actual value of the index does not matter as long as all indexes are unique.
type	Sgroup type	String. Only first 3 letters are significant: SUPeratom MULtiple SRU MONomer COPolymer CROsslink MODification GRAft COMponent MIXture FORmulation DATa ANY GENeric	superatoms are now called abbreviation Sgroups
extindex	External index value	Integer => 0: If 0, positive integer assigned	Use 0 to autogenerate a number.

ATOMS	natoms is the number of atoms that define the Sgroup. atom is the atom index.	Integer > 0 Integer > 0	
XBONDS	nxbonds is the number of crossing bonds. xbond is the crossing-bond index.	Integer > 0 Integer > 0	
CBONDS	ncbonds is the number of containment bonds. cbond is the containment-bond index	Integer > 0 Integer > 0	Only used for data Sgroups.
PATOMS	npatom is the number of paradigmatic repeating unit atoms. patom is the atom index of an atom in the paradigmatic repeating unit for a multiple group.	Integer > 0	This field might become obsolete in future versions.
SUBTYPE	subtype is the Sgroup subtype.	String. Only the first 3 letters are significant: ALTernate RANdom BLOck	
MULT	mult is the multiple group multiplier.	Integer > 0	
CONNECT	connect is the connectivity.	String values are as follows: EU (default) HH HT	The default, if missing, is EU.
PARENT	parent is the parent Sgroup index.	Integer > 0	
COMPNO	compno is the component order number.	Integer > 0	
XBHEAD	nxbonds is the number of crossing bonds that cross the “head” bracket.	Integer > 0	
	xbond is the crossing-bond index.	Integer > 0	If XBHEAD is missing, no bonds are paired as the head or tail of the repeating unit

XBCORR	nxbpairs	2 x the number of pairs of crossing-bond correspondence, that is, the number of values in list.	
	xb1 - xb2 is the pairs of crossing-bond correspondence, that is, xb1 connects to xb2.	Integer > 0	
LABEL	label is the display label for this Sgroup.	String	For example, the abbreviation Sgroup name
BRKXYZ	bx1 - bz3 are the double (X,Y,Z) display coordinates in each bracket.	Angstroms	By specifying 3 triples, the format allows a 3D display. However, only the first two (X, Y) coordinates are currently used. The Z value and last (X, Y) coordinates are currently ignored and should be set to zero.
ESTATE	estate is the expanded display state information for abbreviation Sgroups.	String E = expanded abbreviation Sgroup or multiple group	Only abbreviation Sgroups and multiple groups (shortcuts) in an expanded internal state are supported. This field defines whether a abbreviation Sgroup or multiple group is displayed as expanded or contracted. This field might become obsolete in future versions.
CSTATE	xbond is the crossing bond of the expanded abbreviation Sgroup.	Integer > 0	Display vector information for the contracted abbreviation Sgroup.
	cbvx - cvbz is the vector to contracted abbreviation Sgroup.	Angstroms	Only present for expanded abbreviation Sgroups. One CSTATE entry per crossing bond.
FIELDNAME	fieldname is the name of data field for Data Sgroup.	String	Symyx internal namespace - do not use: data Sgroups having a fieldname that starts with SMMX: (case insensitive). Denotes data that does not differentiate structures.
FIELDINFO	fieldinfo is the program-specific field information.	Free-format string	Example: “<type> <units/format>”
FIELDDISP	fielddisp is the Data Sgroup field display information.	Free-format string	This string is interpreted by V3000 as identical to V2000 appendix for Data Sgroup display (‘M SDD’) except for the index value.
QUERYTYPE	querytype is the type of query or no query if missing.	String ‘ ’ = not a query (default) ‘MQ’ = legacy ‘IQ’ = ISIS query <p>Q’ = <program> query	
QUERYOP	queryop is the query operator.	String. query operator	Example: “=” or “LIKE”
FIELDDATA	fielddata is the query or field data.	Free-format string	Only one entry per query, but can be more than one for actual data. The order of the entries is important.

CLASS	class is the character string for abbreviation Sgroup class.	String	Example: PEPTIDE
SAP	aidx is the index of attachment point or potential attachment point atom.	Integer > 0	
	lvidx is the index of leaving atom.	Allowed integers are: 0 = none or implied H 'aidx' = atom index number of attachment point atom > 0 = atom index number of atom bonded to 'aidx'	
	id is the attachment identifier.	String (two chars in V2000)	There must be multiple entries if the abbreviation Sgroup has more than one attachment point. The order of the entries defines the order of the attachment points. SAP entries might or might not include the actual attachment points, depending on the particular abbreviation Sgroup and its representation.
BRKTYP	bracketType is the displayed bracket style.	Allowed values for this string are: BRACKET (default) PAREN	
SEQID	This property supports a positive integer value to capture residue sequence id information.	* SEQID	

Collection block

A collection block specifies all collection information for objects in the current connection table context. Collection blocks must be provided after the blocks that define the objects included in the collection to minimize the amount of forward object references that must be maintained by the file reader.

```
M  V30 BEGIN COLLECTION
[M  V30 DEFAULT -]M  V30 name/subname -
M  V30 [ATOMS=(natoms atom [atom ...])] -
M  V30 [BONDS=(nbonds bond [bond ...])] -
M  V30 [SGROUPS=(nsgroups sgrp [sgrp ...])] -
M  V30 [OBJ3DS=(nobj3ds obj3d [obj3d ...])] -
M  V30 [MEMBERS=(nmembers member [member ...])] -
M  V30 [RGROUPS=(nrgroups rgroup [rgroup ...])] -
...
M  V30 END COLLECTION
```

Meaning of values in the collection block

Field	Meaning	Value	Notes
name/subname	Collection id	Nonblank string	
ATOMS	natoms is the number of atoms included in the collection	Integer > 0	
	atom is the atom index	Integer > 0	
BONDS	nbonds is the number of bonds included in the collection	Integer > 0	
	bond is the bond index	Integer > 0	
SGROUPS	nsgroups is the number of Sgroups included in the collection	Integer > 0	
	sgrp is the Sgroup index	Integer > 0	
OBJ3DS	nobj3ds is the number of 3D features included in the collection	Integer > 0	
	obj3d is the 3D feature index	Integer > 0	
MEMBERS	nmembers is the number of members included in the collection	Integer > 0	
	member is the member identifier	ROOT or RrMm	r > 0, m > 0
RGROUPS	nrgroups is the number of Rgroups included in the collection	Integer > 0	
	rgroup is the Rgroup identifier	Rr	r > 0

Collections naming must meet the following criteria.

- A two-part name (name and subname) is supported for collections on import and export. The subname designation is required. The name is the unique identifier for the collection contents.
- All collections of the same name are presumed to indicate various pieces of the same collection, which can be provided in one or more `COLLECTION` block entries provided within various subblocks of the full connection table.
- Collection names and subnames are not case sensitive, contain only printable characters, and begin with an alphabetic character. The start of the name cannot be `MDL` (case insensitive).
- Collection objects are presumed to be unordered, and no preservation of input order is necessary or required on subsequent file writes.
- Future enhancements to the collection block information will provide ordered collection support.

The default delimiter for the collection name is the forward-slash (/) character. For example:

```
surfactiveAgent/agent001
```

where the name is `surfactiveAgent` and the subname is `agent001`

A non-default delimiter is specified by using a non-alphabet character before the name. The following example specifies hypen (-) as the delimiter and uses that delimiter for both name and subname:

```
-surfactiveAgent-agent001-
```

If the first character is is a quotation mark ("), it is presumed that the string that represents the collection name is within quotation marks due to the presence of spaces. For example, "surfactive agent".

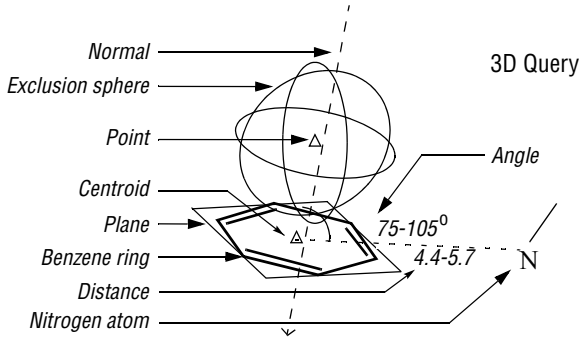
The name MDLV30 is a reserved name used to designate internal collections. User-specified collections can use any other arbitrary naming conventions. The default action for all V3000 readers and writers provided by Symyx is to preserve persistent collections and internal collections for CTfile import/export operations. Internal collections can also be preserved if their contents have been validated as correct input for the specified internal representation. There is no implied validation for user collections other than requiring the collection to refer to valid objects. Collections cannot contain other collections in their definition.

The default action on registration is to strip user collection information without error, but possibly with status or warning messages being issued. The internal collection types are:

MDLV30/HILITE	Highlighting collection. Default action for renderers is to provide a "highlighted" display of the specified objects. All object types are allowed in this collection: atoms, bonds, Sgroups, 3Dfeatures, Rgroups, members, and components.
MDLV30/STEABS	Absolute stereochemistry collection. This collection defines the set of stereocenters in the structure that have absolute configurations. This is an atom collection.
MDLV30/STERAC*n*	"Racemic" stereochemistry collection (*n* >0). This collection defines a set of stereogenic centers whose relative configuration is known. A mixture of the two enantiomeric relative configurations is present. This is an atom collection.
MDLV30/STEREL*n*	Relative stereochemistry collection (*n*>0) that is non-tetrahedral. This collection defines a set of stereogenic centers whose relative configuration is known. Only one of the two enantiomeric relative configurations is present. There is no assumption about which of the two configurations is present. This is an atom collection.

3D block

The 3D block contains the three-dimensional information as shown below.




```
3D Query
          07189510253D 1   1.00000     0.00000     0
Figure 6, J. Chem. Inf. Comput. Sci., Vol 32, No. 3., 1992
  0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 8 7 0 7 0
M  V30 BEGIN ATOM
M  V30 1 C 1.0252 0.2892 1.1122 0
M  V30 2 C -0.4562 0.6578 1.3156 0
M  V30 3 C -1.4813 0.3687 0.2033 0
M  V30 4 C -1.0252 -0.2892 -1.1122 0
M  V30 5 C 0.4562 -0.6578 -1.3156 0
M  V30 6 C 1.4813 -0.3687 -0.2033 0
M  V30 7 N 4.1401 -0.1989 1.3456 0
M  V30 8 C 4.6453 0.5081 1.7417 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 1 2
M  V30 2 2 2 3
M  V30 3 1 3 4
M  V30 4 2 4 5
M  V30 5 1 5 6
M  V30 6 2 6 1
M  V30 7 1 7 8
M  V30 END BOND
M  V30 BEGIN OBJ3D
M  V30 1 -7 6 "" 0 0 BASIS=(3 6 4 2)
M  V30 2 -5 13 "" 0 0 BASIS=(6 1 2 3 4 5 6)
M  V30 3 -8 7 "" 0 0 BASIS=(2 O3D.1 O3D.2)
M  V30 4 -3 6 "" -2 0 BASIS=(2 O3D.1 O3D.3) PNTDIR=1
M  V30 5 -16 12 "" 1.5 0 BASIS=(1 O3D.4) UNCONNOK=1
M  V30 6 -12 10 "" 75 105 BASIS=(3 O3D.4 O3D.1 7)
M  V30 7 -9 3 "" 4.4 5.7 BASIS=(2 7 O3D.1)
M  V30 END OBJ3D
M  V30 END CTAB
M  END
```

Header Block
Comment Line
Counts Line
Atom Block
Connection Table (Ctab)
Bond Block
3D object block

Blocks not used in this Ctab: Sgroup block, Rgroup block

For information on the comment line, see "V3000 Header" on page 35.

A 3D block specifies information for all 3D objects in the connection table. It must follow the atom and bond blocks. As in V2000 molfiles, there can be only one fixed-atom constraint.

The format of the 3D block is as follows:

```
M  V30 BEGIN OBJ3D
M  V30 index type color name value1 value2 -
M  V30 BASIS=(nbvals bval [bval ...]) -
M  V30 [ALLOW=(nvals val [val ...])] [PNTDIR=val] [ANGDIR=val] -
M  V30 [UNCONNOK=val] [DATA=strval] -
M  V30 [COMMENT=comment]
...
M  V30 END OBJ3D
```

Meaning of values in the 3D block

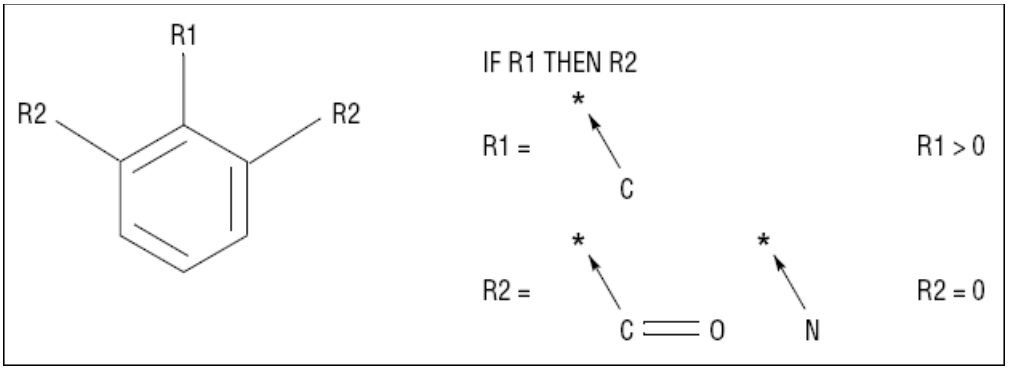
Field	Meaning	Values	Notes
index	3D object index	Integer > 0	The actual value of the index does not matter as long as all indexes are unique. However, extremely large numbers used as indexes can cause the program to fail to allocate memory for the correspondence array.
type	Object type	Integer < 0 for geometric constraints and for data constraints Integer > 0 are field IDs	
color	Color value	Integer > 0	
name	Object name or, for data query, the field name.	String	
value1	Distance, radius, deviation, or minimum value.	Floating point, value1 = 0 if constraint has no floating values	
value2	Maximum value for range constraints.	Floating point, value2 = 0 if not a range constraint	
BASIS	nbvals is the number of objects in basis.	Integer > 0	
	bval is the atom number or 3D object index	Integer or O3D.integer	For objects where order is important, for example, in an angle constructed from three points, the order must be the same as in V2000 molfiles. See "V2000 Connection Table [CTAB]" on page 43.
ALLOW	nvals is the number of atoms allowed in an exclusion sphere.	Integer > 0	
	val is the atom number.	Integer > 0	
PNTDIR		0 = point has no direction 1 = point has direction	
ANGDIR		0 = dihedral angle has no direction 1 = dihedral angle has direction	
UNCONNOK		0 = unconnected atoms are not OK 1 = unconnected atoms are OK	
DATA	strval is the data query string	String	
COMMENT	string comment	String.	0 - 32 characters

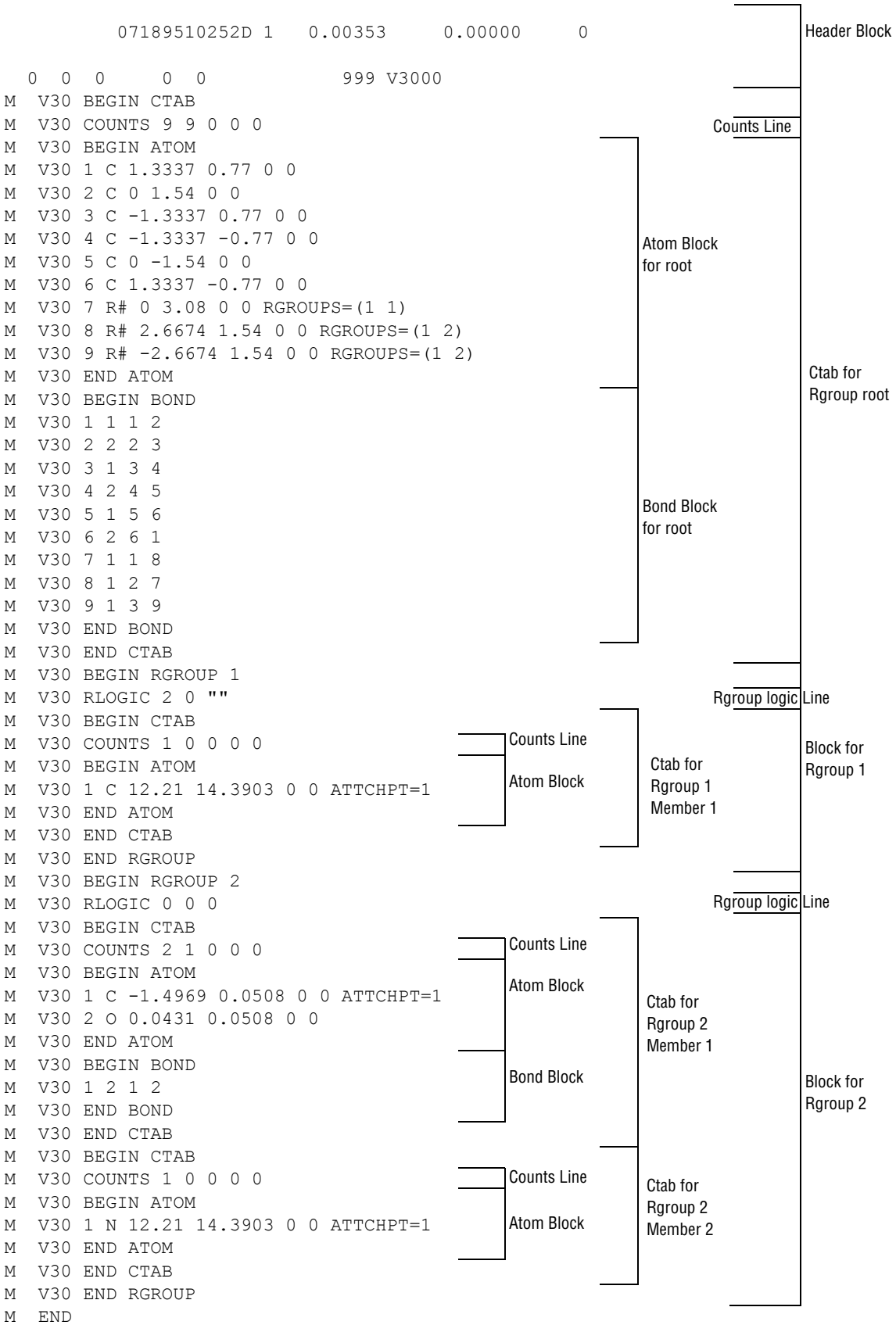
The Rgroup

Rgroup block

The Rgroup query shown below corresponds to the Rgroup file that follows it.

Connection table organization of an Rgroup query




07189510252D 1 0.00353 0.00000 0
0 0 0 0 0 999 V3000
M V30 BEGIN CTAB
M V30 COUNTS 9 9 0 0 0
M V30 BEGIN ATOM
M V30 1 C 1.3337 0.77 0 0
M V30 2 C 0 1.54 0 0
M V30 3 C -1.3337 0.77 0 0
M V30 4 C -1.3337 -0.77 0 0
M V30 5 C 0 -1.54 0 0
M V30 6 C 1.3337 -0.77 0 0
M V30 7 R# 0 3.08 0 0 RGROUPS=(1 1)
M V30 8 R# 2.6674 1.54 0 0 RGROUPS=(1 2)
M V30 9 R# -2.6674 1.54 0 0 RGROUPS=(1 2)
M V30 END ATOM
M V30 BEGIN BOND
M V30 1 1 1 2
M V30 2 2 2 3
M V30 3 1 3 4
M V30 4 2 4 5
M V30 5 1 5 6
M V30 6 2 6 1
M V30 7 1 1 8
M V30 8 1 2 7
M V30 9 1 3 9
M V30 END BOND
M V30 END CTAB
M V30 BEGIN RGROUP 1
M V30 RLOGIC 2 0 ""
M V30 BEGIN CTAB
M V30 COUNTS 1 0 0 0 0
M V30 BEGIN ATOM
M V30 1 C 12.21 14.3903 0 0 ATTCHPT=1
M V30 END ATOM
M V30 END CTAB
M V30 END RGROUP
M V30 BEGIN RGROUP 2
M V30 RLOGIC 0 0 0
M V30 BEGIN CTAB
M V30 COUNTS 2 1 0 0 0
M V30 BEGIN ATOM
M V30 1 C -1.4969 0.0508 0 0 ATTCHPT=1
M V30 2 O 0.0431 0.0508 0 0
M V30 END ATOM
M V30 BEGIN BOND
M V30 1 2 1 2
M V30 END BOND
M V30 END CTAB
M V30 BEGIN CTAB
M V30 COUNTS 1 0 0 0 0
M V30 BEGIN ATOM
M V30 1 N 12.21 14.3903 0 0 ATTCHPT=1
M V30 END ATOM
M V30 END CTAB
M V30 END RGROUP
M END
Header Block
Counts Line
Atom Block for root
Bond Block for root
Ctab for Rgroup root
Rgroup logic Line
Counts Line
Atom Block
Ctab for Rgroup 1 Member 1
Block for Rgroup 1
Rgroup logic Line
Counts Line
Atom Block
Bond Block
Ctab for Rgroup 2 Member 1
Counts Line
Atom Block
Ctab for Rgroup 2 Member 2
Block for Rgroup 2

For information on the comment line, see "V3000 Header" on page 35.

An Rgroup block defines one Rgroup. Each Ctab block specifies one member.

```
M  V30 BEGIN RGROUP rgroup-number
[rgroup-logic-line]
ctab-block
[ctab-block]*
M  V30 END RGROUP
```

Meaning of values in the Rgroup block

Field	Meaning	Values	Notes
rgroup-number	Index of this rgroup	Integer > 0	

Rgroup logic lines

There is zero or one Rgroup logic line for each Rgroup in the molecule. If present, the Rgroup logic line specifies if-then logic between Rgroups, the convention about unfilled valence sites, and the Rgroup occurrence information. Its format is:

```
M  V30 RLOGIC thenR RestH Occur
```

Meaning of values in Rgroup logic line

Field	Meaning	Values	Notes
thenR	Number of a "then" Rgroup	0 = none (default)	
RestH	Attachment(s) at Rgroup position	0 = off, that is, any molecule fragment at any unsatisfied Rgroup location (default) 1 = only hydrogen or a member of Rgroup is allowed	
Occur	String specifying number (range) of Rgroup occurrence sites that need to be satisfied.	String '> 0' = default	Examples: 1,3,5,7 1,2-6,8 >0

Template block

A Template block can be used to represent large biomolecules by defining one or more template definitions.

```
M V30 BEGIN TEMPLATE
[template-definition]*
M V30 END TEMPLATE
```

where * means 1 or many template definitions.

A template definition begins with a line that defines template properties, followed immediately with a single ctab block to provide the ctab definition:

```
M V30 TEMPLATE index [ name | class/name[/alternate_name1[...] ]
[ COMMENT=template_comment ]
M V30 BEGIN CTAB
...
M V30 END CTAB
```

Template definitions can be present only at the root ctab of a CTfile, not within Rgroup blocks or within any other multi-ctab blocks.

Example of a V3000 molfile representing a template-based biologic

One advantage of template atoms is that their repetition is represented in a compact manner. For example, the repetition of alanine glycine as AGAG takes only a few more lines than representing AG.

```
  SMMXDraw06041015442D

  0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 6 5 0 0 0
M  V30 BEGIN ATOM
M  V30 1 Ala 2.4125 -12.9167 0 0 CLASS=AA ATTCHORD=(2 2 Br) SEQID=1
M  V30 2 Gly 3.57 -12.9138 0 0 CLASS=AA ATTCHORD=(4 3 Br 1 Al) SEQID=2
M  V30 3 Ala 4.7275 -12.9138 0 0 CLASS=AA ATTCHORD=(4 2 Al 4 Br) SEQID=3
M  V30 4 Gly 5.8849 -12.9138 0 0 CLASS=AA ATTCHORD=(4 5 Br 3 Al) SEQID=4
M  V30 5 Ala 7.0424 -12.9138 0 0 CLASS=AA ATTCHORD=(4 4 Al 6 Br) SEQID=5
M  V30 6 Gly 8.1999 -12.9138 0 0 CLASS=AA ATTCHORD=(2 5 Al) SEQID=6
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 2 1
M  V30 2 1 3 2
M  V30 3 1 4 3
M  V30 4 1 5 4
M  V30 5 1 6 5
M  V30 END BOND
M  V30 END CTAB
M  V30 BEGIN TEMPLATE
M  V30 TEMPLATE 1 AA/Ala/A/
M  V30 BEGIN CTAB
M  V30 COUNTS 7 6 3 0 0
M  V30 BEGIN ATOM
M  V30 1 O 6.6266 -2.0662 0 0
M  V30 2 H 5.0016 -2.0876 0 0
M  V30 3 N 5.1358 -2.0784 0 0 CFG=3
M  V30 4 C 5.7844 -1.5983 0 0 CFG=2
M  V30 5 C 6.4753 -2.0653 0 0
M  V30 6 O 6.4753 -2.8977 0 0
M  V30 7 C 5.7844 -0.7662 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 3 4
M  V30 2 1 4 5
M  V30 3 2 5 6
M  V30 4 1 4 7 CFG=1
M  V30 5 1 3 2
M  V30 6 1 5 1
M  V30 END BOND
M  V30 BEGIN SGROUP
M  V30 1 SUP 1 ATOMS=(1 1) XBONDS=(1 6) BRKXYZ=(9 7.02 -2.26 0 7.02 -1.85 0 -
M  V30 0 0 0) CSTATE=(4 6 -0.82 -0.01 0) LABEL=OH CLASS=LGRP
M  V30 2 SUP 2 ATOMS=(1 2) XBONDS=(1 5) BRKXYZ=(9 4.58 -1.87 0 4.6 -2.28 0 -
M  V30 0 0 0) CSTATE=(4 5 0.8 0.02 0) LABEL=H CLASS=LGRP
M  V30 3 SUP 3 ATOMS=(5 3 4 5 6 7) XBONDS=(2 5 6) BRKXYZ=(9 3.95 -3.33 0 3.95 -
M  V30 -0.38 0 0 0 0) CSTATE=(4 5 -0.8 -0.02 0) CSTATE=(4 6 0.82 0.01 -
M  V30 0) LABEL=A CLASS=AA SAP=(3 3 2 Al) SAP=(3 5 1 Br)
M  V30 END SGROUP
M  V30 BEGIN COLLECTION
M  V30 MDLV30/STEABS ATOMS=(1 4)
M  V30 END COLLECTION
M  V30 END CTAB
M  V30 TEMPLATE 2 AA/Gly/G/
M  V30 BEGIN CTAB
```

```
M  V30 COUNTS 6 5 3 0 0
M  V30 BEGIN ATOM
M  V30 1 N 3.676 -12.5274 0 0 CFG=3
M  V30 2 C 4.2675 -12.095 0 0
M  V30 3 O 4.8932 -13.2691 0 0
M  V30 4 C 4.8904 -12.5161 0 0
M  V30 5 O 5.1042 -12.5167 0 0
M  V30 6 H 3.4542 -12.5125 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 1 2
M  V30 2 1 2 4
M  V30 3 2 4 3
M  V30 4 1 4 5
M  V30 5 1 1 6
M  V30 END BOND
M  V30 BEGIN SGROUP
M  V30 1 SUP 1 ATOMS=(1 5) XBONDS=(1 4) CSTATE=(4 4 -0.82 -0.01 0) LABEL=OH -
M  V30 CLASS=LGRP
M  V30 2 SUP 2 ATOMS=(4 1 2 3 4) XBONDS=(2 4 5) CSTATE=(4 4 0.82 0.01 0) -
M  V30 CSTATE=(4 5 -0.83 0.01 0) LABEL=G CLASS=AA SAP=(3 4 5 Br) -
M  V30 SAP=(3 1 6 Al)
M  V30 3 SUP 3 ATOMS=(1 6) XBONDS=(1 5) CSTATE=(4 5 0.83 -0.01 0) LABEL=H -
M  V30 CLASS=LGRP
M  V30 END SGROUP
M  V30 END CTAB
M  V30 END TEMPLATE
M  END
```

Molfile

Overview

Your code for writing molfiles should be able to export structure information in the current, preferred V3000 format if any structural features are present that cannot be defined in the legacy V2000 format. The V3000 molfile and V3000 rxnfile formats support the latest features of Symyx chemical representation. See "Molfile in V3000 Format - Advantages Over V2000" on page 6. V3000 is both a superset of V2000 but a *different* format. For information on the earlier V2000 format, see V2000 Connection Table [CTAB].

A "no-structure"

- Is a placeholder structure that has no contents, represented by a counts line with many zeros
- Can be flagged with `the V3000` or the `V2000` version stamp.

```
[name or blank]
     06031011012D 1   1.00000     0.00000     0
[comment or blank]
  0  0  0     0  0            999 V2000
M  END
```

```
[name or blank]
     06031011022D 1   1.00000     0.00000     0
[comment or blank]
  0  0  0     0  0            999 V3000
M  V30 BEGIN CTAB
M  V30 COUNTS 0 0 0 0 0
M  V30 END CTAB
M  END
```

Example of molfile - alanine

```
1
2    SMMXDraw06081014582D
3
4    0  0  0     0  0            999 V3000
5  M  V30 BEGIN CTAB
6  M  V30 COUNTS 6 5 0 0 0
7  M  V30 BEGIN ATOM
8  M  V30 1 O 12.4491 -13.9583 0 0
9  M  V30 2 C 13.0396 -17.0269 0 0
10 M  V30 3 O 14.2207 -14.9811 0 0
11 M  V30 4 C 13.0396 -14.9811 0 0
12 M  V30 5 C 12.4491 -16.0041 0 0 CFG=3
13 M  V30 6 N 11.268 -16.0041 0 0
14 M  V30 END ATOM
15 M  V30 BEGIN BOND
16 M  V30 1 1 4 1
17 M  V30 2 1 5 2
18 M  V30 3 2 4 3
19 M  V30 4 1 5 4
20 M  V30 5 1 6 5
21 M  V30 END BOND
22 M  V30 END CTAB
23 M  END
```

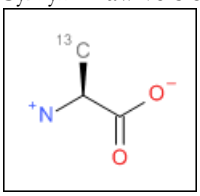
The structure of the molfile is:

Feature	Line(s)
Header Block	1 - 4
Comment Line	3
Connection Table (ctab)	5 - 26
Counts Line	6
Atom Block	7 - 14
Bond Block	15 - 25

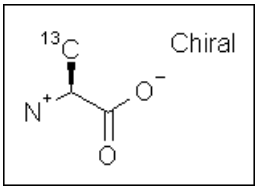
Note: This example does not have the following Ctab blocks: Sgroup, Rgroup, 3D, Template.

Different applications might render the same chemical stucture with some superficial differences.

Symyx Draw version of alanine:



ISIS/Draw version of alanine:



V3000 Header

Line 1:	Molecule name. This line is unformatted, but like all other lines in a molfile cannot extend beyond column 80. If no name is available, a blank line must be present. **Caution:** This line must not contain any of the reserved tags that identify any of the other CTAB file types such as `$MDL` (RGfile), `$$$$` (SDfile record separator), `$RXN` (rxnfile), or `$RDFILE` (RDfile headers).

Line 2:	This line has the format:
	`IIPPPPPPPPMMDDYYHHmmddSSssssssssssEEEEEEEEEEEERRRRRR`
	`(FORTRAN: A2<--A8--><---A10-->A2I2<--F10.5-><---F12.5--><-I6-> )`
	User's first and last initials (`I`), program name (`P`), date/time (`M/D/Y,H:m`), dimensional codes (`d`) such as 2D or 3D, scaling factors (`S`, `s`), energy (`E`) if modeling program input, internal registry number (`R`)
	The "dimensional code" is maintained explicitly. Thus "3D" really means 3D, although "2D" will be interpreted as 3D if any non-zero Z-coordinates are found.
	Note: A blank line can be substituted for line 2.

Line 3:	A line for comments. If no comment is available, a blank line must be present.
Line 4	This line contains the version number, V3000 The number of appendix lines is always written as 999, regardless of how many there actually are. See "Counts Line" on page 13. (For V2000, Line 4 is the no-structure counts line. See "The Counts Line" on page 45)

Note:

- The first 3 lines are for compatibility with legacy V2000 molfile readers.
- Unlike the V2000 molfile, the V3000 extended Rgroup molfile has the same header format as a non-Rgroup molfile.
- Do **not** create a molfile with a pre-V3000 Rgroup header (**"**`$MDL`**"**, and so forth) but with V3000 Ctab blocks. A pre-V3000 Rgroup molfile can only have embedded molfiles that are also pre-V3000 versions, for example, the version is either **"**`V2000`**"** or **"** **"**.

RGfiles (Rgroup file)

RGfile Overview

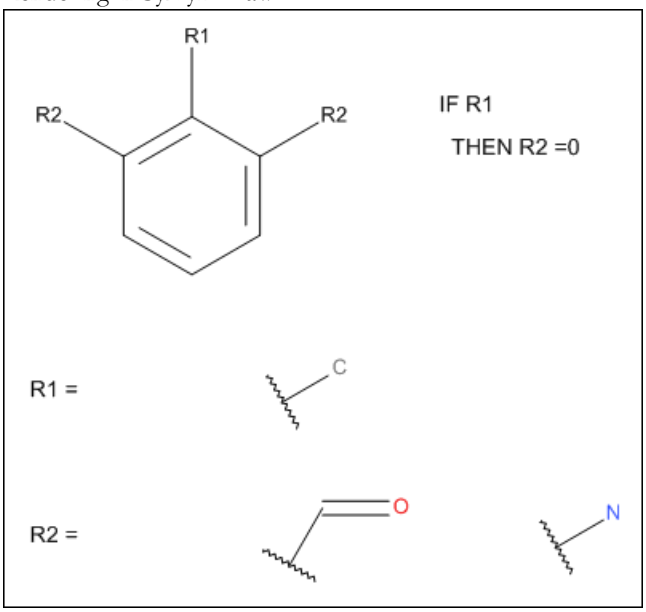
The format of an RGfile (Rgroup file) is shown below. (Rgroups are also known as Markush structures.)

In addition to the primary connection table (Ctab block) for the root structure, a Ctab block defines each member within each Rgroup. For an example of the header, see "Connection table organization of an Rgroup query" on page 27. For a description of the Rgroup, see "The Rgroup" on page 27.

The "Example of an RGfile (Rgroup file)" on page 38 corresponds to the following Rgroup structure.

Example of an RGfile (Rgroup file)

Rendering in Symyx Draw.



What follows is the RGfile that corresponds to the rendering above.

The structure of the RGfile is:

Feature	Lines
RGfile header Block	1 - 4
Ctab for Rgroup root	5 - 29
Block for Rgroup R1	30 - 38
Block for Rgroup R2	39 - 57

```
 1
 2    SMMXDraw06091016012D
 3
 4    0  0  0     0  0            999 V3000
 5  M  V30 BEGIN CTAB
 6  M  V30 COUNTS 9 9 0 0 0
 7  M  V30 BEGIN ATOM
 8  M  V30 1 C 11.5195 -5.2131 0 0
 9  M  V30 2 C 10.1858 -4.4431 0 0
10  M  V30 3 C 8.8521 -5.2131 0 0
11  M  V30 4 C 8.8521 -6.7531 0 0
12  M  V30 5 C 10.1858 -7.5231 0 0
13  M  V30 6 C 11.5195 -6.7531 0 0
14  M  V30 7 R# 10.1858 -2.9031 0 0 RGROUPS=(1 1)
15  M  V30 8 R# 12.8532 -4.4431 0 0 RGROUPS=(1 2)
16  M  V30 9 R# 7.5184 -4.4431 0 0 RGROUPS=(1 2)
17  M  V30 END ATOM
18  M  V30 BEGIN BOND
19  M  V30 1 1 1 2
20  M  V30 2 2 2 3
21  M  V30 3 1 3 4
22  M  V30 4 2 4 5
23  M  V30 5 1 5 6
24  M  V30 6 2 6 1
25  M  V30 7 1 1 8
26  M  V30 8 1 2 7
27  M  V30 9 1 3 9
28  M  V30 END BOND
29  M  V30 END CTAB
30  M  V30 BEGIN RGROUP 1
31  M  V30 RLOGIC 2 0 >0
32  M  V30 BEGIN CTAB
33  M  V30 COUNTS 1 0 0 0 0
34  M  V30 BEGIN ATOM
35  M  V30 1 C 13.0761 -9.3618 0 0 ATTCHPT=1
36  M  V30 END ATOM
37  M  V30 END CTAB
38  M  V30 END RGROUP
39  M  V30 BEGIN RGROUP 2
40  M  V30 RLOGIC 0 0 0
41  M  V30 BEGIN CTAB
42  M  V30 COUNTS 2 1 0 0 0
43  M  V30 BEGIN ATOM
44  M  V30 1 C 12.7367 -12.0793 0 0 ATTCHPT=1
45  M  V30 2 O 14.2767 -12.0793 0 0
46  M  V30 END ATOM
47  M  V30 BEGIN BOND
48  M  V30 1 2 1 2
49  M  V30 END BOND
50  M  V30 END CTAB
51  M  V30 BEGIN CTAB
52  M  V30 COUNTS 1 0 0 0 0
53  M  V30 BEGIN ATOM
54  M  V30 1 N 18.4231 -12.1956 0 0 ATTCHPT=1
55  M  V30 END ATOM
56  M  V30 END CTAB
57  M  V30 END RGROUP
58  M  END
```

Rxnfile

Overview

Rxnfiles contain structural data for the reactants and products of a reaction. This chapter covers the preferred V3000 reaction file format. For the V2000 version of this reaction file format, see "V2000 Rxnfile Overview" on page 71.

Header Block

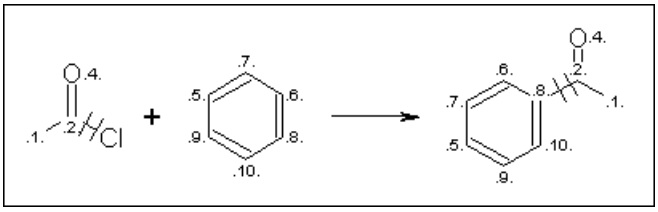
Line 1:	`$RXN` in the first position on this line identifies the file as a reaction file. After one blank space, `V3000` follows.
Line 2:	A line for the reaction name. If no name is available, a blank line must be present.

Line 3:	User's initials (I), program name and version (`P`), date/time (`M/D/Y,H:m`), and reaction registry number (`R`). This line has the format:
	`IIIIIIPPPPPPPPPMMDDYYYYHHmmRRRRRRR`
	`(FORTRAN: <-A6-><---A9--><---A12----><--I7-> )`
	A blank line can be substituted for line 3.
	If the internal registry number is more than 7 digits long, it is stored in an `"M  REG"` line (see "Large REGNO" on page 58).

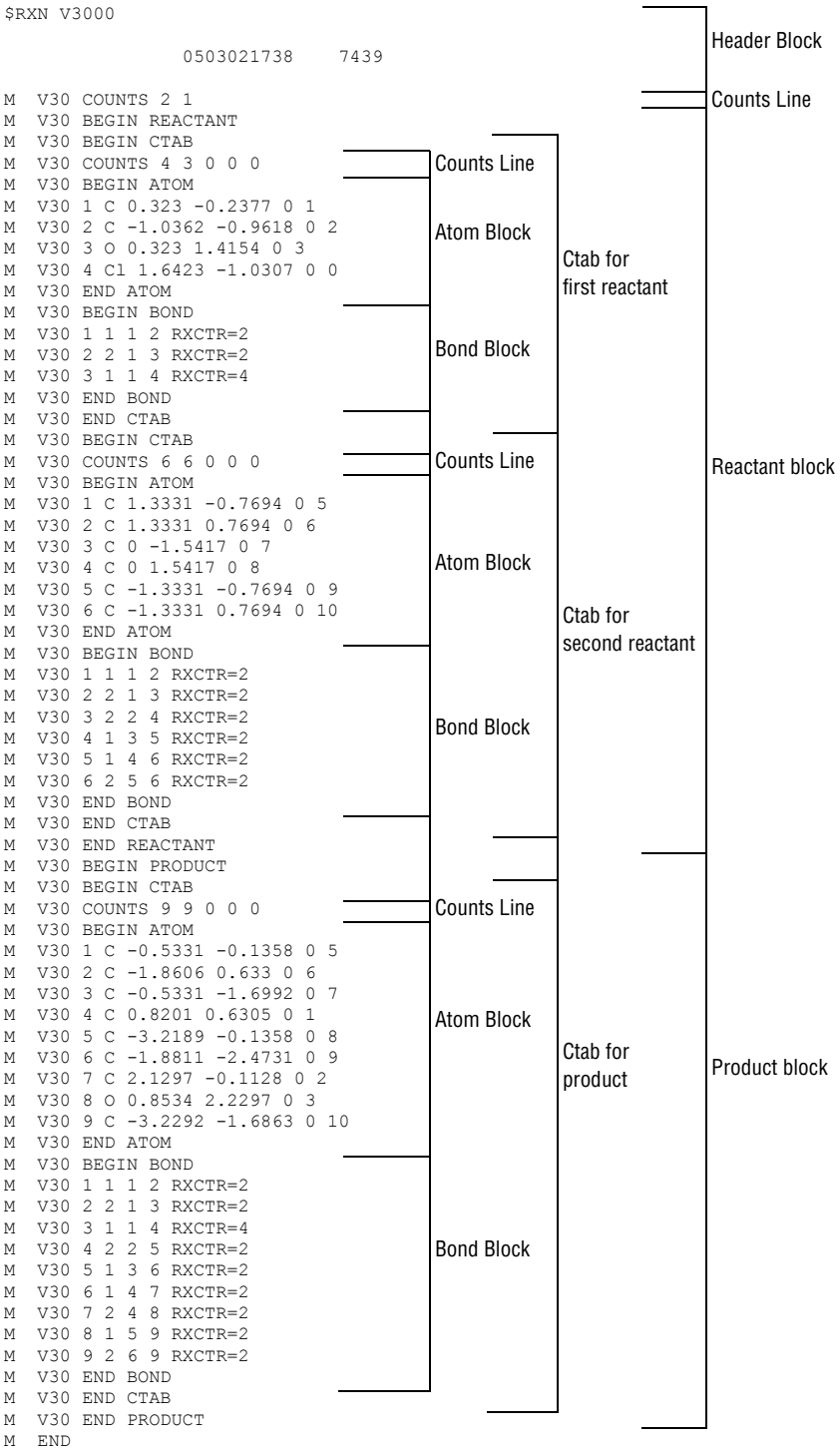
Line 4	A line for comments. If no comment is entered, a blank line must be present.

Counts line

Line 5: The counts line specifies the number of reactants and products. Two reactants and one product are indicated in line 5 of "Example V3000 Rxnfile for the acylation of benzene" on page 42.



Example V3000 Rxnfile for the acylation of benzene



V2000 Connection Table [CTAB]

Overview

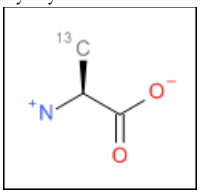
A connection table (Ctab) contains information describing the structural relationships and properties of a collection of atoms. The atoms can be wholly or partially connected by bonds. (An atom can also be an unconnected fragment.) Such collections might, for example, describe molecules, molecular fragments, substructures, substituent groups, polymers, alloys, formulations, mixtures, and unconnected atoms. The connection table is fundamental to all of Symyx's file formats.

This chapter describes the legacy V2000 format. The current V3000 format supports more features. See "Connection Table [CTAB]" on page 11.

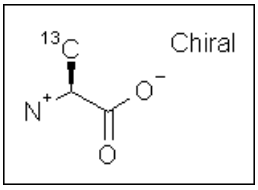
Example: Alanine in V2000 format

Different applications might render the same chemical stucture with some superficial differences.

Symyx Draw version of alanine:



ISIS/Draw version of alanine:



The connection table of alanine:

```
1
2    SMMXDraw06021015152D
3
4    6  5  0  0  1  0  0  0  0  0999 V2000
5      9.7434  -15.8027    0.0000 N   0  3  0  0  0  0  0  0  0  0  0  0
6     10.7663  -15.2121    0.0000 C   0  0  2  0  0  0  0  0  0  0  0  0
7     11.7891  -15.8027    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
8     12.8120  -15.2121    0.0000 O   0  5  0  0  0  0  0  0  0  0  0  0
9     11.7891  -16.9838    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
10    10.7663  -14.0310    0.0000 C   1  0  0  0  0  0  0  0  0  0  0  0
11   1  2  1  0  0  0  0
12   2  3  1  0  0  0  0
13   3  4  1  0  0  0  0
14   3  5  2  0  0  0  0
15   2  6  1  1  0  0  0
16 M  CHG  2   1   1   4  -1
17 M  ISO  1   6  13
18 M  END
```

The structure of the molfile is:

Feature	Line(s)
Header Block	1 - 3
Comment Line	3
Connection Table (ctab)	4 - 18
Counts Line	4
Atom Block	5 - 10
Bond Block	11- 15
Properties Block	16 - 18

This example does not have the following Ctab blocks: Atom List, Stext.

Ctab block format for V2000

The format for a Ctab block is:

Counts line:	Important specifications here relate to the number of atoms, bonds, and atom lists, the chiral flag setting, and the Ctab version.
Atom block:	Specifies the atomic symbol and any mass difference, charge, stereochemistry, and associated hydrogens for each atom.
Bond block:	Specifies the two atoms connected by the bond, the bond type, and any bond stereochemistry and topology (chain or ring properties) for each bond.
Atom list block:	Identifies the atom (number) of the list and the atoms in the list.
Stext (structural text descriptor) block:	Used by ISIS/Desktop programs.
Properties block:	Provides for future expandability of Ctab features, while maintaining compatibility with earlier Ctab configurations.

The detailed format for each block outlined above follows:

Note: A blank *numerical* entry on any line should be read as "0" (zero). Spaces are significant and correspond to one or more of the following:

- Absence of an entry
- Empty character positions within an entry
- Spaces between entries; single unless specifically noted otherwise

The Counts Line

```
aaabbblllfffcccsssxxxrrrpppiiimmmvvvvvv
```

Where:

aaa	= number of atoms (current max 255)*	**[Generic]**
bbb	= number of bonds (current max 255)*	**[Generic]**
lll	= number of atom lists (max 30)*	**[Query]**
fff	= (obsolete)	
ccc	= chiral flag: 0=not chiral, 1=chiral	**[Generic]**
sss	= number of stext entries	**[ISIS/Desktop]**
xxx	= (obsolete)	
rrr	= (obsolete)	
ppp	= (obsolete)	
iii	= (obsolete)	
mmm	= number of lines of additional properties, including the `M  END` line. No longer supported, the default is set to 999.	**[Generic]**

* These limits apply to the ISIS/Host Reaction Gateway, but not to the Symyx ISIS/Host Molecule Gateway or ISIS/Desktop.

For example, the counts line in the Ctab shown above has six atoms, five bonds, the CHIRAL flag *on*, and three lines in the properties block:

```
  6  5  0  0  1  0               3 V2000
```

A V2000 no-structure might have a counts line (line 4) as follows:

```
  0  0  0  0  0  0  0  0  0  0  0999 V2000
```

The Atom Block

The Atom Block is made up of atom lines, one line per atom with the following format:

```
xxxxx.xxxxyyyyy.yyyyzzzzz.zzzz aaaddcccssshhhbbbvvvHHHrrriiimmmnnneee
```

where the values are described in the following table:

Figure 2 Meaning of values in the atom block

Field	Meaning	Values	Notes
x y z	atom coordinates		**[Generic]**
aaa	atom symbol	entry in periodic table or L for atom list, A, Q, * for unspecified atom, and LP for lone pair, or R# for Rgroup label	**[Generic, Query, 3D, Rgroup]**
dd	mass difference	-3, -2, -1, 0, 1, 2, 3, 4 (0 if value beyond these limits)	**[Generic]** Difference from mass in periodic table. Wider range of values allowed by `M ISO` line, below. Retained for compatibility with older Ctabs, `M ISO` takes precedence.
ccc	charge	0 = uncharged or value other than these, 1 = +3, 2 = +2, 3 = +1, 4 = doublet radical, 5 = -1, 6 = -2, 7 = -3	**[Generic]** Wider range of values in `M CHG` and `M RAD` lines below. Retained for compatibility with older Ctabs, `M CHG` and `M RAD` lines take precedence.
sss	atom stereo parity	0 = not stereo, 1 = odd, 2 = even, 3 = either or unmarked stereo center	**[Generic]** Ignored when read.
hhh	hydrogen count + 1	1 = H0, 2 = H1, 3 = H2, 4 = H3, 5 = H4	**[Query]** H0 means no H atoms allowed unless explicitly drawn. Hn means atom must have n or more H's in excess of explicit H's.
bbb	stereo care box	0 = ignore stereo configuration of this double bond atom, 1 = stereo configuration of double bond atom must match	**[Query]** Double bond stereochemistry is considered during SSS only if both ends of the bond are marked with stereo care boxes.
vvv	valence	0 = no marking (default) (1 to 14) = (1 to 14) 15 = zero valence	**[Generic]** Shows number of bonds to this atom, including bonds to implied H's.

HHH	H0 designator	0 = not specified, 1 = no H atoms allowed	**[ISIS/Desktop]** Redundant with hydrogen count information. Might be unsupported in future releases of Symyx software.
rrr	Not used		
iii	Not used		
mmm	atom-atom mapping number	1 - number of atoms	**[Reaction]**
nnn	inversion/retention flag	0 = property not applied 1 = configuration is inverted, 2 = configuration is retained,	**[Reaction]**
eee	exact change flag	0 = property not applied, 1 = change on atom must be exactly as shown	[Reaction, Query]

With Ctab version V2000, the `dd` and `ccc` fields have been superseded by the `M ISO`, `M CHG`, and `M RAD` lines in the properties block, described below. For compatibility, all releases since ISIS 1.0:

- Write appropriate values in both places if the values are in the old range.
- Use the atom block fields if there are no M ISO, M CHG, or M RAD lines in the properties block.

Support for these atom block fields might be removed in future releases of Symyx software.

The Bond Block

The Bond Block is made up of bond lines, one line per bond, with the following format:

```
111222tttsssxxxrrrccc
```

and the values are described below:

Meaning of values in the bond block

Field	Meaning	Values	Notes
111	first atom number	1 - number of atoms	**[Generic]**
222	second atom number	1 - number of atoms	**[Generic]**
ttt	bond type	1 = Single, 2 = Double, 3 = Triple, 4 = Aromatic, 5 = Single or Double, 6 = Single or Aromatic, 7 = Double or Aromatic, 8 = Any	**[Query]** Values 4 through 8 are for SSS queries only.
sss	bond stereo	Single bonds: 0 = not stereo, 1 = Up, 4 = Either, 6 = Down, Double bonds: 0 = Use x-, y-, z-coords from atom block to determine cis or trans, 3 = Cis or trans (either) double bond	**[Generic]** The wedge (pointed) end of the stereo bond is at the first atom (Field 111 above)
xxx	not used		
rrr	bond topology	0 = Either, 1 = Ring, 2 = Chain	**[Query]** SSS queries only.
ccc	reacting center status	0 = unmarked, 1 = a center, -1 = not a center, Additional: 2 = no change, 4 = bond made/broken, 8 = bond order changes 12 = 4+8 (both made/broken and changes); 5 = (4 + 1), 9 = (8 + 1), and 13 = (12 + 1) are also possible	**[Reaction, Query]**

The Atom List Block [Query]

Note: Current Symyx applications use the `M  ALS` item in the properties block in place of the atom list block. The atom list block is retained for compatibility, but information in an `M  ALS` item supersedes atom list block information.

Made up of atom list lines, one line per list, with the following format:

```
aaa kSSSSn 111 222 333 444 555
```

where:

Field	Meaning
aaa	= number of atom (L) where list is attached
k	= T = [NOT] list, = F = normal list
n	= number of entries in list; maximum is 5
111...555	= atomic number of each atom on the list
S	= space

The Stext Block [ISIS/Desktop - not used in current products]

The Stext Block is made up of two-line entries with the following format:

```
xxxxx.xxxxyyyyy.yyyy
TTTT...
```

where:

Field	Meaning
x y	= stext coordinate
T	= stext text

The Properties Block

The Properties Block is made up of `mmm` lines of additional properties, where `mmm` is the number in the counts line described above. If a version stamp is present, `mmm` is ignored and the file is read until an `"M  END"` line is encountered.

Note: `mmm` is no longer supported and is set to 999 as the default.

Most lines in the properties block are identified by a prefix of the form `M  XXX` where two spaces separate the `M` and `XXX`. Exceptions are:

- `A  aaa, V  aaa vvvvvv`, and `G  aaappp`, which indicate the following ISIS/Desktop properties: atom alias, atom value, and group abbreviation (called residue in ISIS), respectively.
- `S  SKPnnn` which causes the next nnn lines to be ignored.

The prefix: `M  END` terminates the properties block.

Variables in the formats can change properties but keep the same letter designation. For example, on the Charge, Radical, or Isotope lines, the "uniformity" of the `vvv` designates a general property identifier. On Sgroup property lines, the `sss` uniformity is used as an Sgroup index identifier.

All lines that are not understood by the program are ignored.

The descriptions below use the following conventions for values in field widths of 3:

`n15`	number of entries on line; value = 1 to 15
`nn8`	number of entries on line; value = 1 to 8
`nn6`	number of entries on line; value = 1 to 6
`nn4`	number of entries on line; value = 1 to 4
`nn2`	number of entries on line; value = 1 or 2
`nn1`	number of entries on line; value = 1
`aaa`	atom number; value = (1 to number of atoms)

The format for the properties included in this block are described in what follows. The format shows one entry and ellipses (. . .) indicate additional entries.

Atom Alias [ISIS/Desktop - not used in current products]

```
A  aaa
x...
```

aaa:	Atom number
x:	Alias text

Atom Value [ISIS/Desktop - not used in current products]

```
V  aaa v...
```

aaa:	Atom number
v:	Value text

Group Abbreviation [ISIS/Desktop - not used in current products]

```
G  aaappp
x...
```

aaa:	Atom number
ppp:	Atom number
x:	Abbreviation label

Abbreviation is required for compatibility with previous versions of ISIS/Desktop, which allowed abbreviations with only one attachment. The attachment is denoted by two atom numbers, aaa and ppp. All of the atoms on the aaa side of the bond formed by aaa-ppp are abbreviated. The coordinates of the abbreviation are the coordinates of aaa. The text of the abbreviation is on the following line (x...). In current versions of ISIS, abbreviations can have any number of attachments and are written out using the Sgroup appendixes. However, any ISIS abbreviations that do have one attachment are also written out in the old style, again for compatibility with older ISIS versions, but this behavior might not be supported in future versions.

Charge [Generic]

```
M  CHGnn8 aaa vvv ...
```

vvv:	-15 to +15. Default of 0 = uncharged atom. When present, this property supersedes all charge and radical values in the atom block, forcing a 0 charge on all atoms not listed in an `M  CHG` or `M  RAD` line.

Radical [Generic]

```
M  RADnn8 aaa vvv ...
```

vvv:	Default of 0 = no radical, 1 = singlet (:), 2 = doublet (. or ^), 3 = triplet (^^). When present, this property supersedes all charge and radical values in the atom block, forcing a 0 (zero) charge and radical on all atoms not listed in an `M  CHG` or `M  RAD` line.

Isotope [Generic]

```
M  ISOnn8 aaa vvv ...
```

vvv: Absolute mass of the atom isotope as a positive integer. When present, this property supersedes all isotope values in the atom block. Default (no entry) means natural abundance. The difference between this absolute mass value and the natural abundance value specified in the PTABLE.DAT file must be within the range of -18 to +12.

Ring Bond Count [Query]

```
M  RBCnn8 aaa vvv ...
```

vvv: Number of ring bonds allowed: default of 0 = off, -1 = no ring bonds (r0),-2 = as drawn (r*); 2 = (r2), 3 = (r3), 4 or more = (r4).

Substitution Count [Query]

```
M  SUBnn8 aaa vvv ...
```

vvv: Number of substitutions allowed: default of 0 = off, -1 = no substitution (s0),-2 = as drawn (s*); 1, 2, 3, 4, 5 = (s1) through (s5), 6 or more = (s6).

Unsaturated Atom [Query]

```
M  UNSnn8 aaa vvv ...
```

vvv: At least one multiple bond: default of 0 = off, 1 = on.

Link Atom [Query]

```
M  LINnn4 aaa vvv bbb ccc
```

vvv,bbb,ccc: Link atom (aaa) and its substituents, other than bbb and ccc, can be repeated 1 to vvv times, (vvv > = 2). The bbb and ccc parameters can be 0 (zero) for link nodes on atoms with attachment point information.

Atom List [Query]

```
M  ALS aaannn e 1111222233334444555...
```

aaa: Atom number, value

nnn: Number of entries on line (16 maximum)

e: Exclusion, value is T if a 'NOT' list, F if a normal list.

1111: Atom symbol of list entry in field of width 4

Note: This line contains the atom symbol rather than the atom number used in the atom list block, and is independent of the Ptable. (For information about the Ptable, see *Symyx Chemical Representation*.). Any data found in this item supersedes data from the atom list block. The number of entries can exceed the fixed limit of *5* in the atom list block entry.

Attachment Point [Rgroup]

```
M  APOnn2 aaa vvv ...
```

vvv: Indicates whether atom aaa of the Rgroup member is the first attachment point (vvv = 1), second attachment point (vvv = 2), both attachment points (vvv = 3); default of 0 = no attachment.

Atom Attachment Order [Rgroup]

```
M  AAL aaann2 111 v1v 222 v2v ...
```

aaa:	Atom index of the Rgroup usage atom
nn2:	Number of pairs of entries that follow on the line
111:	Atom index of a neighbor of aaa
v1v	Attachment order for the aaa-111 bond
222	Atom index of a neighbor of aaa
v2v	Attachment order for the aaa-222 bond

Note: v1v and v2v are either 1 or 2 for the simple doubly attached Rgroup member.

This appendix provides explicit attachment list order information for R# atoms. The appendix contains atom neighbor index and atom neighbor value pairs. The atom neighbor value information identifies the atom neighbor index as the *nth* attachment. The implied ordering in V2000 molfiles is by atom index order for the neighbors of Rgroup usage atoms. If atom index order conflicts with the desired neighbor ordering at the R# atom, this appendix allows you to override to this default order.

If v1v=1 and v2v=2, Symyx ISIS/Host only writes this appendix if 111 is greater than 222. Note, however, that the attachment values can be written in any order.

Rgroup Label Location [Rgroup]

```
M  RGPnn8 aaa rrr ...
```

rrr: Rgroup number, value from 1 to 32 *, labels position of Rgroup on root.

* ISIS/Desktop limit

Rgroup Logic, Unsatisfied Sites, Range of Occurrence [Rgroup]

```
M  LOGnn1 rrr iii hhh ooo...
```

rrr: Rgroup number, value from 1 to 32 *

iii: Number of another Rgroup which must only be satisfied if rrr is satisfied (IF rrr THEN iii)

hhh: RestH property of Rgroup rrr; default is 0 = off, 1 = on. If this property is applied (on), sites labeled with Rgroup rrr can only be substituted with a member of the Rgroup or with H

ooo Range of Rgroup occurrence required: n = exactly n, n – m = n through m,> n = greater than n, < n = fewer than n, default (blank) is > 0. Any non-contradictory combination of the preceding values is also allowed; for example: 1, 3-7, 9, >11.

* ISIS/Desktop limit

Sgroup Type [Sgroup]

```
M  STYnn8 sss ttt ...
```

sss:	Sgroup number
ttt:	Sgroup type: SUP = abbreviation Sgroup (formerly called superatom), MUL = multiple group, SRU = SRU type, MON = monomer, MER = Mer type, COP = copolymer, CRO = crosslink, MOD = modification, GRA = graft, COM = component, MIX = mixture, FOR = formulation, DAT = data Sgroup, ANY = any polymer, GEN = generic.

Note: For a given Sgroup, an STY line giving its type must appear before any other line that supplies information about it. For a data Sgroup, an SDT line must describe the data field before the SCD and SED lines that contain the data (see Data Sgroup Data below). When a data Sgroup is linked to another Sgroup, the Sgroup must already have been defined.

Sgroups can be in any order on the Sgroup Type line. Brackets are drawn around Sgroups with the M SDI lines defining the coordinates.

Sgroup Subtype [Sgroup]

```
M  SSTnn8 sss ttt ...
```

ttt:	Polymer Sgroup subtypes: ALT = alternating, RAN = random, BLO = block

Ctab organization of an Sgroup structure

Sgroup Labels [Sgroup]

```
M  SLBnn8 sss vvv ...
```

vvv:	Unique Sgroup identifier

Sgroup Connectivity [Sgroup]

```
M  SCNnn8 sss ttt ...
```

ttt:	HH = head-to-head, HT = head-to-tail, EU = either unknown. Left justified.

Sgroup Expansion [Sgroup]

```
M  SDS EXPn15 sss ...
```

sss:	Sgroup index of expanded abbreviation Sgroups

Sgroup Atom List [Sgroup]

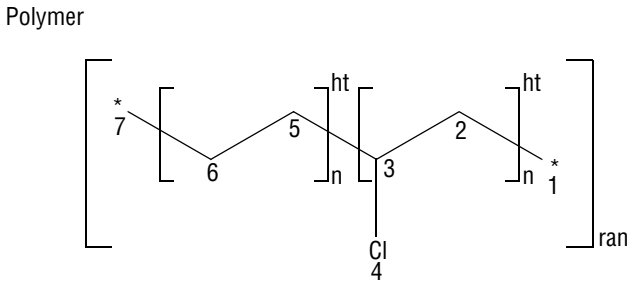
```
M  SAL sssn15 aaa ...
```

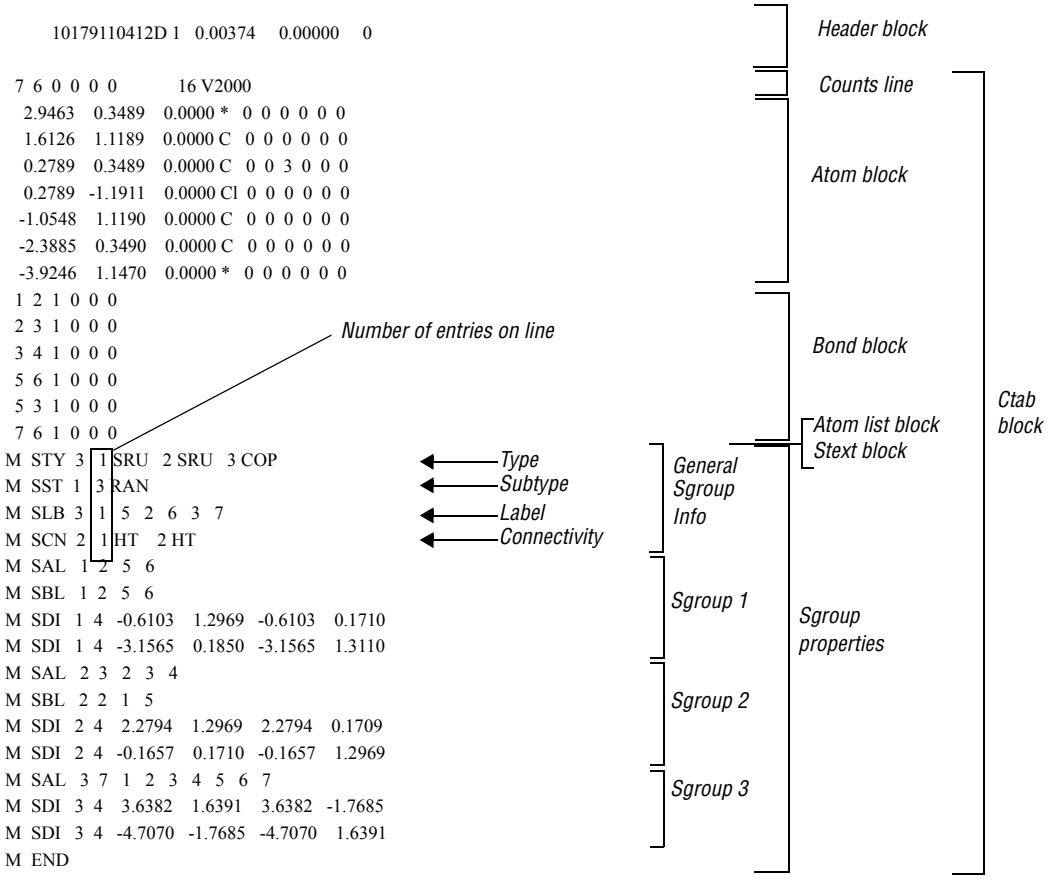
aaa:	Atoms in Sgroup sss

Sgroup Bond List [Sgroup]

```
M  SBL sssn15 bbb ...
```

bbb:	Bonds in Sgroup sss. (For data Sgroups, bbb's are the containment bonds, for all other Sgroup types, bbb's are crossing bonds.)





Multiple Group Parent Atom List [Sgroup]

`M SPA sssn15 aaa ...`

`aaa`: Atoms in paradigmatic repeating unit of multiple group `sss`

Note: To ensure that all current molfile readers consistently interpret chemical structures, multiple groups are written in their fully expanded state to the molfile. The `M SPA` atom list is a subset of the full atom list that is defined by the Sgroup Atom List `M SAL` entry.

Sgroup Subscript [Sgroup]

```
M  SMT sss m...
```

`m...`: Text of subscript Sgroup `sss`. (For multiple groups, `m...` is the text representation of the multiple group multiplier. For abbreviation Sgroups, `m...` is the text of the abbreviation Sgroup label.)

Sgroup Correspondence [Sgroup]

```
M  CRS sssnn6 bb1 bb2 bb3
```

`bb1, bb2`: Crossing bonds that share a common bracket

`bb3`: Crossing bond in repeating unit that connect to bond `bb1`

Sgroup Display Information [Sgroup]

```
M  SDI sssnn4 x1 y1 x2 y2
```

`x1,y1, x2,y2`: Coordinates of bracket endpoints

Abbreviation Sgroup Bond and Vector Information [Sgroup]

```
M  SBV sss bb1 x1 y1
```

`bb1`: Bond connecting to contracted abbreviation Sgroup

`x1,y1`: Vector for bond `bb1` connecting to contracted abbreviation Sgroup `sss`

Data Sgroup Field Description [Sgroup]

```
M  SDT sss fff...fffgghhh...hhhiijjj...
```

`sss`:	Index of data Sgroup
`fff...fff`:	30 character field name - no blanks, commas, or hyphens for MACCS-II
`gg`:	Field type - F = formatted, N = numberic, T = text (ignored)
`hhh...hhh`	20-character field units or format
`ii`:	Nonblank if data line is a query rather than Sgroup data, `MQ` = MACCS-II query, `IQ` = ISIS query, `PQ` = program name code query
`jjj...`:	Data query operator

Data Sgroup Display Information [Sgroup]

```
M  SDD sss xxxxx.xxxxyyyyy.yyyy eeefgh i jjjkkk ll m  noo
```

`sss`:	Index of data Sgroup
`x,y`:	Coordinates (2F10.4)
`eee`:	(Reserved for future use)
`f`:	Data display, `A` = attached, `D` = detached
`g`:	Absolute, relative placement, `A` = absolute, `R` = relative
`h`:	Display units, blank = no units displayed, `U` = display units
`i`:	(Reserved for future use)
`jjj`:	Number of characters to display (1...999 or `ALL`)

kkk:	Number of lines to display (unused, always 1)
ll:	(Reserved for future use)
m:	Tag character for tagged detached display (if non-blank)
n:	Data display DASP position (1...9). (MACCS-II only)
oo:	(Reserved for future use)

Data Sgroup Data [Sgroup]

```
M  SCD sss d...
M  SED sss d...
```

d...: Line of data for data Sgroup sss (69 chars per line, columns 12-80)

Note: A line of data is entered as text in 69-character substrings. Each SCD line adds 69 characters to a text buffer (starting with successive SCDs at character positions 1, 70, and 139). Following zero or more SCDs must be an SED, which can supply a final 69 characters. The SED initiates processing of the buffered line of text: trailing blanks are removed and right truncation to 200 characters is performed, numeric and formatted data are validated, and the line of data is added to data Sgroup sss. Left justification is not performed.

A data Sgroup can have more than one line of data, so more than one set of SCD and SED lines can be present for the same data Sgroup. The lines are added in the same order that they are encountered.

If 69 or fewer characters are to be entered on a line, they can be entered with a single SED not preceded by an SCD. On the other hand, a line can be entered to a maximum of 3 SCDs followed by a blank SED that terminates the line. The set of SCD and SED lines describing one line of data for a given data Sgroup must appear together, with no intervening lines for other data Sgroups' data.

Sgroup Hierarchy Information [Sgroup]

```
M  SPLnn8 ccc ppp ...
```

ccc: Sgroup index of the child Sgroup

ppp: Sgroup index of the parent Sgroup (ccc and ppp must already be defined via an STY line prior to encountering this line)

Sgroup Component Numbers [Sgroup]

```
M  SNCnn8 sss ooo ...
```

sss: Index of component Sgroup

ooo: Integer component order (1...256). This limit applies only to MACCS-II

3D Feature Properties [3D]

```
M  $3Dnnn
```

M $3D...: See below for information on the properties block of a 3D molfile. These lines must all be contiguous

Phantom Extra Atom

The format for phantom extra atom information is as follows:

```
M  PXA aaaxxxxx.xxxxyyyyy.yyyyzzzzz.zzzz H   e...
```

where:

`aaa:`	Index of (real) atom for attachment
`xyz:`	Coordinates for the added atom
`H:`	Atom symbol
`e...:`	Additional text string (for example, the abbreviation Sgroup label)

The bond to the added phantom atom is added as a crossing bond to the outermost Sgroup that contains atom aaa. Note this appendix supplies coordinates and up to 35 characters of 'label' that can be used for the ISIS/Desktop abbreviation Sgroup conversion mechanism. ISIS/Desktop uses this appendix to register hydrogen-only abbreviation Sgroups, which are often used as abbreviation Sgroup leaving groups on the desktop, but which cannot be directly registered into Symyx ISIS/Host databases. The hydrogen-only leaving groups are converted to PXA appendices for registration, and converted back when ISIS/Desktop reads the structure.

The following are limitations on phantom extra atom:

- Abbreviation Sgroup nesting cases
- No bonded phantom atom-phantom atom support

Abbreviation Sgroup Attachment Point

The format for abbreviation Sgroup attachment point is as follows:

```
M  SAP sssnn6 iii ooo cc
```

where:

`sss:`	Index of abbreviation Sgroup attachment point
`nn6:`	Number of `iii,ooo,cc` entries on the line (6 maximum)
`iii:`	Index of the attachment point atom
`ooo:`	Index of atom in `sss` that leaves when `iii` is substituted
`cc:`	2 character attachment identifier (for example, `H` or `T` for head/tail). No validation of any kind is performed, and ' ' is allowed. ISIS/Desktop uses the first character as the ID of the leaving group to attach if the bond between `ooo` and `iii` is deleted, and uses the second character to indicate the sequence polarity: `l` for left, `r` for right, and `x` for none (a crosslink).

The bond `iii-ooo` is either a sequence bond, a sequence crosslink bond, or a bond to a leaving group that terminates a sequence or caps a crosslink bond. This bond might have been deleted by the user to perform a substructure search. In this case, `ooo` will be 0. If the leaving group attached to iii consists of only a hydrogen, the leaving group will be replaced by a Phantom Extra Atom, as previously described. In this case, `iii` is set equal to `ooo` as a signal to ISIS/Desktop that a hydrogen-only leaving group must be reattached to iii.

An attachment point entry is one `iii,ooo,cc` triad.

Multiple M SAP lines are permitted for each abbreviation Sgroup to the maximum of the atom attachment limit. The order of the attachment entries is significant because the first `iii,ooo,c` becomes the first connection made when drawing to the collapsed abbreviation Sgroup, and so forth.

Abbreviation Sgroup Class

The format for abbreviation Sgroup class is as follows:

```
M  SCL sss d...
```

where:

`sss:`	Index of abbreviation Sgroup
`d...:`	Text string (for example, `PEPTIDE`, ...) 69 characters maximum

This appendix identifies the class of the abbreviation Sgroup. It distinguishes, for example, peptide groups from nucleotides.

Large REGNO

The format for the regno appendix is as follows:

```
M  REG r...
```

where:

`rrr:`	Free format integer regno

This appendix supports overflow of the I6 regno field in the molfile header. If this appendix is present, the value of the regno in the molfile header is superceded.

Sgroup Bracket Style

The format for the Sgroup bracket style is as follows:

```
M  SBTnn8 sss ttt ...
```

where:

`sss:`	Index of Sgroup
`ttt:`	Bracket display style: 0 = default, 1 = curved (parenthetic) brackets

This appendix supports altering the display style of the Sgroup brackets.

End of Block

```
M  END
```

This entry goes at the end of the properties block and is required for molfiles which contain a version stamp in the counts line.

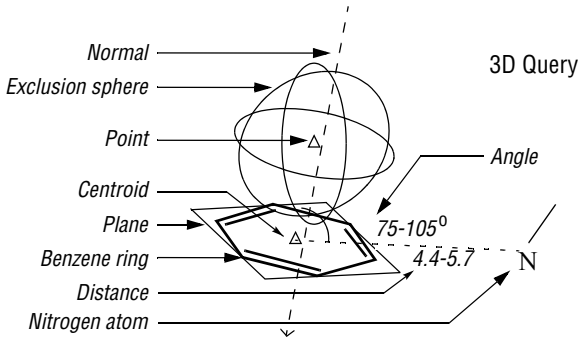
The Properties Block for 3D Features [3D]

For each 3D feature, the properties block includes:

- One 3D features count line
- One or more 3D features detail lines

The characters `M  $3D` appear at the beginning of each line describing a 3D feature. The information for 3D features starts in column 7.

Ctab organization of a 3D query




```
3D Query
      10179109553D 1   1.00000    0.00000     0

  8  7  0  0  0  0            18 V2000
    1.0252    0.2892    1.1122 C   0  0  0  0  0  0
   -0.4562    0.6578    1.3156 C   0  0  0  0  0  0
   -1.4813    0.3687    0.2033 C   0  0  0  0  0  0
   -1.0252   -0.2892   -1.1122 C   0  0  0  0  0  0
    0.4562   -0.6578   -1.3156 C   0  0  0  0  0  0
    1.4813   -0.3687   -0.2033 C   0  0  0  0  0  0
    4.1401   -0.1989    1.3456 N   0  0  0  0  0  0
    4.6453    0.5081    1.7417 C   0  0  0  0  0  0
  1  2  1  0  0  0
  2  3  2  0  0  0
  3  4  1  0  0  0
  4  5  2  0  0  0
  5  6  1  0  0  0
  6  1  2  0  0  0
  7  8  1  0  0  0
M  $3D  7
M  $3D -7  6
M  $3D  3
M  $3D  6  4  2
M  $3D -5 13
M  $3D  6    0.0000
M  $3D  1  2  3  4  5  6
M  $3D -8  7
M  $3D  9 10
M  $3D -3  6
M  $3D  9 11   -2.0000
M  $3D-16 12
M  $3D 12  1  0    1.5000
M  $3D-12 10
M  $3D 12  9  7   75.0000  105.0000
M  $3D -9  3
M  $3D  7  9    4.4000    5.7000
M  END
```

3D features count line

The first line in the properties block is the 3D features count line and has the following format:

```
M  $3Dnnn
```

where nnn is the number of 3D features on a model.

3D features detail lines

The lines following the 3D features count line describe each 3D feature on a model. Each 3D feature description consists of an identification line and one or more data lines:

The identification line is the first line and contains the 3D feature's type identifier, color, and name.

Each data line describes the construction of the 3D feature.

Identification line

The 3D feature identification line has the following format:

```
M  $3Dfffccc aaa...aaa ttt...ttt
```

where the variables represent:

fff:	3D feature type
ccc:	Color number (an internal number which is terminal dependent)
aaa...aaa:	3D feature name (up to 32 characters)
ttt...ttt:	Text comments (up to 32 characters) used by Symyx programs (see "3D data constraints [3D, Query]" on page 67)

3D feature type identifiers

Identifier	Meaning
-1	Point defined by two points and a distance (in Angstroms)
-2	Point defined by two points and a percentage
-3	Point defined by a point, a normal line, and a distance
-4	Line defined by two or more points (a best fit line if more than two points)
-5	Plane defined by three or more points (a best fit plane if more than three points)
-6	Plane defined by a point and a line
-7	Centroid defined by points
-8	Normal line defined by a point and a plane
-9	Distance defined by two points and a range (in Angstroms)
-10	Distance defined by a point, line, and a range (in Angstroms)
-11	Distance defined by a point, plane, and a range (in Angstroms)
-12	Angle defined by three points and a range (in degrees)
-13	Angle defined by two intersecting lines and a range (in degrees)
-14	Angle defined by two intersecting planes and a range (in degrees)
-15	Dihedral angle defined by 4 points and a range (in degrees)
-16	Exclusion sphere defined by a point and a distance (in Angstroms)
-17	Fixed atoms in the model
nnn	A positive integer indicates atom or atom-pair data constraints

Data line

The 3D feature defines the data line format. Each 3D object is treated as a pseudoatom and identified in the connection table by a number. The 3D object numbers are assigned sequentially, starting with the next number greater than the number of atoms. The data line formats for the 3D feature types are:

3D feature type identifiers

Type	Description of Data Line

-1	The data line for a point defined by two points and a distance (Å) has the following format:
	`M  $3Diiijjjddddd.dddd`
	where the variables represent:
	`iii`: ID number of a point
	`jjj`: ID number of a second point
	`ddddd.dddd` Distance from first point in direction of second point (Å), 0 if not used
	The following example shows `POINT_1` created from the atoms 1 and 3 with a constraint distance of 2Å.
	The first line is the identification line. The second line is the data line.
	`M  $3D -1  4 POINT_1`
	`M  $3D  1  3    2.0000`

-2	The data line for a point defined by two points and a percentage has the format: `M  $3Diiijjjddddd.ddddd` where the variables represent: `iii` ID number of a point `jjj` ID number of a second point `ddddd.dddd` Distance (fractional) relative to distance between first and second points, 0 if not used
-3	The data line for a point defined by a point, a normal line, and a distance (Å) has the format: `M  $3Diiilllddddd.dddd` where the variables represent: `iii` ID number of a point `lll` ID number of a normal line `ddddd.dddd` Distance (Å), 0 if not used **Note:** For chiral models, the distance value is signed to specify the same or opposite direction of the normal.

-4

The data lines for a best fit line defined by two or more points have the following format:

```
M  $3Dpppttttt.tttt
M  $3Diiijjj...zzz . . .
```

where the variables represent:

`ppp`	Number of points defining the line
`ttttt.tttt`	Deviation (Å), 0 if not used.
`iii`	Each `iii`, `jjj`, and `zzz` is the ID number `jjj` of an item in the model that defines the line
`jjj`	
`...`	
`zzz`	(to maximum of 20 items per data line)

The following line is defined by the four points 1, 14, 15, and 19 and has a deviation of 1.2Å. The first line is the identification line. The second and third lines are the data lines.

```
M  $3D -4  2 N_TO_AROM
M  $3D  4    1.2000
M  $3D  1 14 15 19
```

-5

The data lines for a plane defined by three or more points (a best fit plane if more than three points) have the following format:

```
M  $3Dpppttttt.tttt
M  $3Diiijjj...zzz
...
```

where the variables represent:

`ppp`	Number of points defining the line
`ttttt.tttt`	Deviation (Å), 0 if not used.
`iii`	Each `iii`, `jjj`, and `zzz` is the ID number jjj of an item in the model that defines the plane
`jjj`	
`...`	
`zzz`	(to maximum of 20 items per data line)

The following plane is defined by the three points 1, 5, and 14. The first line is the identification line. The second and third lines are the data lines.

```
M  $3D -5  4 PLANE_2
M  $3D  3    0.0000
M  $3D  1 5 14
```

-6

The data line for a plane defined by a point and a line has the following format:

```
M  $3Diiilll
```

where the variables represent:

`iii`	ID number of a point
`lll`	ID number of a line

The following plane is defined by the point 1 and the plane 16. The first line is the identification line. The second line is the data line.

```
M  $3D -6  3 PLANE_1
M  $3D  1 16
```

-7

The data lines of a centroid defined by points have the following format:

```
M  $3Dppp
M  $3Diiijjj...zzz ...
```

where the variables represent:

ppp	Number of points defining the centroid
iii	Each iii, jjj, and zzz is the ID number jjj of an item in the model that defines the centroid
jjj	
...	
zzz	(maximum of 20 items per data line).

The following centroid, ARO_CENTER, is defined by 3 items: 6, 8, and 10. The first line is the identification line. The second and third lines are the data lines.

```
M  $3D -7  1 ARO_CENTER
M  $3D  3
M  $3D  6  8 10
```

-8

The data line for a normal line defined by a point and a plane has the following format:

```
M  $3Diiijjj
```

where the variables represent:

iii	ID number of a point
jjj	ID number of a plane

The following normal line, ARO_NORMAL, is defined by the point 14 and the plane 15. The first line is the identification line. The second line is the data line.

```
M  $3D -8  1 ARO_NORMAL
M  $3D 14 15
```

-9

The data line for a distance defined by two points and a range (Å) has the following format:

```
M  $3Diiijjjddddd.ddddzzzzz.zzzz
```

where the variables represent:

iii	ID number of a point
jjj	ID number of a second point
ddddd.dddd	Minimum distance (Å)
zzzzz.zzzz	Maximum distance (Å)

The following distance, L, is between items 1 and 14 and has a minimum distance of 4.9Å and a maximum distance of 6.0Å. The first line is the identification line. The second line is the data line.

```
M  $3D -9  6 L
M  $3D  1 14    4.9000    6.0000
```

-10

The data line for a distance defined by a point, line, and a range (Å) has the format:

```
M  $3Diiilllddddd.ddddzzzzz.zzzz
```

where the variables represent:

iii	ID number of a point
lll	ID number of a line
ddddd.dddd	Minimum distance (Å)
zzzzz.zzzz	Maximum distance (Å)

-11

The data line for a distance defined by a point, plane, and a range (Å) has the format:

```
M  $3Diiijjjddddd.ddddzzzzz.zzzz
```

where the variables represent:

`iii`	ID number of a point
`jjj`	ID number of a plane
`ddddd.dddd`	Minimum distance (Å)
`zzzzz.zzzz`	Maximum distance (Å)

-12

The data line for an angle defined by three points and a range (in degrees) has the following format:

```
M  $3Diiijjjkkkddddd.ddddzzzzz.zzzz
```

where the variables represent:

`iii`	ID number of a point
`jjj`	ID number of a second point
`kkk`	ID number of a third point
`ddddd.dddd`	Minimum degrees
`zzzzz.zzzz`	Maximum degrees

The following angle, `THETA1`, is defined by the three points: 5, 17, and 16. The minimum angle is 80° and the maximum is 105°. The first line is the identification line. The second line is the data line.

```
M  $3D-12  5 THETA1
M  $3D  5 17 16   80.0000  105.0000
```

-13

The data line for an angle defined by two lines and a range (in degrees) has the following format:

```
M  $3Dlllmmmddddd.ddddzzzzz.zzzz
```

where the variables represent:

`lll`	ID number of a line, mmm ID number of a second line
`ddddd.dddd`	Minimum degrees
`zzzzz.zzzz`	Maximum degrees

`THETA2` is defined by the lines 27 and 26 with maximum and minimum angles of 45° and 80°. The first line is the identification line. The second line is the data line.

```
M  $3D-13  5 THETA2
M  $3D 27 26   45.0000   80.0000
```

-14

The data line for an angle defined by two planes and a range (in degrees) has the following format:

```
M  $3Diiijjjddddd.ddddzzzzz.zzzz
```

where the variables represent:

`iii`	ID number of a plane
`jjj`	ID numbers of a second plane
`ddddd.dddd`	Minimum degrees
`zzzzz.zzzz`	Maximum degrees

-15

The data line for a dihedral angle defined by four points and a range (in degrees) has the following format:

```
M  $3Diiijjjkkkllldddddd.ddddzzzzz.zzzz
```

where the variables represent:

`iii`	ID number of a point
`jjj`	ID number of a second point
`kkk`	ID number of a third point
`lll`	ID number of a fourth point
`ddddd.dddd`	Minimum degrees
`zzzzz.zzzz`	Maximum degrees

`DIHED1` is defined by the items 7, 6, 4, and 8 with minimum and maximum angles of 45° and 80°, respectively. The first line is the identification line. The second line is the data line.

```
M  $3D-15  5 DIHED1
M  $3D  7  6  4  8   45.0000   80.0000
```

-16

The data lines for an exclusion sphere defined by a point and a distance (Å) have the following format:

```
M  $3Diiiuuuaaaddddd.dddd
M  $3Dbbbccc...zzz ...
```

where the variables represent:

`iii`	ID number of the center of the sphere
`uuu`	1 or 0. 1 means unconnected atoms are ignored within the exclusion sphere during a search; 0 otherwise
`aaa`	Number of allowed atoms
`ddddd.dddd`	Radius of sphere (Å)
`bbb`	Each `bbb`, `ccc`, and `zzz` is an ID number of an allowed atom
`ccc`	
...	
`zzz`	(to maximum of 20 items per data line)

The following exclusion sphere is centered on point 24, has a radius of 5, and allows atom 9 within the sphere. The first line is the identification line. The second and third lines are the data lines.

```
M  $3D-16  7 EXCL_SPHERE
M  $3D 24  0  1    5.0000
M  $3D  9
```

-17

The data lines of the fixed atoms have the following format:

```
M  $3Dppp
M  $3Diiijjj...zzz ...
```

where the variables represent:

ppp Number of fixed points

`iii`	Each `iii`, `jjj`, and `zzz` is an ID number of a fixed atom
`jjj`	
`zzz`	(to maximum of 20 items per data line)

The following examples shows 4 fixed atoms. The first line is the identification line. The second and third lines are the data lines.

```
M  $3D-17
M  $3D  4
M  $3D  3  7 12 29
```

3D data constraints [3D, Query]

A positive integer is used as a type identifier to indicate an atom or atom-pair data constraint. Two lines are used to describe a data constraint. The lines have the following format:

```
M  $3Dnnncccaaa...aaabbbbbbbbbpppppppppsss...sss
M  $3Diiijjjddd...ddd
```

where the variables represent:

nnn:	Database-field number
ccc:	Color
aaa...aaa:	Database-field name (up to 30 characters)
bbbbbbbb:	/BOX = box-number (source of data) (up to 8 characters)
ppppppppp:	/DASP = n1, n2 where n1 and n2 are digits from 1-9 (data size and position) (up to 9 characters)
sss...sss:	/DISP = 3DN (name), 3DV (value), 3DQ (query), NOT (no text). First three in any combination to maximum total of 15 characters
iii:	ID number of an atom
jjj:	ID number of a second atom for atom-pair data, 0 if data is atom data
ddd...ddd:	Data constraint (based on format from database) (up to 64 characters)

Note: The ISIS data query requires a search operator, a blank space, then one or more operands. For more information on ISIS data query syntax, see the ISIS Help system entries on SBF (Search By Form) or QB (Query Builder) for entering text in a query.

The following example shows a numeric data constraint for the field CNDO.CHARGE on atom 12. The first line is the identification line. The second line is the data line.

```
M  $3D  7  0 CNDO.CHARGE
M  $3D 12  0   -0.3300   -0.1300
```

The following example shows a numeric data constraint for the field BOND.LENGTH on the atom pair 1 and 4. The first line is the identification line. The second line is the data line.

```
M  $3D  9  0 BOND.LENGTH
M  $3D  1  4    2.0500    1.8200
```

The following example shows a data constraint allowing any charge value for the field CHARGE on all the atoms. The first line is the identification line. The second line is the data line.

```
M  $3D 12  0 CHARGE
M  $3D999  0 @
```

Correspondence with V3000 appendices

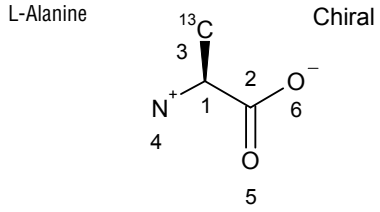
```
M  STY = type
M  SST = SUBTYPE
M  SLB = extindex
M  SCN = CONNECT
M  SDS = ESTATE
M  SAL = ATOMS
M  SBL = XBONDS or CBONDS
M  SPA = PATOMS
M  SMT = LABEL and MULT
M  CRS = XBHEAD,XBCORR
M  SDI = BRKXYZ
M  SBV = CSTATE
M  SDT = FIELDNAME, FIELDINFO, QUERYTYPE, QUERYOP
M  SDD = FIELDDISP
M  SCD = (not required)
M  SED = FIELDDATA
M  SPL = PARENT
M  SNC = COMPNO
M  SAP = SAP
M  SCL = CLASS
M  SBT = BRKTYP
```

V2000 Molfile

V2000 Molfile Overview

A molfile consists of a header block and a connection table. The following shows a V2000 molfile for alanine corresponding to the following structure:

Molfile organization illustrated using alanine




```
L-Alanine (13C)
          10169115362D 1   0.00366     0.00000     0
  6  5  0  0  1  0              3 V2000
   -0.6622    0.5342    0.0000 C   0  0  2  0  0  0
    0.6622   -0.3000    0.0000 C   0  0  0  0  0  0
   -0.7207    2.0817    0.0000 C   1  0  0  0  0  0
   -1.8622   -0.3695    0.0000 N   0  3  0  0  0  0
    0.6220   -1.8037    0.0000 O   0  0  0  0  0  0
    1.9464    0.4244    0.0000 O   0  5  0  0  0  0
  1  2  1  0  0  0
  1  3  1  1  0  0
  1  4  1  0  0  0
  2  5  2  0  0  0
  2  6  1  0  0  0
M  CHG  2   4   1   6  -1
M  ISO  1   3  13
M  END
```

Header Block

Counts Line

Atom Block

Bond Block

Properties Block

Connection Table (Ctab)

The format for a molfile is:

Header Block:	Identifies the molfile: molecule name, user's name, program, date, and other miscellaneous information and comments
Ctab Block:	Connection table

The detailed format for the header block follows.

Header Block for V2000 Molfile

Line 1:	Molecule name. This line is unformatted, but like all other lines in a molfile cannot extend beyond column 80. If no name is available, a blank line must be present. **Caution:** This line must not contain any of the reserved tags that identify any of the other CTAB file types such as `$MDL` (RGfile), `$$$$` (SDfile record separator), `$RXN` (rxnfile), or `$RDFILE` (RDfile headers).

Line 2:	This line has the format:
	`IIPPPPPPPPMMDDYYHHmmddSSssssssssssEEEEEEEEEEEERRRRRR`
	`(FORTRAN: A2<--A8--><---A10-->A2I2<--F10.5-><---F12.5--><-I6-> )`
	User's first and last initials (`I`), program name (`P`), date/time (`M/D/Y,H:m`), dimensional codes (`d`), scaling factors (`S, s`), energy (`E`) if modeling program input, internal registry number (`R`) if input through MDL form.
	A blank line can be substituted for line 2.
	If the internal registry number is more than 6 digits long, it is stored in an `M REG` line (see "Large REGNO" on page 58).

Line 3:	A line for comments. If no comment is entered, a blank line must be present.

V2000 Rxnfile

V2000 Rxnfile Overview

Rxnfiles contain structural data for the reactants and products of a reaction. See the example "V2000 Rxnfile for the acylation of benzene" on page 73. The format is:

[Rxnfile Header]
rrrppp
*r — $MOL
[Molfile of reactant]
*p — $MOL
[Molfile of product]

where:

`*r` is repeated for each reactant
`*p` is repeated for each product

Header Block

Line 1:	`$RXN` in the first position on this line identifies the file as a reaction file.
Line 2:	A line for the reaction name. If no name is available, a blank line must be present.

Line 3:	User's initials (I), program name and version (`P`), date/time (`M/D/Y,H:m`), and reaction registry number (`R`). This line has the format:
	`IIIIIIPPPPPPPPPMMDDYYYYHHmmRRRRRRR`
	`(FORTRAN: <-A6-><---A9--><---A12----><--I7-> )`
	A blank line can be substituted for line 3.
	If the internal registry number is more than 7 digits long, it is stored in an `"M  REG"` line (see "Large REGNO" on page 58).
	Note: In rxnfiles produced by earlier versions of Symyx ISIS/Host, the year occupied two digits instead of four. There are corresponding minor changes in the adjacent fields. The format of the line is:
	`IIIIIIPPPPPPPPPMMDDYYHHmmRRRRRRRR`
	`(FORTRAN: <-A6-><---A10--><---A10--><--I8--> )`

Line 4	A line for comments. If no comment is entered, a blank line must be present.

Reactants/Products

A line identifying the number of reactants and products, in that order. The format is:

`rrrppp`

where the variables represent:

`rrr` Number of reactants*

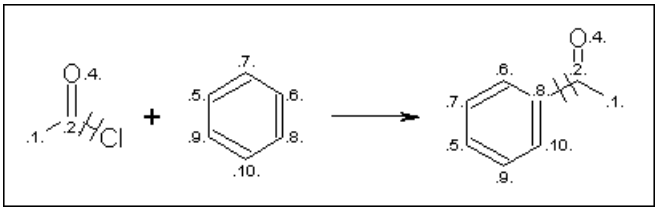
`ppp` Number of products*

* Symyx ISIS/Host has a limit of 8 reactants and 8 products. Symyx Direct does not impose any limits.

Molfile Blocks

A series of blocks, each starting with `$MOL` as a delimiter, giving the molfile for each reactant and product in turn. The molfile blocks are always in the same order as the molecules in the reaction; reactants first and products second.

The rxnfile in "V2000 Rxnfile for the acylation of benzene" on page 73 corresponds to the following reaction (which happens to have atom-atom mapping):



V2000 Rxnfile for the acylation of benzene

```
$RXN

          1017911041    7439

 2  1
$MOL

 REACCS8110179110412D 1   0.00380     0.00000   315

 4  3  0  0  0  0            1 V2000
   0.3929   -0.2577    0.0000 C   0  0  0  0  0  0  0  0  0  0  1  0  0
  -1.0590   -0.7710    0.0000 C   0  0  0  0  0  0  0  0  0  0  2  0  0
   0.3929    1.2823    0.0000 O   0  0  0  0  0  0  0  0  0  0  3  0  0
   1.6503   -1.1468    0.0000 Cl  0  0  0  0  0  0  0  0  0  0  0  0  0
 1  2  1  0  0  0  2
 1  3  2  0  0  0  2
 1  4  1  0  0  0  4
M  END
$MOL

       10179110412D 1   0.00371     0.00000     8

 6  6  0  0  0  0            1 V2000
   1.3335   -0.7689    0.0000 C   0  0  0  0  0  0  0  0  0  0  5  0  0
   1.3335    0.7689    0.0000 C   0  0  0  0  0  0  0  0  0  0  6  0  0
   0.0000   -1.5415    0.0000 C   0  0  0  0  0  0  0  0  0  0  7  0  0
   0.0000    1.5415    0.0000 C   0  0  0  0  0  0  0  0  0  0  8  0  0
  -1.3335   -0.7689    0.0000 C   0  0  0  0  0  0  0  0  0  0  9  0  0
  -1.3335    0.7689    0.0000 C   0  0  0  0  0  0  0  0  0  0 10  0  0
 1  2  1  0  0  0  2
 1  3  2  0  0  0  2
 2  4  2  0  0  0  2
 3  5  1  0  0  0  2
 4  6  1  0  0  0  2
 5  6  2  0  0  0  2
M  END
$MOL

       10179110412D 1   0.00374     0.00000   255

 9  9  0  0  0  0            1 V2000
  -0.5311   -0.1384    0.0000 C   0  0  0  0  0  0  0  0  0  0  5  0  0
  -1.8626    0.6321    0.0000 C   0  0  0  0  0  0  0  0  0  0  6  0  0
  -0.5311   -1.6943    0.0000 C   0  0  0  0  0  0  0  0  0  0  7  0  0
   0.8191    0.6284    0.0000 C   0  0  0  0  0  0  0  0  0  0  1  0  0
  -3.2278   -0.1346    0.0000 C   0  0  0  0  0  0  0  0  0  0  8  0  0
  -1.8813   -2.4723    0.0000 C   0  0  0  0  0  0  0  0  0  0  9  0  0
   2.1282   -0.1085    0.0000 C   0  0  0  0  0  0  0  0  0  0  2  0  0
   0.8191    2.2292    0.0000 O   0  0  0  0  0  0  0  0  0  0  3  0  0
  -3.2278   -1.6831    0.0000 C   0  0  0  0  0  0  0  0  0  0 10  0  0
 1  2  1  0  0  0  2
 1  3  2  0  0  0  2
 1  4  1  0  0  0  4
 2  5  2  0  0  0  2
 3  6  1  0  0  0  2
 4  7  1  0  0  0  2
 4  8  2  0  0  0  2
 5  9  1  0  0  0  2
 6  9  2  0  0  0  2
M  END
```

Header block

#Reactants and #Products

Molfile delimiter

Molfile for first reactant

Molfile delimiter

Molfile for second reactant

Molfile delimiter

Molfile for product

V2000 RGfiles (Rgroup file)

RGfile Overview

The format of an RGfile (Rgroup file) is shown below. Lines beginning with $ define the overall structure of the Rgroup query. The molfile header block is embedded in the Rgroup header block.

In addition to the primary connection table (Ctab block) for the root structure, a Ctab block defines each member (*m) within each Rgroup (*r).

```
$MDL REV 1 date/time
$MOL
$HDR
[Molfile Header Block = name, pgm info, comment]
$END HDR
$CTAB
[Ctab Block = count + atoms + bonds + lists + props]
$END CTAB
$RGP
rrr   [where rrr = Rgroup number]
$CTAB
[Ctab Block]
$END CTAB
$END RGP
$END MOL
```

*r

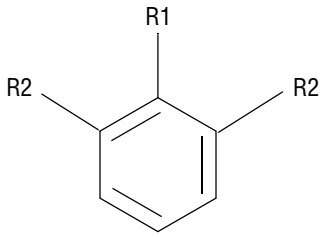
*m

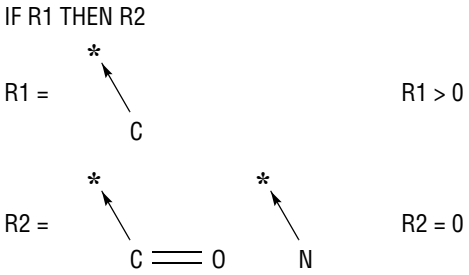
where:

*r (Rgroup) is repeated. ISIS/Desktop has an internal limit of 32 Rgroups.

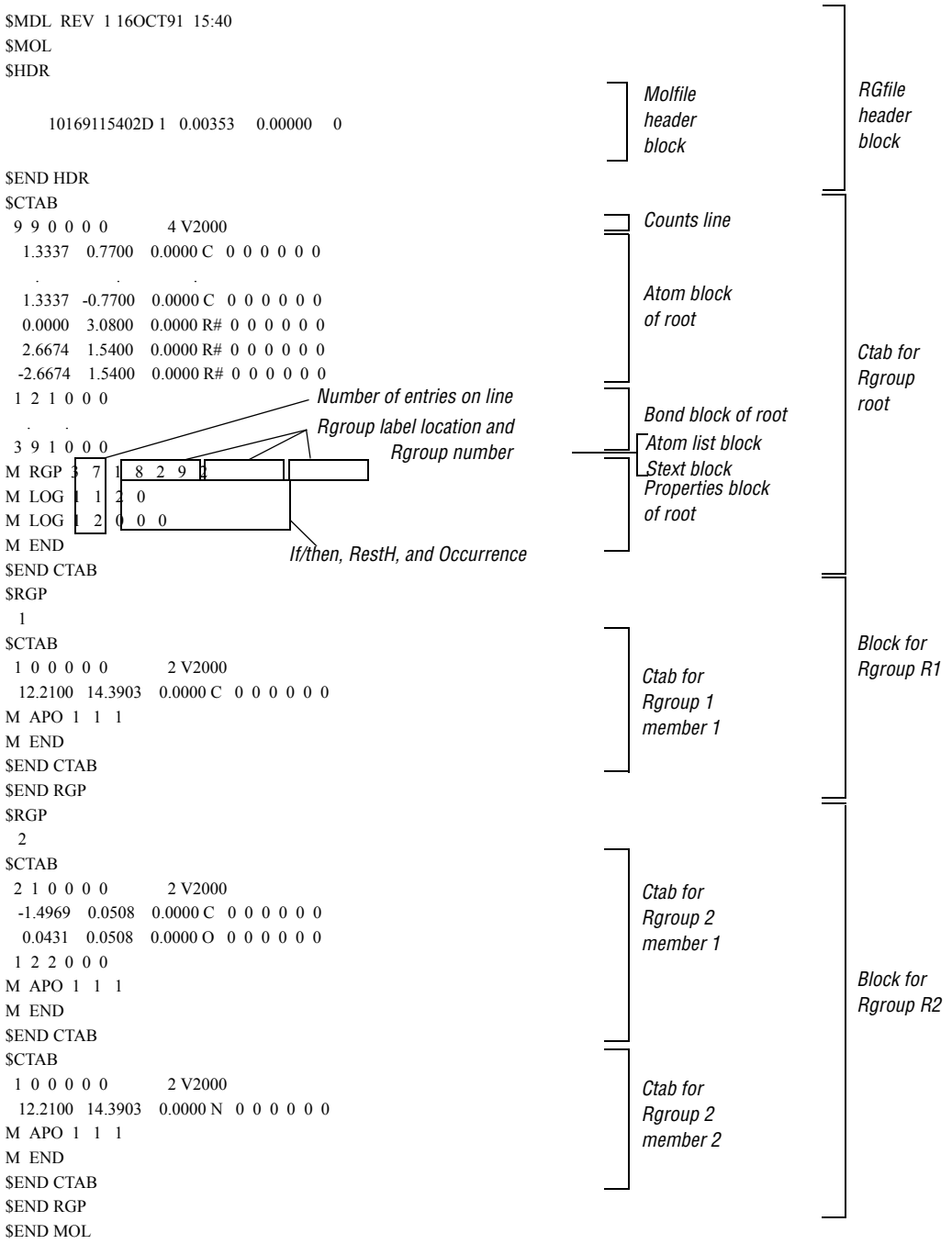
*m (member) is repeated.

The "Example of an RGfile (Rgroup query file)" on page 76 corresponds to the following Rgroup query:





Example of an RGfile (Rgroup query file)

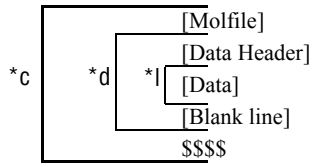


SDfiles (multiple structures and optional data)

SDfile Overview

An SDfile (structure-data file) contains the structural information and associated data items for one or more compounds, which can be V3000, V2000, or a combination of both.

The format is:



where:

`*l` is repeated for each line of data
`*d` is repeated for each data item
`*c` is repeated for each compound

A *[Molfile]* block has the molfile format.

A *[Data Header]* (one line) precedes each item of data, starts with a *greater than* (>) sign, and contains at least one of the following:

- The field name enclosed in angle brackets. For example: `<melting.point>`
- The field number, `DTn` , where *n* represents the number assigned to the field in a MACCS-II database

Note: The > sign is a reserved character. A field name cannot contain hyphen (-), period (.), less than (<), greater than (>), equal sign (=), percent sign (%) or blank space ( ). Field names must begin with an alpha character and can contain alpha and numeric characters after that, including underscore.

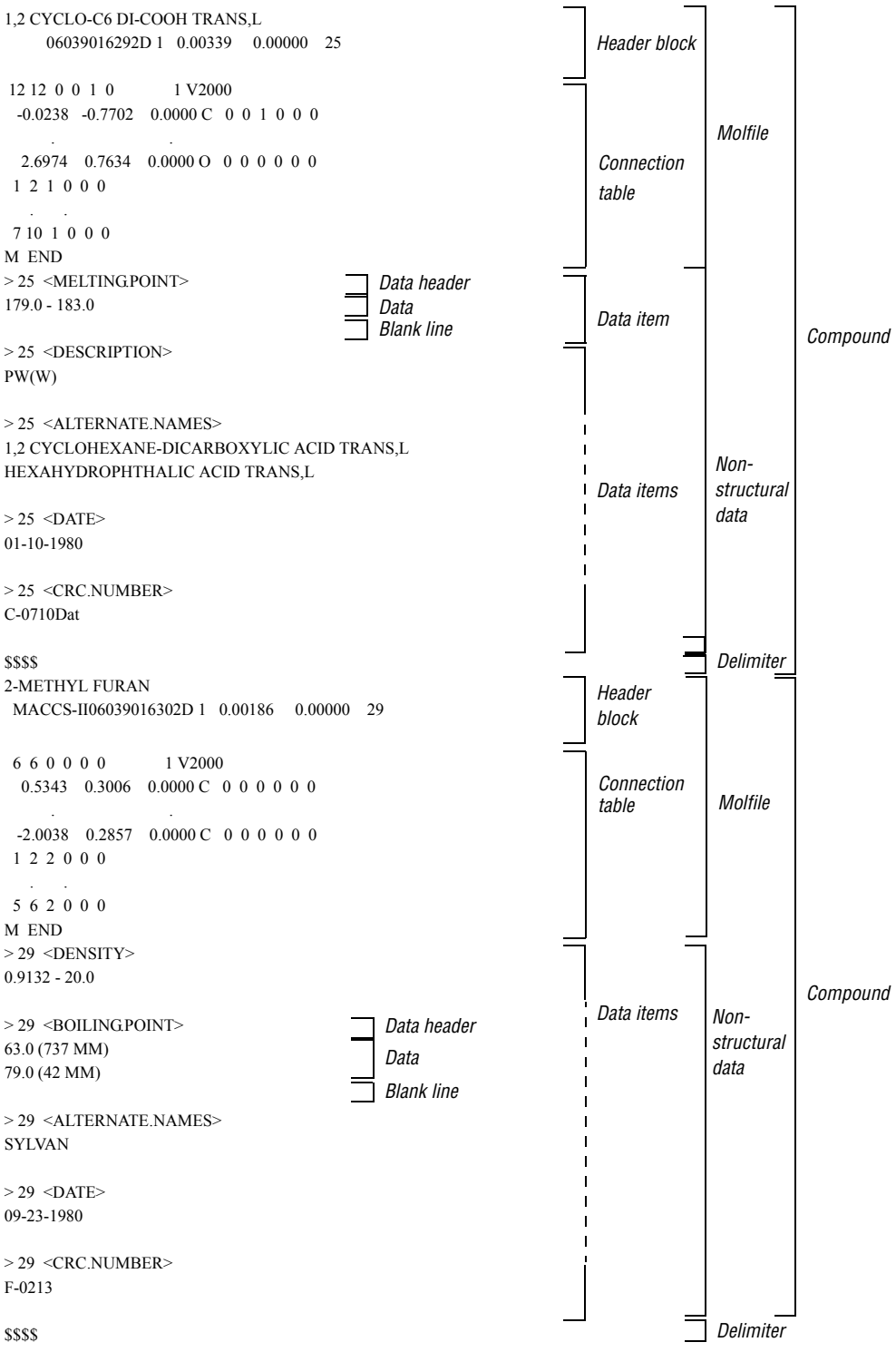
Optional information for the data header includes:

- The compound's external and internal registry numbers. External registry numbers must be enclosed in parentheses.
- Any combination of information

The following are examples of valid data headers:

```
> <MELTING.POINT>
> 55     (MD-08974)     <BOILING.POINT>   DT12
> DT12   55
> (MD-0894)   <BOILING.POINT>   FROM ARCHIVES
```

Example of an SDfile



A *[Data]* value can extend over multiple lines containing up to 200 characters each. A blank line terminates each data item.

A line beginning with four dollar signs ($$$$) terminates each complete data block describing a compound.

A datfile (data file) is effectively an SDfile with no *[Molfile]* descriptions or $$$$ delimiters. The *[Data Header]* in a datfile must include either an external or internal registry number in addition to a field name or number.

Notes about using blank lines

- *Only* one blank line should terminate a data item.
- There should *only* be one blank line between the last data item and the $$$$ delimiter line.
- If the SDfile only contains structures, there can be *no* blank line between the last "M END" and the $$$$ delimiter line.

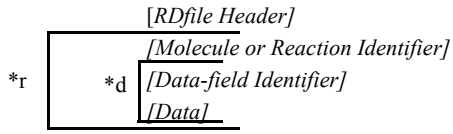
RDfiles (multiple reactions and optional data)

RDfile Overview

An RDfile (reaction-data file) consists of a set of editable "records.", which can be which can be V3000, V2000, or a combination of both.

Each record defines a molecule or reaction, and its associated data. See "Example of a reaction RDfile" on page 84).

The format for an RDfile is:



where:

- `*d` is repeated for each data item
- `*r` is repeated for each reaction or molecule

Each logical line in an RDfile starts with a keyword in column 1 of a physical line. One or more blanks separate the first argument (if any) from the keyword. The blanks are ignored when the line is read. After the first argument, blanks are significant.

An argument longer than 80 characters breaks at column 80 and continues in column 1 of the next line. (The argument can continue on additional lines up to the physical limits on text length imposed by the database.)

Note: For data lines, a line break between portions of text items is interpreted as a newline character in the text unless column 81 contains a + mark.

The RDfile must not contain any blank lines except as part of embedded molfiles, rxnfiles, or data. An identifier separates records.

RDfile Header

Line 1:	`$RDFILE 1`: The *[RDfile Header]* must occur at the beginning of the physical file and identifies the file as an RDfile. The version stamp "1" is intended for future expansion of the format.
Line 2:	`$DATM`: Date/time (`M/D/Y, C`) stamp. This line is treated as a comment and ignored when the program is read.

Molecule and Reaction Identifiers

A *[Molecule or Reaction Identifier]* defines the start of each complete record in an RDfile. The form of a *molecule* identifier must be one of the following:

```
$MFMT [$MIREG internal-regno embedded molfile
$MFMT [$MEREG external-regno]] embedded molfile
$MIREG internal-regno
$MEREG external-regno
```

where:

- `$MFMT` defines a molecule by specifying its connection table as a molfile
- `$MIREG` *internal-regno* is the internal registry number (sequence number in the database) of the molecule
- `$MEREG` *external-regno* is the external registry number of the molecule (any uniquely identifying character string known to the database, for example, CAS number)
- Square brackets (`[]`) enclose optional parameters
- An embedded `molfile` follows immediately after the `$MFMT` line

The forms of a *reaction* identifier closely parallel that of a molecule:

```
$RFMT [$RIREG internal-regno] embedded rxnfile
$RFMT [$REREG external-regno] embedded rxnfile
$RIREG internal-regno
$REREG external-regno
```

where:

- `$RFMT` defines a reaction by specifying its description as a rxnfile
- `$RIREG` *internal-regno* is the internal registry number (sequence number in the database) of the reaction
- `$REREG` *external-regno* is the external registry number of the reaction (any uniquely identifying character string known to the database)
- Square brackets (`[]`) enclose optional parameters
- An embedded `rxnfile` follows immediately after the `$RFMT` line

Note: Host and Isentris allow a reaction or molecule to be identified by internal or external regno but not both types of regno.

Data-field Identifier

The *[Data-field Identifier]* specifies the name of a data field in the database. The format is:

```
$DTYPE field name
```

Data

Data associated with a field follows the field name on the next line and has the form:

```
$DATUM datum
```

The format of *datum* depends upon the data type of the field as defined in the database. For example: integer, real number, real range, text, molecule regno.

For fields whose data type is "molecule regno," the *datum* must specify a molecule and, with the exception noted below, use one of the formats defined above for a molecular identifier. For example:

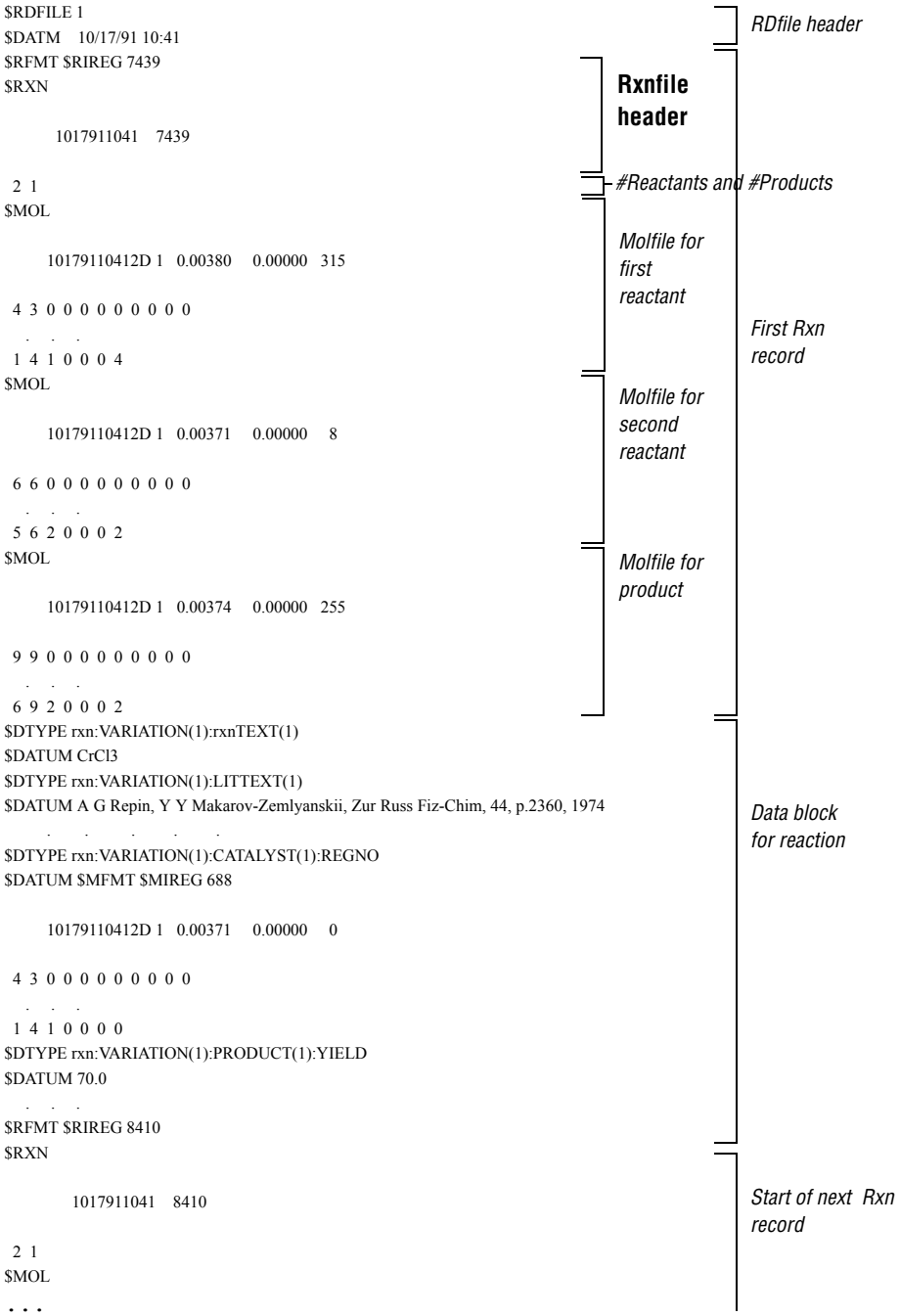
```
$DATUM $MFMT embedded molfile
$DATUM $MEREG external-regno
$DATUM $MIREG internal-regno
```

In addition, the following special format is accepted:

```
$DATUM molecule-identifier
```

Here, *molecule-identifier* acts in the same way as *external-regno* in that it can be any text string known to the database that uniquely identifies a molecule. (It is usually associated with a data field different from the *external-regno*.)

Example of a reaction RDfile



Appendix A: Stereo Notes

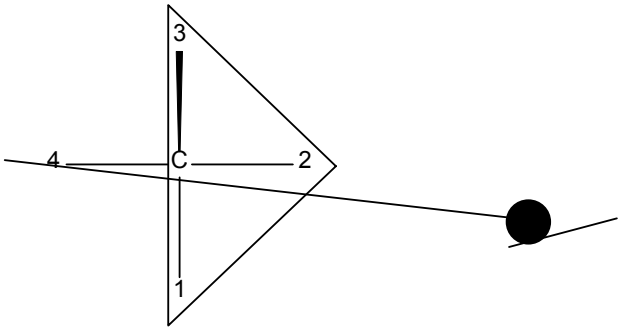
Parity values can appear in the atom blocks of CTFiles. See `CFG` (for stereo configuration) and and `sss` (atom stereo parity) in the table for "Atom Block" on page 13.

Parity is illustrated as follows:

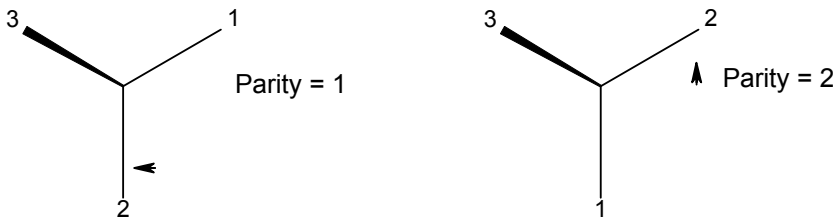
Mark a bond attached at a stereo center Up or Down to define the configuration

Number the atoms surrounding the stereo center with 1, 2, 3, and 4 in order of increasing atom number (position in the atom block) (a hydrogen atom should be considered the highest numbered atom, in this case atom 4). View the center from a position such that the bond connecting the highest-numbered atom (4) projects behind the plane formed by atoms 1, 2, and 3.

Note: In the figure, atoms 1, 2, and 4 are all in the plane of the paper, and atom 3 is above the plane.



Sighting towards atom number 4 through the plane (123), you see that the three remaining atoms can be arranged in either a clockwise or counterclockwise direction in ascending numerical order.



The Ctab lists a parity value of 1 for a clockwise arrangement at the stereo center and 2 for counterclockwise. A center with an Either bond has a parity value of 3. An unmarked stereo center is also assigned a value of 3. The first example above has a parity value of 2.

Note: For additional information about Symyx stereochemistry, see the document entitled *Symyx Chemical Representation*, which has an appendix entitled *Representation of Stereochemistry in Symyx Databases*.

Index

Numbers

A

B

C

D

E

F

H

I

L

M

N

O

P

Q

R

S

T

U

V

X