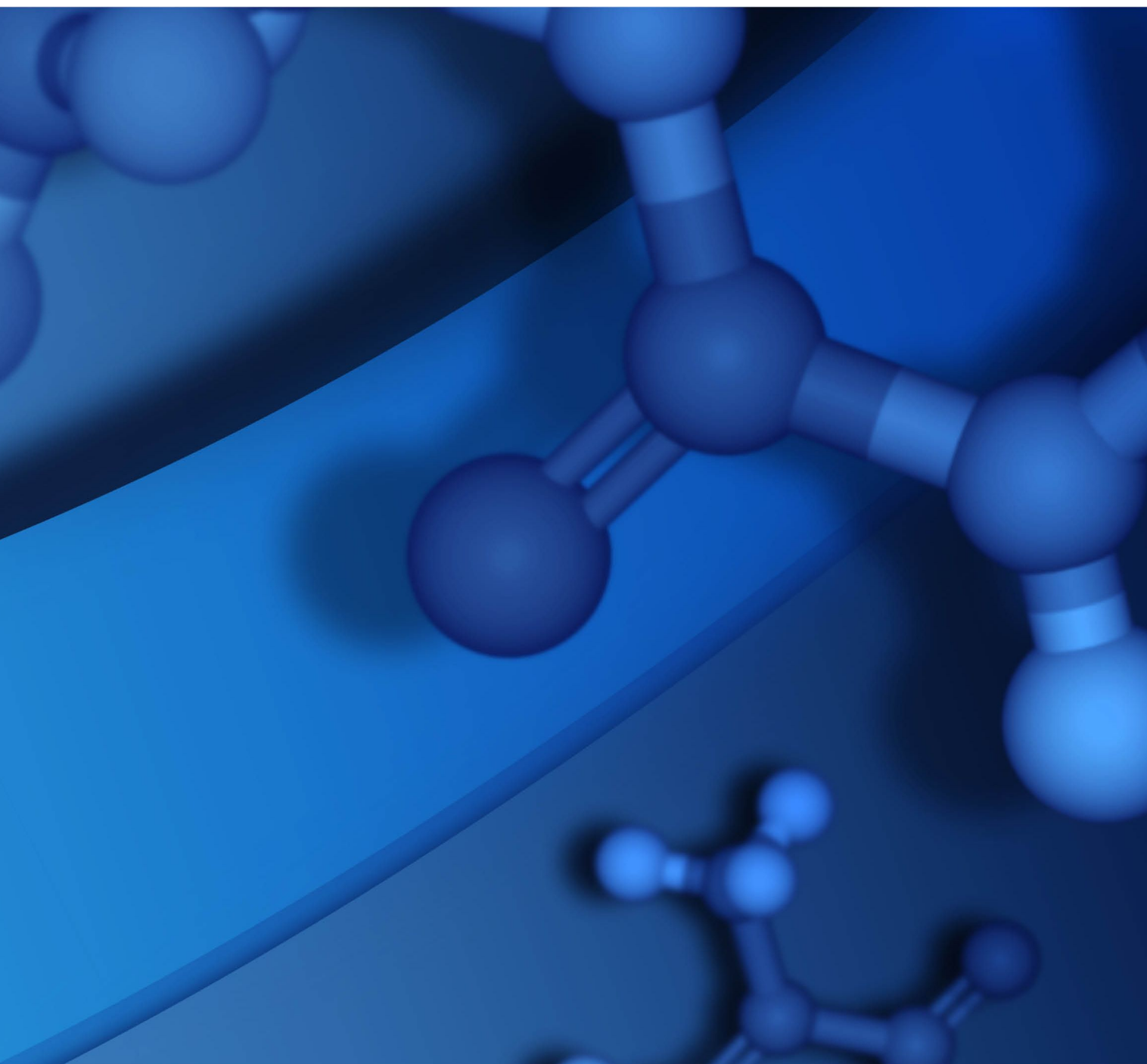


CTFILE FORMATS

BIOVIA DATABASES 2020



Copyright Notice

©2019 Dassault Systèmes. All rights reserved. 3DEXPERIENCE, the Compass icon and the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3DVIA, 3DSWYM, BIOVIA, NETVIBES, IFWE and 3DEXCITE, are commercial trademarks or registered trademarks of Dassault Systèmes, a French "société européenne" (Versailles Commercial Register # B 322 306 440), or its subsidiaries in the U.S. and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

Acknowledgments and References

To print photographs or files of computational results (figures and/or data) obtained by using Dassault Systèmes software, acknowledge the source in an appropriate format. For example:

"Computational results were obtained by using Dassault Systèmes BIOVIA software programs. CTFFile Formats was used to perform the calculations and to generate the graphical results."

Dassault Systèmes may grant permission to republish or reprint its copyrighted materials. Requests should be submitted to Dassault Systèmes Customer Support, either by visiting <https://www.3ds.com/support/> and clicking **Call us** or **Submit a request**, or by writing to:

Dassault Systèmes Customer Support
10, Rue Marcel Dassault
78140 Vélizy-Villacoublay
FRANCE

Contents

Chapter 1: About BIOVIA CTfile formats	1	Example V3000 Rxnfile for the Acylation of Benzene	39
Overview	1		
Advantages of V3000 Format	1		
Automatic V3000 Output	1		
CTfiles - Summary of Each Format	2		
Description of the Formats	3		
XDfile Types	3		
Properties Supported by CTfile Types	4		
Conventions	4		
Chapter 2: Connection Table (CTAB)	5	Chapter 6: V2000 Connection Table (CTAB)	41
CTAB Block Format	5	CTAB Block Format for V2000	41
General Syntax of Entries	6	Example: Alanine in V2000 format	41
The Connection Table	6	The Counts Line	43
CTAB Block	7	The Atom Block	43
Counts Line	7	The Bond Block	45
Atom Block	7	The Atom List Block[Query]	46
Bond Block	11	The Properties Block	47
Link Atom Line	12	Atom Alias [not used in current products]	47
Sgroup Block	13	Atom Value	48
Collection Block	20	Group Abbreviation [not used in current products]	48
3D Block	22	Charge [Generic]	49
The Rgroup	25	Radical [Generic]	49
Rgroup Block	25	Isotope [Generic]	49
Rgroup Logic Lines	28	Ring Bond Count [Query]	49
Template Block	28	Substitution Count [Query]	49
Chapter 3: Molfile	32	Unsaturated Count [Query]	49
Example of molfile - Alanine	32	Link Atom [Query]	50
V3000 Header	33	Atom List [Query]	50
Chapter 4: RGfiles (Rgroup file)	35	Attachment Point [Rgroup]	50
Example of an RGfile (Rgroup file)	35	Atom Attachment Order [Rgroup]	50
Chapter 5: Rxnfile	38	Rgroup Label Location [Rgroup]	51
Header Block	38	Rgroup Logic, Unsatisfied Sites, Range of Occurrence [Rgroup]	51
Counts line	38	Sgroup Type [Sgroup]	51
		Sgroup Subtype [Sgroup]	52
		Sgroup Labels [Sgroup]	52
		Sgroup Connectivity [Sgroup]	52
		Sgroup Expansion [Sgroup]	52
		Sgroup Atom List [Sgroup]	52

Sgroup Bond List[Sgroup]	52	V2000 Rxnfile for the Acylation of Benzene .	72
Multiple Group Parent Atom List [Sgroup]	53	Chapter 9: V2000 RGfiles (Rgroup file) ...	73
Sgroup Subscript [Sgroup]	54	RGfile Overview	73
Sgroup Correspondence [Sgroup]	54	Example of an RGfile	74
Sgroup Display Information [Sgroup]	54	Chapter 10: SDfiles (multiple structures and optional data)	75
Abbreviation Sgroup Bond and Vector Information [Sgroup]	54	Example of an SDfile	76
Data Sgroup Field Description [Sgroup] ..	54	Chapter 11: RDfiles	78
Data Sgroup Display Information [Sgroup]	54	RDfile Overview	78
Data Sgroup Data [Sgroup]	55	RDfile Header	78
Sgroup Hierarchy Information [Sgroup] ..	55	Molecule and Reaction Identifiers	78
Sgroup Component Numbers [Sgroup] ..	55	Data-field Identifier	79
3D Feature Properties [3D]	56	Data	79
Phantom Extra Atom	56	Example of a reaction RDfile	80
Abbreviation Sgroup Attachment Point ..	56	Chapter 12: XDfiles	81
Abbreviation Sgroup Class	57	Overview	81
Large REGNO	57	Data Formatting	81
Sgroup Bracket Style	57	XML Processing Instructions	82
End of Block	57	Hierarchy of Elements	83
The Properties Block for 3D Features [3D] ..	58	Alphabetical List of Elements	83
3D Features Count Line	59	Copyright	84
3D Features Detail Lines	59	CreateDate	84
Identification Line	59	CreateTime	85
3D Feature Type Identifiers	60	CreatorName	85
Data Line	60	Data	86
3D Data Constraints [3D, Query]	66	Dataset	87
Chapter 7: V2000 Molfile	68	DataSource	88
V2000 Molfile Overview	68	Description	88
Header Block for V2000 Molfile	69	Field	88
Chapter 8: V2000Rxnfile	70	FieldDef	89
V2000 Rxnfile Overview	70	Parent	92
Header Block	70	ParentDef	92
Reactants/Products	70	ProgramSource	94
Molfile Blocks	71	Metadata	95
		Record	95

Source	96
XDfile	97
Appendix A: Stereo Notes	99

Chapter 1:

About BIOVIA CTfile formats

BIOVIA 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 BIOVIA product supports, see the documentation for that product.

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.

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 BIOVIA 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 41.

Advantages of V3000 Format

The V3000 format is intended to be the primary means for communication of future enhancements to BIOVIA 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 28.
- 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

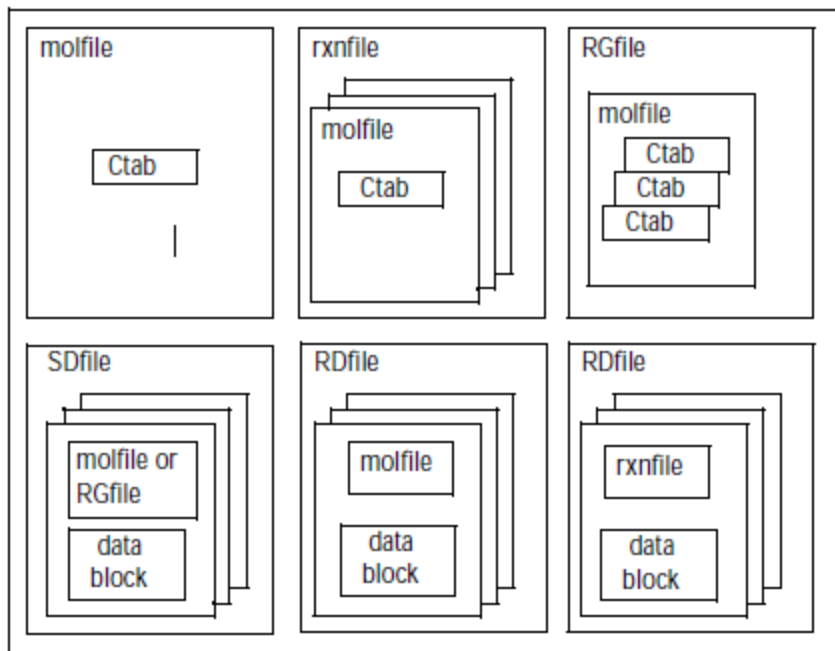
Current BIOVIA products support reading and writing of both V2000 and V3000 formats. These products continue to default to writing V2000 molfiles to maximize interoperability with third party applications. Future product versions might default to output of the preferred V3000 format.

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 [Collection Block](#) on page 20.
- Enhanced stereochemistry features - The V3000 format is required when using the enhanced stereochemical representations. See the [Collection Block](#) on page 20. For a complete discussion of BIOVIA enhanced chemical representation, see BIOVIA 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 [Counts Line](#) on page 7). For more details about the fixed field widths in the V2000 format, see [V2000 Connection Table \(CTAB\)](#) on page 41.
- Template block - useful for representing large structures, such as biological molecules. See [Template Block](#) on page 28.
- New properties supported only in V3000 format - such as atom CLASS and SEQID. See [Meaning of values in the atom block](#) on page 8.

CTfiles - Summary of Each Format

The following diagram illustrates the relationship between the various file formats. See [Description of the Formats](#) on page 3.

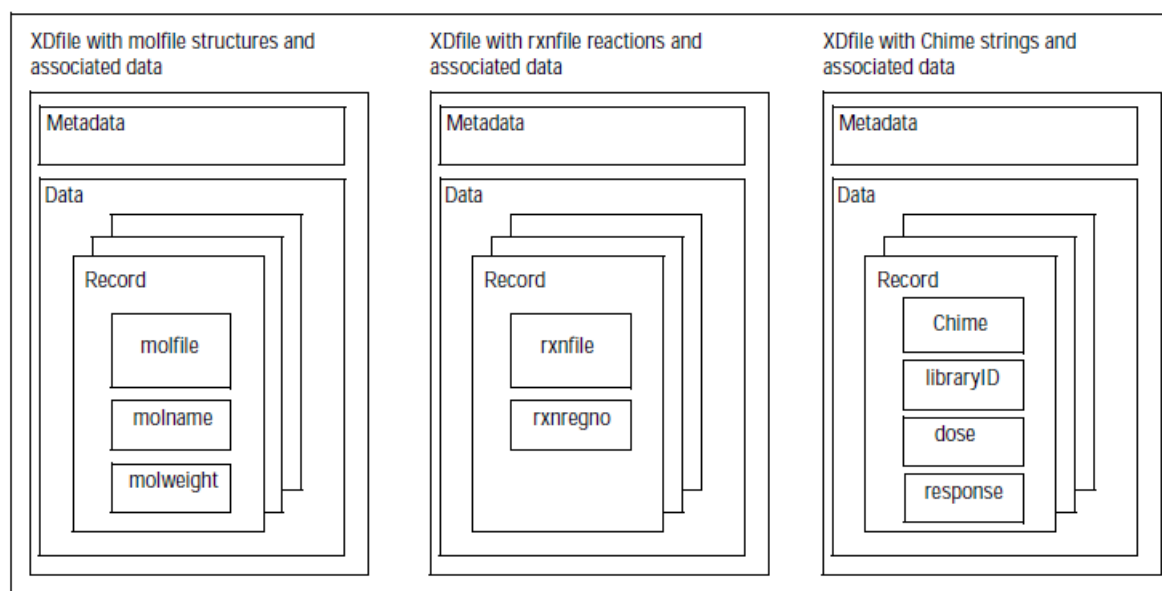


Description of the Formats

molfiles	Molecule files: Each molfile describes a single molecular structure which can contain disjoint fragments. See Molfile on page 32.
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 35.
rxnfiles	Reaction files: Each rxnfile contains the structural information for the reactants and products of a single reaction. See Rxnfile on page 38.
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 BIOVIA databases.
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.
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 CTFfile 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 Isentris Developer Documentation.

XDfile Types

The following diagram illustrates three examples of XDfiles:



Properties Supported by CTFfile Types

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

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.

Chapter 2:

Connection Table (CTAB)

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 BIOVIA file formats.

This chapter describes the preferred V3000 format, which supports more features than the legacy V2000 format. For information on the legacy format, see [V2000 Connection Table \(CTAB\)](#) on page 41.

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. See Counts Line on page 7.
Atom block:	Specifies the atomic symbol and any mass difference, charge, stereochemistry, and associated hydrogens for each atom. See Atom Block on page 7.
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. See Bond Block on page 11.
Link atom line:	Supports the representation of certain kinds of Sgroups with repeating units, such as link nodes. For details about Sgroups, see <i>BIOVIA Chemical Representation</i> . See Link Atom Line on page 12.
Sgroup block:	Sgroup block. For details about Groups, see <i>BIOVIA Chemical Representation</i> . See Sgroup Block on page 13.
Collection block:	A collection block specifies all collection information for objects in the current connection table context. See Collection Block on page 20.
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 22.
Rgroup block:	Supports Markush representation. See Rgroup Logic Lines on page 28. For details about Markush structures (Rgroups), see <i>BIOVIA Chemical Representation</i> .

	See Rgroup Block on page 25.
Template block	Supports the representation of large molecules, such as biologics. See Template Block on page 28.

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 val, 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 "abcdef"
```

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]*
[collection-block]
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] - MV30 [VAL=val] -
M V30 [HCOUNT=val] [STBOX=val] [INVRET=val] [EXACHG=val] - MV30
[SUBST=val] [UNSAT=val] [RBCNT=val] -
M V30 [ATTCHPT=val] -
M V30 [Rgroups=(nvals val [val ...])] -
M V30 [ATTCHORD=(nvals nbr1 val1 [nbr2 val2 ...])] - MV30
```

```

[CLASS=template_class] -
M V30 [SEQID=sequence_id] -
M V30 [SEQNAME=name] -
. . .
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 >= -1 0 = not specified (default)	-1 = H0, 0 = not specified (default), 1 = H1, 2 = H2, 3=H3, 4 =

Field	Meaning	Values	Notes
		-1 = zero	H4 [Query] H0 means no H atoms allowed unless explicitly drawn. Hn means atom must have n or more Hs in excess of explicit Hs.
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	

Field	Meaning	Values	Notes
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 property 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 support text values (for example, Al,Br). See Example of a V3000 molfile for Hybrid Representation for Sequences on page 29 for the hybrid representation for sequences (template representation of residues).
CLASS	Provides the class information for a collapsed template atom.	String. One of the following classes: AA dAA DNA RNA SUGAR BASE PHOSPHATE LINKER CHEM	Example: for a collapsed alanine template atom (AA/Ala), its type=Ala and CLASS=AA. CHEM class: represents attachments (for HELM strings converted to SCSR).
SEQID	Captures residue sequence id information.	Positive integer	This property supports a positive integer value to capture residue sequence id information for a template.
SEQNAME	Specifies the name used to represent the collapsed template atom in sequence views.	String	Optional keyword. It is mostly used to specify a sequence name for a group of collapsed granular nucleic acid templates that belong to the same nucleotide and have the same sequence ID. The name can be present in any of the templates in the group. For example, for an Adenine

Field	Meaning	Values	Notes
			nucleotide represented by separate ribose, adenine base, and phosphate templates, SEQNAME='A' specifies to use 'A' to represent the nucleotide in the sequence string.

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 -
M V30 [CFG=val] -
M V30 [TOPO=val] -
M V30 [RXCTR=val] -
M V30 [STBOX=val] -
M V30 [ATTACH=[ALL|ANY]] -
M V30 [ENDPTS]=(natoms atom1 atoms2 [atom3 ...])
M V30 [DISP=[HBOND1|HBOND2|COORD|DATIVE]] -
...
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 9 = coordination 10 = hydrogen	Types 4 through 8 are for queries only. Type 9 has display options for dipolar bonds: COORD or DATIVE COORD for coordinative bonds used in metal complexes DATIVE for dative bonds used in molecules such as Lewis acids and bases Type 10 has display options: HBOND1 or HBOND2
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	

Field	Meaning	Values	Notes
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 change 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.
ENDPTS	multiple endpoint	ALL or ANY	One-to-all for organometallics or one-to-any for generics. For example, ENDPTS=(5 1 2 3 4 5) ATTACH=ALL

Link Atom Line

One link atom line exists for each link atom in the CTAB. A link atom line has the format:

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

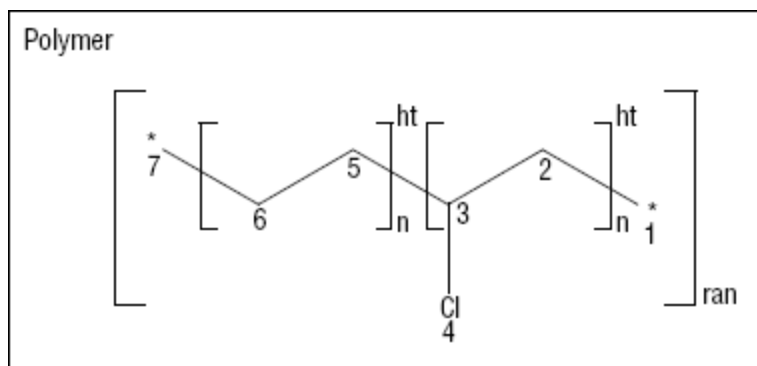
Meaning of values in link lines

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	

Field	Meaning	Values	Notes
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 68.



Connection Table Organization of an Sgroup Structure

Header Block		Comment Line
		Polymer 07189510252D 10.003740.000000 Figure 5, J. Chem. Inf. Comput. Sci., Vol 32, No. 3., 1992 00000999 V3000
		M V30 BEGIN CTAB
Counts Line		M V30 COUNTS 7 6 3 0 0
		M V30 BEGIN ATOM
Atom Block		M V30 4 C1 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
Bond Block		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 MV30 6 1 7 6
		M V30 END BOND
		M V30 BEGIN SGROUP
SGroup Block		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=(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

For information on the comment line, see [V3000 Header](#) on page 33.

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 [FIELDISP=fielddisp] -
M V30 [QUERYTYPE=querytype] [QUERYOP=queryop] -
M V30 [FIELDATA=fielddata] ... -
M V30 [CLASS=class] -
M V30 [SAP=(3 aidx lvidx id)]* -
M V30 [BRKTYP=bracketType] -
...
M V30 [SEQID=sequence_id] -
M V30 [NATREPLACE=[class]/name/[altname]] -
M V30 [SEQNAME=name] -
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	

Field	Meaning	Values	Notes
XBONDS	nxbonds is the number of crossing bonds. xbond is the crossing-bond index.	Integer > 0	
CBONDS	ncbonds is the number of containment bonds. cbond is the containment-bond index.	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	
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	

Field	Meaning	Values	Notes
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.
CSTATE	xbond is the crossing bond of the expanded abbreviation Sgroup.	Integer > 0	Display vector information for the contracted abbreviation Sgroup.

Field	Meaning	Values	Notes
	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	BIOVIA internal namespace - do not use: data Sgroups having a fieldname that starts with SMMX: (case insensitive). Denotes data that does not differentiate structures.
<p>Example for FIELDNAME, FIELDDISP, and FIELDDATA to describe a sequence chain:</p> <pre>MV30 1 DAT 1 ATOMS=(2 1 2) FIELDNAME=SMMX:CHAIN:type;name;color - MV30 FIELDDISP="0.00000.0000DAALL15" - MV30 FIELDDATA="active site;myPeptide;Green"</pre> <p>where active site, myPeptide, and Green correspond to type, name, and color.</p>			
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.
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.
FIELDINFO	fieldinfo is the program-specific field information.	Free-format string	Example: <type><units/format>.
QUERYTYPE	querytype indicates whether it is a query.	String. ' ' = not a query (default) 'IQ' = is a query	
QUERYOP	queryop is the query operator.	String. query operator	Examples: = 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	String. One of the following classes: AA	CHEM class: represents attachments (for HELM strings converted to SCSR).

Field	Meaning	Values	Notes
	class.	dAA DNA RNA SUGAR BASE PHOSPHATE LINKER CHEM LGRP MODAA MODdAA MODDNA MODRNA XLINKAA XLINKdAA XLINKDNA XLINKRNA	MOD prefix: indicates a modified natural template. XLINK prefix: indicates a cross-linked template.
SAP	aidx is the index of attachment point or potential attachment point atom.	Integer > 0	
	lidx 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	

Field	Meaning	Values	Notes
SEQID	This property supports a positive integer value to capture residue sequence id information.	* SEQID	
NATREPLACE	Specifies the natural residue replacement for modified templates.	A string: [class/]name [/altname]	The class of the natural replacement is optional. If not defined, it is assumed to be the same class as the template.
SEQNAME	Specifies the name used to represent the collapsed template atom in sequence views.	String	Optional keyword that applies to SUP Sgroups. It is mostly used to specify a sequence name for a group of expanded granular nucleic acid templates that belong to the same nucleotide and have the same sequence ID. The name can be present in any of the SUPs in the group. For example, for an Adenine nucleotide represented by separate ribose, adenine base, and phosphate SUP Sgroups, SEQNAME='A' specifies to use 'A' to represent the nucleotide in the sequence string.

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 hyphen (-) as the delimiter and uses that delimiter for both name and surname:

-surfactiveAgent-agent001-

If the first character 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.

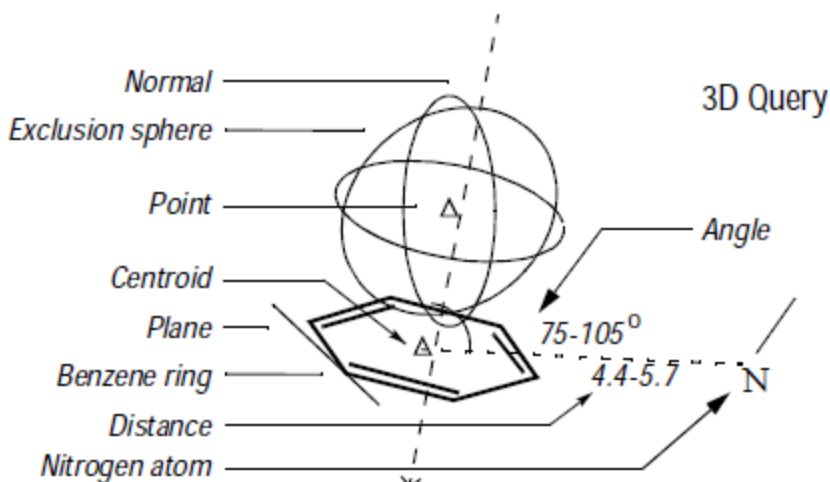
User-specified collections can use any other arbitrary naming conventions. The default action for all V3000 readers and writers provided by BIOVIA is to preserve persistent collections and internal collections for CTfile import/export operations. Internal collections can also be preserved if their contents 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/STERACn	"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/STERELn	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.



Connection Table Organization of an 3D Structure

Header Block	Comment Line
	3D Query 07189510253D 11.000000.000000 Figure 5, J. Chem. Inf. Comput. Sci., Vol 32, No. 3., 1992 00000999 V3000
	M V30 BEGIN CTAB
Counts Line	M V30 COUNTS 8 7 0 7 0
	M V30 BEGIN ATOM
Atom Block	M V30 1 C1 0.2892 1.1122 0 M V30 2 C -0.4562 0.6578 1.13156 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 1.4562 -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 1.17417 0 M V30 END ATOM
	M V30 BEGIN BOND
Bond Block	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
3D Object Block	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) MV30 3 -8 7 "" 0 0 BASIS=(2 03D.1 03D.2) MV30 4 -3 6 "" -2 0 BASIS=(2 03D.1 03D.3) PNTDIR=1 MV30 5 -16 12 "" 1.5 0 BASIS=(1 03D.4) UNCONNOK=1 MV30 6 -12 10 "" 75 105 BASIS=(3 03D.4 03D.1 7) MV30 7 -9 3 "" 4.4 5.7 BASIS=(2 7 03D.1) MV30 END OBJ3D
	M V30 END CTAB
	M END

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

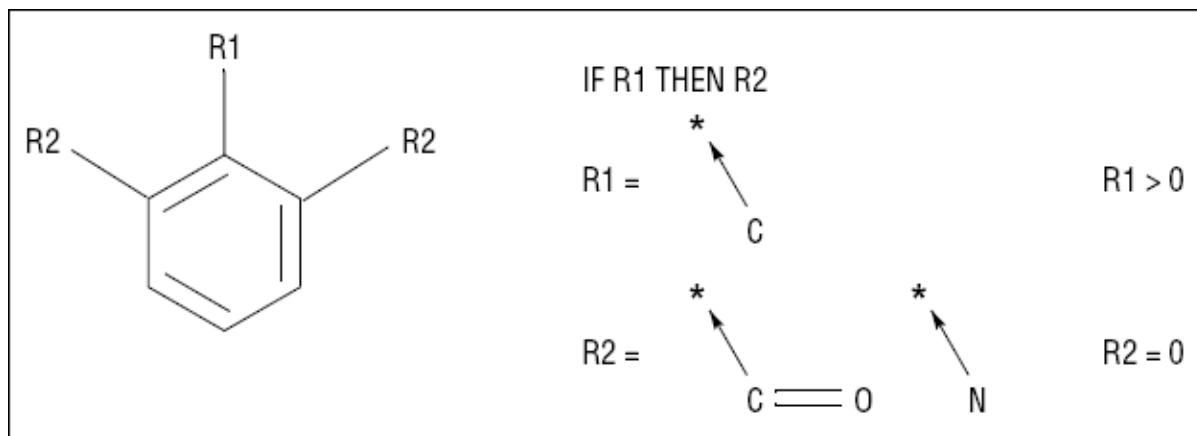
Field	Meaning	Values	Notes
index	3D object index	Integer > 0	The actual value of the index does not matter as long as all indexes are unique. However, extremely large numbers used as indexes can cause the program to fail to allocate memory for the correspondence array.
type	Object type	Integer < 0 for 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	

Field	Meaning	Values	Notes
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 41.
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	
UNCONNO K		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

Header Block		0 7189510252D 1 0.00353 0.00000 0 0 0 0 0 0 9 9 9 V3000		
Ctab for Rgroup Root	Counts Line	M V30 BEGIN CTAB M V30 COUNTS 9 9 0 0 0 M V30 BEGIN ATOM		
	Atom Block for Root	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)		
		Bond Block for Root	M V30 END ATOM M V30 BEGIN BOND MV30 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		
		Rgroup Logic Line	M V30 BEGIN RGROUP 1 M V30 RLOGIC 2 0 ""	
		Block for Rgroup 1	Ctab for Rgroup 1 Member 1	Counts Line
				Atom Block
			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	
		Block for Rgroup 2	Rgroup Logic Line	M V30 BEGIN RGROUP 2 M V30 RLOGIC 0 0 0
	Ctab for Rgroup 2 member 1		Counts Line	
			Atom Block	
			Bond Block	
	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			
	Ctab for Rgroup 2 Member 2		Counts Line	
			Atom Block	
			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			

For information on the comment line, see [V3000 Header](#) on page 33.

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]*
  [collection-block]
M V30 END Rgroup
```

Meaning of values in the Rgroup block

Field	Meaning	Values
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[...]] ]
  [NATREPLACE=[class]/name/[altname]] [ 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. The optional NATREPLACE keyword specifies the natural residue replacement for modified templates. The class of the natural replacement is optional. If not defined, it is assumed to be the same class as the template.

Example of a V3000 molfile for Hybrid Representation for Sequences

The hybrid representation uses templates to represent residues, which allows for significant compression of the structure stored in the database and transmitted in molfile format between different programs. This hybrid representation is formally known as Self-Contained Sequence Representation (SCSR). One advantage of the hybrid representation for biologic sequences is that repetition is represented in a compact manner.

```

1 A G meS A G
ACCLDraw03281611052D

0 0 0 0 0          999 v3000
M V30 BEGIN CTAB
M V30 COUNTS 5 4 1 0 1
M V30 BEGIN ATOM
M V30 1 A|a 1.5563 -9.2328 0 0 CLASS=AA ATTCHORD=(2 2 Br) SEQID=1
M V30 2 G|y 2.7137 -9.23 0 0 CLASS=AA ATTCHORD=(4 3 Br 1 A|) SEQID=2
M V30 3 meS 3.8712 -9.23 0 0 CLASS=AA ATTCHORD=(4 4 Br 2 A|) SEQID=3
M V30 4 A|a 5.4539 -9.23 0 0 CLASS=AA ATTCHORD=(4 3 A| 5 Br) SEQID=4
M V30 5 G|y 6.6114 -9.23 0 0 CLASS=AA ATTCHORD=(2 4 A|) SEQID=5
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 END BOND
M V30 BEGIN SGROUP
M V30 1 DAT 1 ATOMS=(1 1) FIELDNAME="SMMX:sequence position data" -
M V30 FIELDSDISP=" 1.5563 -9.2328 DA ALL 1 "
M V30 END SGROUP
M V30 END CTAB
M V30 BEGIN TEMPLATE
M V30 TEMPLATE 1 AA/A|a/A/
M V30 BEGIN CTAB
M V30 COUNTS 7 6 3 0 1
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 A1) 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 A1)
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 TEMPLATE 3 AA/meS/meS/ NATREPLACE=AA/S
```

```
M V30 BEGIN CTAB
M V30 COUNTS 9 8 3 0 1
M V30 BEGIN ATOM
M V30 1 C 9.9525 -5.6641 0 0
M V30 2 O 9.9451 -6.8641 0 0
M V30 3 N 7.3518 -5.6442 0 0 CFG=3
M V30 4 C 8.6579 -4.9049 0 0 CFG=1
M V30 5 C 8.6671 -3.4041 0 0
M V30 6 O 7.6319 -2.7971 0 0
M V30 7 C 6.3173 -5.0361 0 0
M V30 8 O 10.8217 -5.1697 0 0
M V30 9 H 7.3436 -6.6442 0 0
M V30 END ATOM
M V30 BEGIN BOND
M V30 1 2 2 1
M V30 2 1 1 4
M V30 3 1 4 3
M V30 4 1 4 5 CFG=1
M V30 5 1 5 6
M V30 6 1 3 7
M V30 7 1 1 8
M V30 8 1 3 9
M V30 END BOND
M V30 BEGIN SGROUP
M V30 1 SUP 1 ATOMS=(7 1 2 3 4 5 6 7) XBONDS=(2 7 8) CSTATE=(4 7 2.17 0.48
-
M V30 0) CSTATE=(4 8 -1.31 -0.99 0) LABEL=meS CLASS=AA SAP=(3 1 8 Br) -
M V30 SAP=(3 3 9 A1) NATREPLACE=AA/S
M V30 2 SUP 2 ATOMS=(1 8) XBONDS=(1 7) CSTATE=(4 7 -2.17 -0.48 0) LABEL=OH
-
M V30 CLASS=LGRP
M V30 3 SUP 3 ATOMS=(1 9) XBONDS=(1 8) CSTATE=(4 8 1.31 0.99 0) LABEL=H -
M V30 CLASS=LGRP
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 END TEMPLATE
M END
```

Chapter 3: Molfile

The V3000 molfile and V3000 rxnfile formats support the latest features of BIOVIA chemical representation, see [Advantages of V3000 Format](#) on page 1.

V3000 is both a superset of V2000 but a different format. For information on the earlier V2000 format, see [V2000 Connection Table \(CTAB\)](#) on page 41.

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.

```
1 [name or blank]
2          06031011012D 1    1.00000    0.00000    0
3 [comment or blank]
4    0 0 0 0 0 0          999 V2000
5 M  END
```

```
1 [name or blank]
2          06031011022D 1    1.00000    0.00000    0
3 [comment or blank]
4    0 0 0 0 0 0          999 V3000
5 M  V30 BEGIN CTAB
6 M  V30 COUNTS 0 0 0 0 0
7 M  V30 END CTAB
8 M  END
```

Example of molfile - Alanine

ACCLDraw12161507522D

```
    0 0 0 0 0          999 v3000
M  V30 BEGIN CTAB
M  V30 COUNTS 5 4 0 0 1
M  V30 BEGIN ATOM
M  V30 1 N 16.998 -17.4463 0 0
M  V30 2 C 16.998 -16.2972 0 0 CFG=1
M  V30 3 C 18.1179 -15.9033 0 0
M  V30 4 O 18.1179 -14.7224 0 0
M  V30 5 C 16.0183 -15.6308 0 0
M  V30 END ATOM
M  V30 BEGIN BOND
M  V30 1 1 2 5 CFG=3
M  V30 2 2 3 4
M  V30 3 1 2 3
M  V30 4 1 1 2
M  V30 END BOND
M  V30 BEGIN COLLECTION
M  V30 MDLV30/STEABS ATOMS=(1 2)
M  V30 END COLLECTION
M  V30 END CTAB
M  END
```

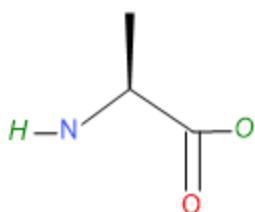
The structure of the molfile is:

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

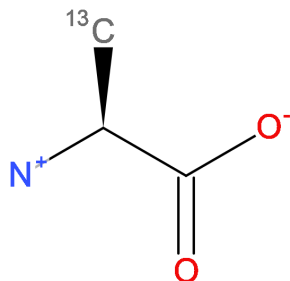
Note: This example does not have the following CTAB blocks: Sgroup, Rgroup, 3D, Template.

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

BIOVIA Draw version of alanine:



Pipette Sketcher 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: <pre>IIPPPPPPPMDDYYHHmddSSSSSSSSSSSEEEEEEEEEERRRRR A2<--A8--><---A10-->A2I2<--F10.5--><---F12.5--><-I6->)</pre> 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)

	<p>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.</p> <p>Note: A blank line can be substituted for line 2.</p>
Line 3:	A line for comments. If no comment is available, a blank line must be present.
Line 4:	<p>This line contains the version number, V3000</p> <p>The number of appendix lines is always written as 999, regardless of how many there actually are. See Counts Line on page 7.</p> <p>(For V2000, Line 4 is the no-structure counts line.)</p>

Notes:

- 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 "".

Chapter 4: RGfiles (Rgroup file)

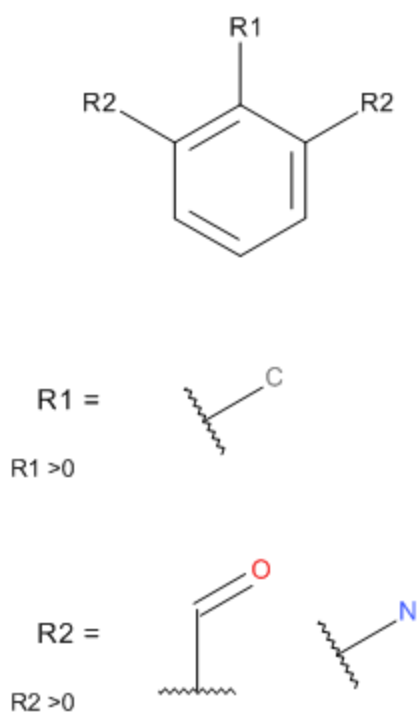
The format of an RGfile (Rgroup file) is shown below. Rgroups are also known as Markush structures. In addition to the primary connection table (CTAB block) for the root structure, a CTAB block defines each member within each Rgroup.

For an example of the header, see [The Connection Table](#) on page 6.

For a description of the Rgroup, see [The Rgroup](#) on page 25.

Example of an RGfile (Rgroup file)

Rendering in BIOVIA Draw.



The following is the RGfile corresponding 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  ACCLDraw12231508522D
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 3.509 -8.7214 0 0
9 M   V30 2 C 5.5513 -8.7208 0 0
10 M  V30 3 C 4.5321 -8.1311 0 0
11 M  V30 4 C 5.5513 -9.9023 0 0
12 M  V30 5 C 3.509 -9.9076 0 0
13 M  V30 6 C 4.5347 -10.4918 0 0
14 M  V30 7 R# 2.4858 -8.1307 0 0 RGROUPS=(1 2)
15 M  V30 8 R# 4.5321 -6.9496 0 0 RGROUPS=(1 1)
16 M  V30 9 R# 6.5745 -8.1301 0 0 RGROUPS=(1 2)
17 M  V30 END ATOM
18 M  V30 BEGIN BOND
19 M  V30 1 1 6 4
20 M  V30 2 2 5 6
21 M  V30 3 1 2 3
22 M  V30 4 1 1 5
23 M  V30 5 2 4 2
24 M  V30 6 2 3 1
25 M  V30 7 1 1 7
26 M  V30 8 1 3 8
27 M  V30 9 1 2 9
28 M  V30 END BOND
29 M  V30 END CTAB
30 M  V30 BEGIN RGROUP 1
31 M  V30 RLOGIC 0 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 5.0898 -12.4244 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 3.8627 -16.0165 0 0 ATTCHPT=1
45 M  V30 2 O 4.8856 -15.426 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 7.1764 -16.0755 0 0 ATTCHPT=1
55 M  V30 END ATOM
56 M  V30 END CTAB
```

```
57 M V30 END RGROUP  
58 M END
```

Chapter 5:

Rxnfile

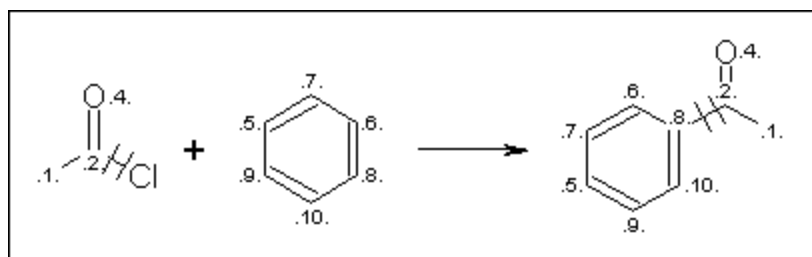
Rxnfiles contain structural data for the reactants and products of a reaction. This section covers the preferred V3000 reaction file format. For the V2000 version of this reaction file format, see [V2000 Rxnfile Overview](#) on page 70.

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: IIIIIIPPPPPPPPMDDYYYYHHmmRRRRRRR <-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 "MREG" line. See Large REGNO on page 57.
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 the following example.



Example V3000 Rxnfile for the Acylation of Benzene

Header Block		\$RXN V3000 05030217387439
		M V30 COUNTS 2 1
		M V30 BEGIN REACTANT
		M V30 BEGIN CTAB
Reactant block	Ctab for first reactant	M V30 COUNTS 4 3 0 0 0
		M V30 BEGIN ATOM
		M V30 1 C 0.323 -0.2377 0 1
		M V30 2 C -1.0362 -0.9618 0 2
		M V30 3 O 0.323 1.4154 0 3
		M V30 4 Cl 1.6423 -1.0307 0 0
	M V30 END ATOM	
	M V30 BEGIN BOND	
	M V30 1 1 1 2 RXCTR=2	
	M V30 2 2 1 3 RXCTR=2	
	M V30 3 1 1 4 RXCTR=4	
	M V30 END BOND	
	M V30 END CTAB	
	M V30 BEGIN CTAB	
	M V30 COUNTS 6 6 0 0 0	
	M V30 BEGIN ATOM	
	M V30 1 C 1.3331 -0.7694 0 5	
	M V30 2 C 1.3331 0.7694 0 6	
M V30 3 C 0 -1.5417 0 7		
M V30 4 C 0 1.5417 0 8		
M V30 5 C -1.3331 -0.7694 0 9		
M V30 6 C -1.3331 0.7694 0 10		
M V30 END ATOM		
M V30 BEGIN BOND		
M V30 1 1 1 2 RXCTR=2		
M V30 2 2 1 3 RXCTR=2		
M V30 3 2 2 4 RXCTR=2		
M V30 4 1 3 5 RXCTR=2		
M V30 5 1 4 6 RXCTR=2		
M V30 6 2 5 6 RXCTR=2		
M V30 END BOND		
M V30 END CTAB		
M V30 END REACTANT		
M V30 BEGIN PRODUCT		
M V30 BEGIN CTAB		
M V30 COUNTS 9 9 0 0 0		
M V30 BEGIN ATOM		
M V30 1 C -0.5331 -0.1358 0 5		
M V30 2 C -1.8606 0.633 0 6		
M V30 3 C -0.5331 -1.6992 0 7		
M V30 4 C 0.8201 0.6305 0 1		
M V30 5 C -3.2189 -0.1358 0 8		
M V30 6 C -1.8811 -2.4731 0 9		
M V30 7 C 2.1297 -0.1128 0 2		
M V30 8 O 0.8534 2.2297 0 3		
M V30 9 C -3.2292 -1.6863 0 10		
M V30 END ATOM		
M V30 BEGIN BOND		
M V30 1 1 1 2 RXCTR=2		
M V30 2 2 1 3 RXCTR=2		
M V30 3 1 1 4 RXCTR=		
M V30 4 2 2 5 RXCTR=2		
M V30 5 1 3 6 RXCTR=2		
M V30 6 1 4 7 RXCTR=2		
M V30 7 2 4 8 RXCTR=2		
M V30 8 1 5 9 RXCTR=2		
M V30 9 2 6 9 RXCTR=2		
M V30 END BOND		
M V30 END CTAB		
M V30 END PRODUCT		

Chapter 6:

V2000 Connection Table (CTAB)

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 BIOVIA file formats.

This chapter describes the legacy V2000 format. The current V3000 format supports more features. For the V3000 version, see [Connection Table \(CTAB\)](#) on page 5.

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.
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:

Notes:

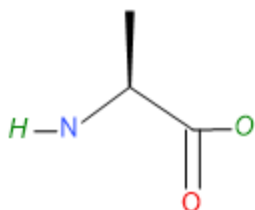
A blank numerical entry on any line should be read as "0" (zero). Spaces are significant and correspond to one or more of the following:

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

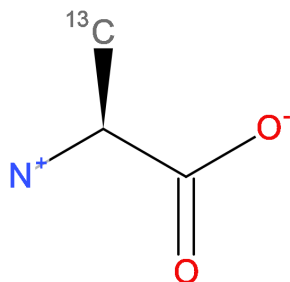
Example: Alanine in V2000 format

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

BIOVIA Draw version of alanine:



Pipette Sketcher version of alanine:



The connection table of alanine:

```

1
2  ACCLDraw12231509452D
3
4      6  5  0  0  1  0  0  0  0  0999 v2000
5      3.5938  -8.5438  0.0000 N  0  3  0  0  0  0  0  0  0  0  0
5      4.6169  -7.9530  0.0000 C  0  0  2  0  0  0  0  0  0  0  0
7      5.6401  -8.5438  0.0000 C  0  0  0  0  0  0  0  0  0  0  0
8      6.6633  -7.9530  0.0000 O  0  5  0  0  0  0  0  0  0  0  0
9      5.6401  -9.7252  0.0000 O  0  0  0  0  0  0  0  0  0  0  0
10     4.6169  -6.7716  0.0000 C  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  END
18

```

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.

The Counts Line

```
aaabbb111ffcccsssxxrrpppiiimmvwww
```

Where:

aaa	= number of atoms	[Generic]
bbb	= number of bonds	[Generic]
lll	= number of atom lists	[Query]
fff	= (obsolete)	
ccc	= chiral flag: 0=not chiral, 1=chiral	[Generic]
sss	= (obsolete)	
xxx	= (obsolete)	
rrr	= (obsolete)	
ppp	= (obsolete)	
iii	= (obsolete)	
mmm	= number of lines of additional properties, including the MEND line. No longer supported. The default is set to 999.	[Generic]

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 0999 v2000
```

The Atom Block

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

```
xxxxx.xxxxxyyyy.yyyyzzzz.zzzz aaaddcccssshhhbbbvwwwHHHrrriimmnnnee
```

where the values are described in the following table:

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#	[Generic, Query, 3D,

Field	Meaning	Values	Notes
		for Rgroup label	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, 1, 2, 3, 4, 5, 6, 7 (0 if uncharged; 1, 2, 3 if positive charges where 1 = +3, 2 = +2, 3 = +1; 4 if doublet radical; 5, 6, 7 if negative charges where 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 Hs in excess of explicit Hs.

Field	Meaning	Values	Notes
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 Hs.
HHH	H0 designator	0 = not specified (default) 1 = no H atoms allowed	Redundant with hydrogen count information.
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.

The Bond Block

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

```
111222tttsssxxrrrccc
```

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); Also possible: 5 = (4 + 1), 9 = (8 + 1), and 13 = (12 + 1)	[Reaction, Query]

The Atom List Block[Query]

Note: Current BIOVIA 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 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 *MXXX* where two spaces separate the *M* and *XXX*. Exceptions are:

- *A aaa, V aaa vvvvvv,* and *G aaappp,* which indicate the following properties: atom alias, atom value, and group abbreviation, 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 [not used in current products]

A	aaa
X	. . .
aaa:	Atom number
X:	Alias text

Atom Value

V aaa	
v...	
aaa:	Atom number
v:	Value text

Supported Molecular Objects

The substructure mapper in Pipeline Pilot Chemistry interprets the following values in the Atom Value property of the CTAB.

- Molecule
- Atom
- Bond
- Atomset (collection of atoms)
- Bondset (collection of bonds)
- Macromolecule (proteins, nucleic acids read from a PDB file)
- Chain
- Residue
- Reaction
- SGMapper (Subgraph mapper)
- SGMMap (Subgraph map returned by the Subgraph mapper)
- MolecularUtilities

In PilotScript, only molecules, atoms, bonds, and reactions are directly supported in the Molecular API. You can create your own atom and bond sets using `Bitset` functions. Functionality to perform substructure mapping and other molecular utilities is provided via the `RunComponent` functions.

The atom values used in Pipeline Pilot Chemistry for the substructure mapper are:

- aliph/aliphatic
- arom/aromatic
- sp/sp2/sp3
- nomap
- clip/extend (for reactions)

Group Abbreviation [not used in current products]

G aaapp	
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...).

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 MCHG or MRAD 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 Count [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 11112222333344445555...
--

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 *BIOVIA 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
-----	---------------------------------------

Notes:

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

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, this appendix is written only 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.
------	---

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.
------	---

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

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 ...

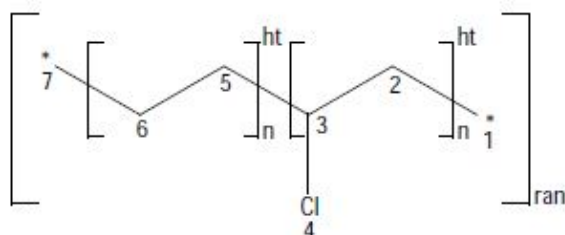
aaa: Atoms in Sgroup sss

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.)

Polymer



```

10179110412D 1 0.00374 0.00000 0
7 6 0 0 0 0 16 V2000
2.9463 0.3489 0.0000 * 0 0 0 0 0 0
1.6126 1.1189 0.0000 C 0 0 0 0 0 0
0.2789 0.3489 0.0000 C 0 0 3 0 0 0
0.2789 -1.1911 0.0000 Cl 0 0 0 0 0 0
-1.0548 1.1190 0.0000 C 0 0 0 0 0 0
-2.3885 0.3490 0.0000 C 0 0 0 0 0 0
-3.9246 1.1470 0.0000 * 0 0 0 0 0 0
1 2 1 0 0 0
2 3 1 0 0 0
3 4 1 0 0 0
5 6 1 0 0 0
5 3 1 0 0 0
7 6 1 0 0 0
M STY 3 1 5RU 2 5RU 3 COP
M SST 1 3 RAN
M SLB 3 1 5 2 6 3 7
M SCN 2 1 HT 2 HT
M SAL 1 2 5 6
M SBL 1 2 5 6
M SDI 1 4 -0.6103 1.2969 -0.6103 0.1710
M SDI 1 4 -3.1565 0.1850 -3.1565 1.3110
M SAL 2 3 2 3 4
M SBL 2 2 1 5
M SDI 2 4 2.2794 1.2969 2.2794 0.1709
M SDI 2 4 -0.1657 0.1710 -0.1657 1.2969
M SAL 3 7 1 2 3 4 5 6 7
M SDI 3 4 3.6382 1.6391 3.6382 -1.7685
M SDI 3 4 -4.7070 -1.7685 -4.7070 1.6391
M END
  
```

Number of entries on line

Type
Subtype
Label
Connectivity

Header block
Counts line
Atom block
Bond block
Atom list block
Sext block
General Sgroup Info
Sgroup 1
Sgroup 2
Sgroup 3
Sgroup properties
Ctab block

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...ffgghhh...hhhiijj...	
sss:	Index of data Sgroup.
fff...fff:	30 character field name - no blanks, commas, or hyphens.
gg:	Field type - F = formatted, N = numeric, T = text (ignored).
hhh...hhh:	20-character field units or format.
ii:	IQ if data line is a query rather than Sgroup data. Blank if it is not a query.
jjj...:	Data query operator.

Data Sgroup Display Information [Sgroup]

M SDD sss xxxxx.xxxxxyyyy.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]

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

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 000 ...	
sss:	Index of component Sgroup
000:	Integer component order (1...256). This limit applies only to MACCS-II

3D Feature Properties [3D]

M	\$3Dnnn	
M	\$3D. . . :	See The Properties Block for 3D Features 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.yyyzzzzz.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 Draw abbreviation Sgroup conversion mechanism. Draw 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 Direct databases. The hydrogen-only leaving groups are converted to PXA appendices for registration, and converted back when Draw 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. Draw 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 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, AA, ...) 69 characters maximum

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 superseded.

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
```

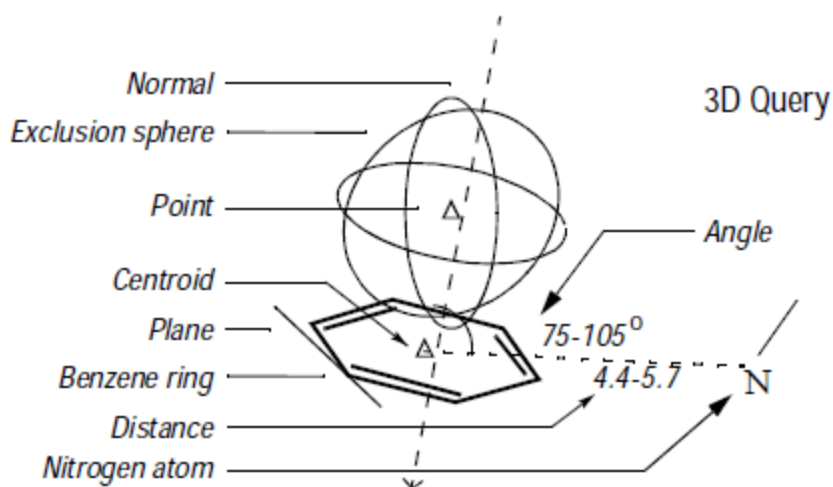
This entry goes at the end of the properties block and is required for molfiles which contain a version stamp in the counts line.

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

		Header Block	3D Query 10179109553D 1 1.00000 0.00000 0	
		Counts Line	8 7 0 0 0 0 18V2000	
Ctab Block	Atom Block		1.0252 0.2892 1.1122C 00 00 00 -0.4562 0.6578 1.3156C 00 00 00 -1.4813 0.3687 0.2033C 00 00 00 -1.0252 -0.2892 -1.1122C 00 00 00 0.4562 -0.6578 -1.3156C 00 00 00 1.4813 -0.3687 -0.2033C 00 00 00 4.1401 -0.1989 1.3456N 00 00 00 4.6453 0.5081 1.7417C 00 00 00	
			12 1 0 0 0 23 2 0 0 0 34 1 0 0 0 45 2 0 0 0 56 1 0 0 0 61 2 0 0 0 78 1 0 0 0	
		Bond Block		
		Properties Block	3D Features Count	M \$3D 7
			Centroid	M \$3D -7 6 M \$3D 3 M \$3D 6 4 2 M \$3D -5 13
			Plane	M \$3D 6 0.0000 M \$3D 1 2 3 4 5 6
			Normal to Plane	M \$3D -8 7 M \$3D 9 10
			Point	M \$3D -3 6 M \$3D 9 11 -2.0000
			Exclusion Sphere	M \$3D -16 12 M \$3D 12 1 0 1.5000
			Angle	M \$3D -12 10 M \$3D 12 9 7 75.0000 105.0000
Distance	M \$3D -9 3 M \$3D 7 9 4.4000 5.7000 M END			
		Atom List Block Stext Block		

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 BIOVIA programs (see "3D data constraints [3D, Query]")

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	<p>The data line for a point defined by two points and a distance (Å) has the following format:</p> <pre>M \$3Diiiijjjddddd.dddd</pre> <p>where the variables represent:</p> <p>iii: ID number of a point jjj: ID number of a second point dddd.dddd: Distance from first point in direction of second point (Å), 0 if not used</p> <p>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.</p> <pre>M \$3D -14 POINT_1 M \$3D132.0000</pre>
-2	<p>The data line for a point defined by two points and a percentage has the format:</p> <pre>M \$3Diiiijjjddddd.dddd</pre> <p>where the variables represent:</p> <p>iii: ID number of a point jjj: ID number of a second point dddd.dddd: Distance (fractional) relative to distance between first and second points, 0 if not used</p>
-3	<p>The data line for a point defined by a point, a normal line, and a distance (Å) has the format:</p> <pre>M \$3Diii111ddddd.dddd</pre> <p>where the variables represent:</p> <p>iii: ID number of a point 111: ID number of a normal line dddd.dddd: Distance (Å), 0 if not used</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>Note: For chiral models, the distance value is signed to specify the same or opposite direction of the normal.</p> </div>
-4	<p>The data lines for a best fit line defined by two or more points have the following format:</p> <pre>M \$3Dppptttt.tttt M \$3Diiiijjj...zzz . . .</pre> <p>where the variables represent:</p> <p>ppp: Number of points defining the line tttt.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)</p> <p>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.</p>

Type	Description of Data Line
	<pre>M \$3D -42 N_TO_AROM M \$3D41.2000 M \$3D1 14 15 19</pre>
-5	<p>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:</p> <pre>M \$3Dpppttttt.tttt M \$3Diii jjj...zzz ...</pre> <p>where the variables represent:</p> <p>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)</p> <p>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.</p> <pre>M \$3D -54 PLANE_2 M \$3D30.0000 M \$3D1 5 14</pre>
-6	<p>The data line for a plane defined by a point and a line has the following format:</p> <pre>M \$3Diii111</pre> <p>where the variables represent:</p> <p>iii: ID number of a point 111: ID number of a line</p> <p>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.</p> <pre>M \$3D -63 PLANE_1 M \$3D1 16</pre>

Type	Description of Data Line
-7	<p>The data lines of a centroid defined by points have the following format:</p> <pre>M \$3Dppp M\$3Di i i j j j . . . z z z . . .</pre> <p>where the variables represent: ppp: Number of points defining the centroid i i i: Each i i i, j j j, and z z z is the ID number j j j of an item in the model that defines the centroid j j j . . . z z z: (maximum of 20 items per data line).</p> <p>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.</p> <pre>M \$3D -71 ARO_CENTER M\$3D3 M \$3D68 10</pre>
-8	<p>The data line for a normal line defined by a point and a plane has the following format:</p> <pre>M \$3Di i i j j j</pre> <p>where the variables represent: i i i: ID number of a point j j j: ID number of a plane</p> <p>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.</p> <pre>M \$3D -81 ARO_NORMAL M\$3D 14 15</pre>
-9	<p>The data line for a distance defined by two points and a range (Å) has the following format:</p> <pre>M \$3Di i i j j j d d d d d . d d d z z z z z . z z z z</pre> <p>where the variables represent: i i i: ID number of a point j j j: ID number of a second point d d d d d . d d d d: Minimum distance (Å) z z z z z . z z z z: Maximum distance (Å)</p> <p>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.</p> <pre>M \$3D -96 L M \$3D1 144.90006.0000</pre>
-10	<p>The data line for a distance defined by a point, line, and a range (Å) has the format:</p> <pre>M \$3Di i i l l l d d d d d . d d d z z z z z . z z z z</pre> <p>where the variables represent: i i i: ID number of a point l l l l: ID number of a line d d d d d . d d d d: Minimum distance (Å)</p>

Type	Description of Data Line
	zzzzz.zzzz: Maximum distance (Å)
-11	<p>The data line for a distance defined by a point, plane, and a range (Å) has the format:</p> <pre>M \$3Di i i j j d d d d d . d d d d z z z z z . z z z z</pre> <p>where the variables represent:</p> <p>iii: ID number of a point jjj: ID number of a plane d d d d d . d d d d d : Minimum distance (Å) z z z z z . z z z z z : Maximum distance (Å)</p>
-12	<p>The data line for an angle defined by three points and a range (in degrees) has the following format:</p> <pre>M \$3Di i i j j k k k d d d d d . d d d d z z z z z . z z z z</pre> <p>where the variables represent:</p> <p>iii: ID number of a point jjj: ID number of a second point kkk: ID number of a third point d d d d d . d d d d d : Minimum degrees z z z z z . z z z z z : Maximum degrees</p> <p>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.</p> <pre>M \$3D-125 THETA1 M \$3D5 17 1680.0000105.0000</pre>
-13	<p>The data line for an angle defined by two lines and a range (in degrees) has the following format:</p> <pre>M \$3D111 m m m d d d d d . d d d d z z z z z . z z z z</pre> <p>where the variables represent:</p> <p>111: ID number of a line mmm: ID number of a second line d d d d d . d d d d d : Minimum degrees z z z z z . z z z z z : Maximum degrees</p> <p>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.</p> <pre>M \$3D-135 THETA2 M \$3D 27 2645.000080.0000</pre>
-14	<p>The data line for an angle defined by two planes and a range (in degrees) has the following format:</p> <pre>M \$3Di i i j j j d d d d d . d d d d z z z z z . z z z z</pre> <p>where the variables represent:</p> <p>iii: ID number of a plane jjj: ID numbers of a second plane d d d d d . d d d d d : Minimum degrees</p>

Type	Description of Data Line
	zzzzz.zzzz: Maximum degrees
-15	<p>The data line for a dihedral angle defined by four points and a range (in degrees) has the following format:</p> <pre>M \$3Diiiijjkkkllldddd.ddddzzzz.zzzz</pre> <p>where the variables represent:</p> <p>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 dddd.dddd: Minimum degrees zzzz.zzzz: Maximum degrees</p> <p>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.</p> <pre>M \$3D-155 DIHED1 M \$3D764845.000080.0000</pre>
-16	<p>The data lines for an exclusion sphere defined by a point and a distance (Å) have the following format:</p> <pre>M \$3Diiiuuuaaaddddd.dddd M \$3Dbbbccc...zzz ...</pre> <p>where the variables represent:</p> <p>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 dddd.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)</p> <p>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.</p> <pre>M \$3D-167 EXCL_SPHERE M \$3D 24015.0000 M \$3D9</pre>

Type	Description of Data Line
-17	<p>The data lines of the fixed atoms have the following format:</p> <pre>M \$3Dppp M \$3Di i i j j j . . . z z z . . .</pre> <p>where the variables represent:</p> <p>ppp: Number of fixed points i i i: Each i i i, j j j, and z z z is an ID number of a fixed atom j j j z z z: (to maximum of 20 items per data line)</p> <p>The following examples shows 4 fixed atoms. The first line is the identification line. The second and third lines are the data lines.</p> <pre>M \$3D-17 M \$3D4 M \$3D37 12 29</pre>

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...aaabbbbbbbppppppppppss...sss
M $3Di i i j j j d d d . . . d d d
```

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.
i i i:	ID number of an atom.
j j j:	ID number of a second atom for atom-pair data, 0 if data is atom data.
d d d . . . d d d:	Data constraint (based on format from database) (up to 64 characters).

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.

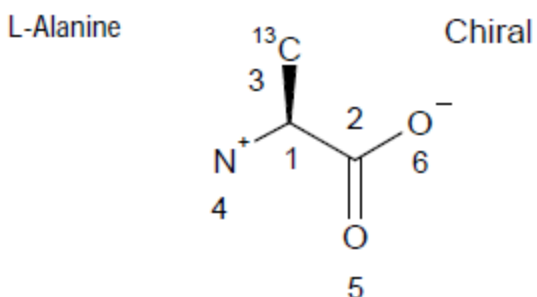
```
M $3D 120 CHARGE M$3D9990 @
```

Chapter 7:

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

Connection Table (Ctab)	<i>Header Block</i>	L-Alanine (13C) 10169115362D 1 0.00366 0.00000 0
	<i>Counts Line</i>	6 5 0 0 1 0 3 V2000
	<i>Atom Block</i>	-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	
	<i>Bond Block</i>	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		
<i>Properties Block</i>	M ISO 1 3 13	
	M END	

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

Header Block for V2000 Molfile

Line 1:	<p>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.</p> <p>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).</p>
Line 2:	<p>This line has the format:</p> <pre> IIPPPPPPPMDDYYHHmmdSSSSSSSSSSSEEEEEEEEEERRRRRR A2<--A8--><---A10-->A2I2<--F10.5-><---F12.5--><-I6->) </pre> <p>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.</p> <p>A blank line can be substituted for line 2.</p> <p>If the internal registry number is more than 6 digits long, it is stored in an M REG line. See Large REGNO on page 57.</p>
Line 3:	<p>A line for comments. If no comment is entered, a blank line must be present.</p>

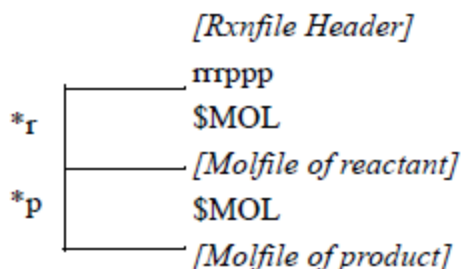
Chapter 8:

V2000Rxnfile

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 72.

The format is:



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: <pre>IIIIIPPPPPPPPMDDYYHHmmRRRRRRR <-A6-><---A9--><---A12-----><--I7->)</pre> <p>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 57.</p> <div style="border: 1px solid black; padding: 5px;"><p>Note: In rxnfiles produced by earlier product versions, the year occupied two digits instead of four. There are corresponding minor changes in the adjacent fields. The format of the line is: <pre>IIIIIPPPPPPPPMDDYYHHmmRRRRRRR <-A6-><---A10--><---A10--><--I8-->)</pre></p></div>
	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

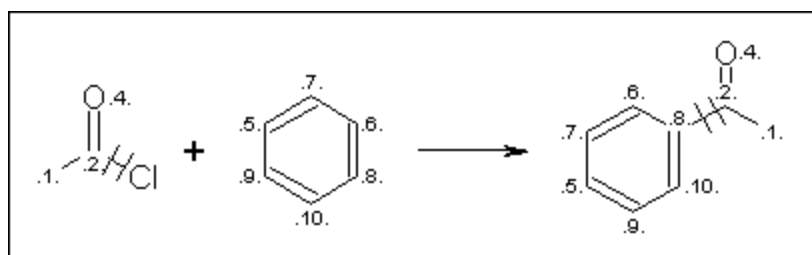
where the variables represent:

rrr	Number of reactants
ppp	Number of products

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](#) corresponds to the following reaction which includes atom-atom mapping:



V2000 Rxnfile for the Acylation of Benzene

<i>Header Block</i>	\$RXN 1017911041 7439
<i>#Reactants & #Products</i>	2 1
<i>Molfile delimiter</i>	\$MOL
<i>Molfile for 1st reactant</i>	REACCS8110179110412D 1 0.00380 0.00000 315 4 3 0 0 0 01 V2000 0.3929 -0.2577 0.0000 C 0 0 0 0 0 0 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 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 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 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
<i>Molfile delimiter</i>	\$MOL
<i>Molfile for 2nd reactant</i>	10179110412D 1 0.00371 0.00000 8 6 6 0 0 0 01 V2000 1.3335 -0.7689 0.0000 C 0 0 0 0 0 0 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 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 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 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 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 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
<i>Molfile delimiter</i>	\$MOL
<i>Molfile for product</i>	10179110412D 1 0.00374 0.00000 255 9 9 0 0 0 01 V2000 -0.5311 -0.1384 0.0000 C 0 0 0 0 0 0 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 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 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 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 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 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 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 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 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

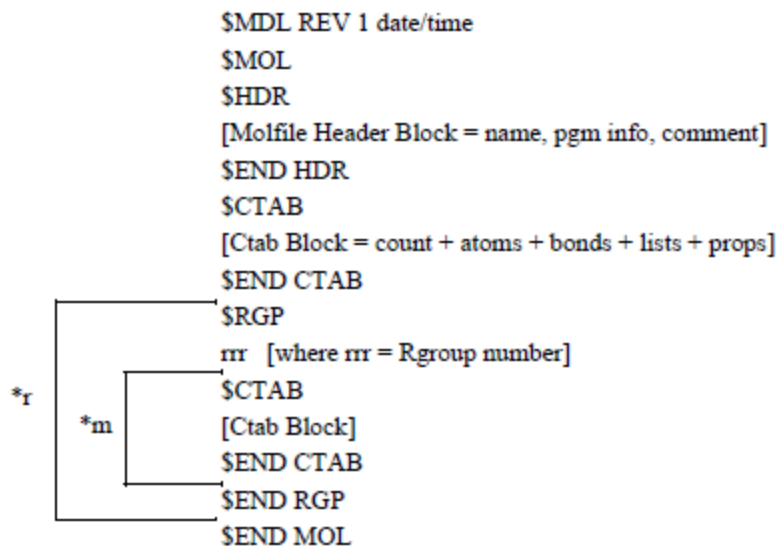
Chapter 9:

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).

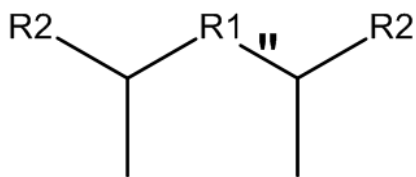


where:

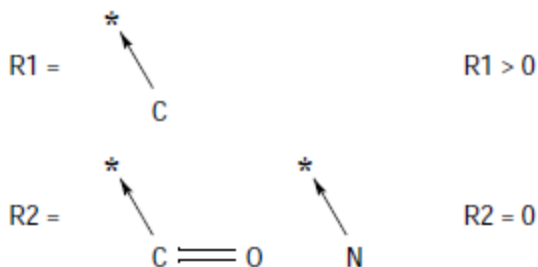
*r (Rgroup) is repeated (limit of 32 Rgroups).

*m (member) is repeated.

The [Example of an RGfile](#) on page 74 corresponds to the following Rgroup query:



IF R1 THEN R2



Example of an RGfile

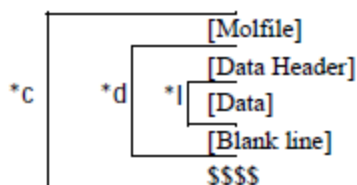
Rgfile header block		\$MDL REV 1 16OCT91 15:40 \$MOL \$HDR 10169115402D 1 0.00353 0.00000 0
Molfile header block		\$END HDR
Ctab for Rgroup root	Counts Line	\$CTAB 9 9 0 0 0 04 V2000 1.3337 0.7700 0.0000 C 0 0 0 0 0 0
	Atom block of root	1.3337 -0.7700 0.0000 C 0 0 0 0 0 0 0.0000 3.0800 0.0000 R# 0 0 0 0 0 0 2.6674 1.5400 0.0000 R# 0 0 0 0 0 0 -2.6674 1.5400 0.0000 R# 0 0 0 0 0 0
	Bond block of root	#of entries on line RGroup Label Location And Rgroup number 1 2 1 0 0 0 3 9 1 0 0 0
	Properties block of root	M RGP 3 7 1 8 2 9 2 M LOG 1 1 2 0 M LOG 1 2 0 0 0
	If/then, RestH, Occurrence	M END
Block for Rgroup R1		\$END CTAB \$RGP 1 \$CTAB 1 0 0 0 0 02 V2000 12.2100 14.3903 0.0000 C 0 0 0 0 0 0 M APO 1 1 1 M END \$END CTAB \$END RGP
Block for Rgroup R2	Ctab for Rgroup 2 Member 1	\$RGP 2 \$CTAB 2 1 0 0 0 02 V2000 -1.4969 0.0508 0.0000 C 0 0 0 0 0 0 0.0431 0.0508 0.0000 O 0 0 0 0 0 0 1 2 2 0 0 0 M APO 1 1 1 M END \$END CTAB
	Ctab for Rgroup 2 Member 2	\$CTAB 1 0 0 0 0 02 V2000 12.2100 14.3903 0.0000 N 0 0 0 0 0 0 M APO 1 1 1 M END \$END CTAB \$END RGP
		\$END MOL

Chapter 10:

SDfiles (multiple structures and optional data)

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
> DT1255
> (MD-0894)<BOILING_POINT>FROM ARCHIVES
```

Example of an SDfile

Compound	Molfile	Header block		1,2 CYCLO-C6 DI-COOH TRANS,L 06039016292D 1 0.00339 0.00000 25
		Connection table		12 12 0 0 1 01 V2000 -0.0238 -0.7702 0.0000 C 0 0 1 0 0 0 .. 2.6974 0.7634 0.0000 O 0 0 0 0 0 0 1 2 1 0 0 0 .. 7 10 1 0 0 0 M END
	Non-structural data	Data item	Data header	> 25 <MELTING_POINT>
			Data	179.0 - 183.0
			Blank Line	>25 <DESCRIPTION> PW(w) >25 <ALTERNATE_NAMES> 1,2 CYCLOHEXANE-DICARBOXYLIC ACID TRANS,L HEXAHYDROPHTHALIC ACID TRANS,L > 25 <DATE> 01-10-1980 > 25 <CRC_NUMBER> C-0710Dat
		Delimiter		\$\$\$\$
Compound	Molfile	Header Block		2-METHYL FURAN MACCS-II06039016302D 1 0.00186 0.00000 29
		Connection Table		6 6 0 0 0 01 V2000 0.5343 0.3006 0.0000 C 0 0 0 0 0 0 .. -2.0038 0.2857 0.0000 C 0 0 0 0 0 0 1 2 2 0 0 0 .. 5 6 2 0 0 0 M END
	Non- structural Data		Data header	> 29 <BOILING_POINT>
			Data	63.0 (737 MM) 79.0 (42 MM)
			Blank Line	> 29 <ALTERNATE_NAMES> SYLVAN > 29 <DATE> 09-23-1980 > 29 <CRC_NUMBER> F-0213
		Delimiter		\$\$\$\$

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.

Chapter 11:

RDfiles

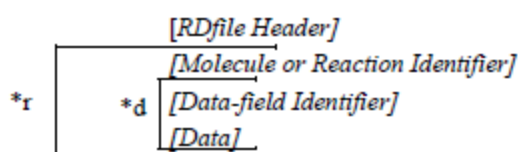
RDfile Overview

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 80.

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 interna]-regno embedded molfile
$MFMT [$MREG externa]-regno]] embedded molfile
```

```
$MIREG internal-regno
$MREG 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
- \$MREG 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
$RFMT [$RREG external-regno] embedded rxnfile
$RIREG internal-regno
$RREG 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
- \$RREG 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 $MREG 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 RFile

<i>RD Header</i>	<pre>\$RDFILE 1 \$DATM 10/17/91 10:41 \$RFMT \$RIREG 7439 \$RXN 1017911041 7439</pre>
<i>Rxn file header</i>	
<i>#Reactants/ #products</i>	2 1
<i>Molfile for 1st reactant</i>	<pre>\$MOL 10179110412D 1 0.00380 0.00000 315 4 3 0 0 0 0 0 0 0 0 . . . 1 4 1 0 0 0 4</pre>
<i>Molfile for 2nd reactant</i>	<pre>\$MOL 10179110412D 1 0.00371 0.00000 8 6 6 0 0 0 0 0 0 0 0 . . . 5 6 2 0 0 0 2</pre>
<i>Molfile for product</i>	<pre>\$MOL 10179110412D 1 0.00374 0.00000 255 9 9 0 0 0 0 0 0 0 0 . . . 6 9 2 0 0 0 2</pre>
<i>Data Block for record</i>	<pre>\$DTYPE rxn:VARIATION(1):rxnTEXT(1) \$DATUM CrCl3 \$DTYPE rxn:VARIATION(1):LITTEXT(1) \$DATUM A G Repin, Y Y Makarov-Zemlyanskii, Zur Russ Fiz-Chim, 44, p.2360, 1974 \$DTYPE rxn:VARIATION(1):CATALYST(1):REGNO \$DATUM \$MFMT \$MIREG 688 10179110412D 1 0.00371 0.00000 0 4 3 0 0 0 0 0 0 0 0 . . . 1 4 1 0 0 0 0 \$DTYPE rxn:VARIATION(1):PRODUCT(1):YIELD \$DATUM 70.0 . . . \$RFMT \$RIREG 8410 \$RXN</pre>
<i>Start of next rxn record</i>	<pre>1017911041 8410 2 1 \$MOL . . .</pre>

Chapter 12:

XDfiles

Overview

An XDfile (XML-data file) uses a standard set of XML elements that represent records of data.

The XDfile format also provides:

- Metadata or information about the origin of the data. Unlike the other CTfile formats such as the SDfile or RDfile, the XDfile enables the consumer of the data to correctly interpret it.
- The ability to handle generalized data models, such as multiple structures and nonstructure fields per record, multiple reactions per record, multiline data, and binary data. None of the other CTfile formats such as SDFiles or RDFiles have this ability.
- Very few restrictions on data formatting within the actual content. Data formatting is based on XML, which does not have restrictions on line length or blank lines. See [Data Formatting](#) on page 81.
- Fast and easy parsing by using any XML parser. The XML data can be validated by using a DTD (Document Type Definition) or an XML schema that defines primitive rules which the data must follow.

Note: Due to limitations in the capabilities of DTDs and XML schemas, an XDfile that is validated by the DTD and XML schema is not guaranteed to be correct. For example, it is possible to create a record with duplicate field values. The DTD and XML schema only provide a low level of validation. Applications which process XDfiles must provide high level checks on the consistency of the data.

- Flexibility in creating application-specific XML tags. Note, however, that it is the responsibility of the client application to interpret these custom tags.

Note: Although it is possible to convert SDfiles and RDfiles to XDfiles, there are risks involved in converting an XDfile to the original format. The backward conversion will lose the metadata which is not supported in the other CTfile formats.

See also

[Hierarchy of Elements](#) on page 83

[Alphabetical List of Elements](#) on page 83

Basic Chemistry Guide: Pipeline Pilot Chemistry Collection, chapters on *Readers* and *Writers*

Data Formatting

XML puts little or no restrictions on data formatting within the actual content. Data that is sensitive to white space, molfiles and rxnfiles in particular, must be enclosed in a CDATA section. You must set the `xml:space="preserve"` attribute to preserve. The following example is a Field element that contains a molfile:

```
<Field name="Mo1" xml:space="preserve"><![CDATA[
  SciTegic09161914072D
  1 0 0 0 0 0 0 0 0 0999 v2000
  -2.1000 -0.1208 0.0000 C 0 0 0 0 0 0 0 0 0 0
M END
]]></Field>
```

For data outside a CDATA section, certain characters must be escaped as described in the following table:

Character	Representation
<	<
>	>
"	"
'	'
&	&

For example, **mp < 100** becomes **mp < 100**.

XML Processing Instructions

The XDfile uses the following instructions as XML file headers.

XML Declaration

The first line in the XDfile specifies the XML version and the character encoding used in the document. The following example shows that the XDfile conforms to the 1.0 specification of XML and uses the UTF-8 Unicode character set.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Java-Locale

The second line in the XDfile specifies the java-locale processing instruction:

```
<?java-locale value="en_US"?>
```

The value consists of the ISO language and country codes. The language code is a lower-case, two-letter code defined by ISO-639; the country code is an upper-case, two-letter code defined by ISO-3166. Applications can read this processing instruction using an XML parser.

The

The following example shows how to get the locale value using a parser called XMLPullParser:

```
try {
    // XMLPullParser is a sample XML parser.
    XMLPullParser node = new XMLPullParser();
    // myexport_data.xml is an output XDfile from Core Interface.
    if (node.createXMLParse("c:\\home\\myexport_data.xml")) {
        int jj = node._xmlReader.next();
        jj = node._xmlReader.next();
        if (jj ==
            javax.xml.stream.XMLStreamReader.PROCESSING_INSTRUCTION) {
            // This returns "java-locale"
            String target = node._xmlReader.getPITarget();
            // This returns " value=\"en_US\""
            String data = node._xmlReader.getPIData();
        }
    }
} catch (Exception e) {
```



```

    System.out.println("Got an exception -- " + e.getMessage());
}

```

Hierarchy of Elements

The following shows the hierarchy of elements within an XDfile. The attributes of the elements are not shown in order to more clearly show the structure of the XML:

```

<XDfile>
  <Dataset>
    <Source>
      <DataSource/>
      <ProgramSource/>
      <CreatorName/>
      <CreateDate/>
      <CreateTime/>
      <Description/>
      <Copyright/>
    </Source>
    <Metadata>
      <FieldDef/>
      <ParentDef>
        <FieldDef/>
      </ParentDef>
    </Metadata>
    <Data>
      <Record>
        <Field>
        </Field>
        <Parent>
          <Record>
            <Field>
            </Field>
          </Record>
        </Parent>
      </Record>
    </Data>
  </Dataset>
</XDfile>

```

Alphabetical List of Elements

Element	Parent	Description
Copyright	Source	A copyright for the attached data
CreateDate	Source	The date the data was created
CreateTime	Source	The time the data was created
CreatorName	Source	The person who created the data
Data	Dataset	Contains records of data

Element	Parent	Description
Dataset	XDfile	Contains a collection of data
DataSource	Source	The source of the data
Description	Source	A description or comment about the data
Field	Record	The value of a field in a record
FieldDef	Metadata ParentDef	The definition of a field in a record
XDfile		The root element
Metadata	Dataset	Contains definitions of fields in the data
Parent	Record	Contains subrecords of data
ParentDef	Metadata	The definition of a parent field in the data
ProgramSource	Source	The program that created the data
Record	Data Parent	The unit of data; contains a set of field values
Source	Dataset	Contains information about the source of the data

Copyright

A copyright notice for the data.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Source](#) on page 96

Child Elements

None

Example

```
<Copyright>Copyright 2003 by Acme Corporation </Copyright>
```

CreateDate

The date the actual data was created.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
dateOrder	No	The format for the date in <CreateDate>. The separator character is specified as part of the format. The valid formats are: M-D-Y D-M-Y Y-M-D Y-D-M The default format is M-D-Y

Parent Element

[Source](#) on page 96

Child Elements

None

Example

```
<CreateDate dateOrder="M/D/Y">7/22/99</CreateDate>
```

CreateTime

The time the actual data was created.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
timeFormat	No	The format for time in < CreateTime >. The valid formats are: 24 12 The default format is 24.

Parent Element

[Source](#) on page 96

Child Elements

None

Example

```
<CreateTime>24:15</CreateTime>
```

CreatorName

The name of the person who created the data.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Source](#) on page 96

Child Elements

None

Example

```
<CreatorName>John Doe</CreatorName>
```

Data

Contains a collection of `Record` child elements that contain the actual data. Each `Record` element contains a collection of `Parent` and `Field` elements whose hierarchical structure must match the hierarchy in the `Metadata` element. The `Data` element can be empty if there is no data or only the metadata is to be transferred.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
<code>totalRecords</code>	No	The total number of root-level records in the data.

Parent Element

[Dataset](#) on page 87

Child Elements

Element	Required	Description
Record	Yes	A record of data

Example

```
<Data totalRecords="2">
  <Record>
    <Field name="Molstructure" xml:space="preserve">
      <![CDATA[7YALEn$Ak1$QPbas35quasdfsdf38...]]></Field>
    <Field name="Reaction Vessel ID">SAR6077R-01A02</Field>
    <Field name="Library ID">SAR6077R</Field>
    <Field name="Tag Id">yes</Field>
  <Parent name="XYZ Test Results" totalRecords="2">
    <Record>
      <Field name="Dose">1.0</Field>
      <Field name="Response">99.0</Field>
    </Record>
    <Record>
      <Field name="Dose">2.0</Field>
      <Field name="Response">999.0</Field>
    </Record>
  </Parent>
</Data>
```

```

</Record>
<Record>
  <Field name="Molstructure" xml:space="preserve">
    <![CDATA[7YALEn$Ak1$QPbas35quasdfsdf38...]]></Field>
  <Field name="Reaction Vessel ID">SAR6077R-01A03</Field>
  <Field name="Library ID">SAR6077R</Field>
  <Field name="Tag Id">no</Field>
</Record>
</Data>

```

Dataset

Dataset is a self-contained, single collection of data.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
name	Yes, if multiple datasets	The name of the dataset. If XDfile contains only one dataset, name is optional. If XDfile contains multiple datasets, name is required in order to identify each dataset.

Parent Element

[XDfile](#)

Child Elements

Element	Required	Description
Source	No	Describes the source of the data
Metadata	No	Describes the individual fields in the data
Data	Yes	The actual data

Example

```

<Dataset>
  <Metadata>
    <FieldDef name="CTAB" type="Structure" molFormat="Chime"/>
    <FieldDef name="Regno" type="Integer" isPrimaryKey="true"/>
    <FieldDef name="Molname" type="FixedText"/>
    <FieldDef name="Date" type="Date"/>
    <FieldDef name="Doublevalue" type="Double"/>
  </Metadata>
  <Data totalRecords="1">
    <Record>
      <Field name="CTAB">40Aieg6Mprlczdb^fg37JiOh</Field>
      <Field name="Regno">1</Field>
      <Field name="Molname">Test Structure</Field>
      <Field name="Date">05/10/2002</Field>
    </Record>
  </Data>
</Dataset>

```

See also

[Hierarchy of Elements](#)

[Alphabetical List of Elements](#)

DataSource

The source of the data.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Source](#)

Child Elements

None

Example

```
<DataSource>ACD99.1 Hview</DataSource>
```

Description

A description or comment about the data.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Source](#) on page 96

Child Elements

None

Example

```
<Description>This is test data.</Description>
```

Field

Contains data for a single field in a record. Note that the molFormat, molVersion, and rxnFormat attributes can be used on individual Field elements to override the format specified in the Metadata element.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
name	Yes	The name of the parent field.
molFormat	No	The format for the structure, if the field contains structures. This value overrides the molFormat specified in the Metadata element. The valid values are: Molfile Chime Smiles The default value is Molfile.
rxnFormat	No	The format for the reaction, if the field contains reactions. This value overrides the rxnFormat specified in the Metadata element. The valid values are: Rxnfile Chime The default value is Rxnfile.
molVersion	No	The version of the data format. molVersion is currently only applicable to structures. The absence of this attribute implies a pre-V2000 molfile. This value overrides the molVersion specified in the Metadata element. The valid values are: V2000 V3000

Parent Element

[Record](#) on page 95

Child Elements

None

Example

```
<Field name="Dose">1.0</Field>
```

FieldDef

Contains information about a single field within the data.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
name	Yes	The name of the parent field. This is a simple name because the context of the field in the hierarchy is implied by the XML structure.
type	Yes	The type of data in the field. Applications can define their own type, but it will be the responsibility of client applications to interpret these. The valid values are:

Attribute	Required	Description
		Reaction Structure Integer Double FixedText VariableText Date Time DateTime Binary
isKey	No	true if the field is a key. The valid values are true or false. The default value is false.
isPrimaryKey	No	true if the field is a primary key. The valid values are true or false. The default value is false.
nativeName	No	The name of the field specific to the source from which it is derived. For example, a native name for an Oracle database is <code>schema.table.column</code> .
encoding	No	The encoding set used by the data. Sample values are: binhex rot64
charset	No	The character set used by the text. A sample value is ISO-8859-1.
decimalSeparator	No	The decimal separator used for floating point numbers. The valid values are: Period. Comma The default value is Period .
maxLength	No	The maximum length of the data in this field. For binary fields in a relational data source (such as CLOB and BLOB fields), the maximum length varies depending on the version of the Oracle JDBC driver.
molFormat	No	The format for the structure, if the field contains structures. The valid values are: Molfile Chime Smiles The default value is Molfile.
rxnFormat	No	The format for the reaction, if the field contains reactions. The valid values are: Rxnfile Chime The default value is Rxnfile.

Attribute	Required	Description
molVersion	No	The version of the data format. <code>molVersion</code> is currently only applicable to structures. The absence of this attribute implies a pre-V2000 molfile. The valid values are: V2000 V3000
dateOrder	No	The order and format of a date field. The value specifies the separator character. Dates must be specified using numbers only; do not use month names. The valid orders are: M-D-Y D-M-Y Y-M-D Y-D-M The default value is M-D-Y.
timeOrder	No	The order of a date field. The value specifies the separator character. The valid orders are: h:m m:h The default value is h:m.
timeFormat	No	The format of a time field. The valid values are: 24 12 The default value is 24.
precision	No	The precision for a real number field. This is the total number of digits.
scale	No	The scale for a real number field. This is the number of digits to the right of the decimal point.
nullsAllowed	No	<code>true</code> if the field can contain nulls. The valid values are <code>true</code> or <code>false</code> . The default value is <code>false</code> .
isIndexed	No	<code>true</code> if the field is indexed. The valid values are <code>true</code> or <code>false</code> . The default value is <code>false</code> .
isHidden	No	<code>true</code> if the field is hidden. The valid values are <code>true</code> or <code>false</code> . The default value is <code>false</code> .
units	No	The units of measurement associated with the field.
javaFormat	No	The Java format to use when the field value is parsed. This is the Java output format used by the <code>NumberFormat</code> or <code>DateFormat</code> class.

Parent Element

[Metadata](#) on page 95

[ParentDef](#) on page 92

Child Elements

None

Example

```
<FieldDef name="Molstructure"
  type="Structure"
  molFormat="Chime"/>
<FieldDef name="Reaction Vessel ID"
  type="FixedText"
  maxLength="15" />
```

Parent

Contains child records of data. A `Parent` can contain other `Parent` elements, thus, creating a sub-hierarchy of data.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
name	Yes	The name of the field.
totalRecords	No	The total number of root-level records in the data.

Parent Element

Record

Child Elements

Element	Required	Description
Record	Yes	A record of data

Example

```
<Parent name="XYZ Test Results" ID="100">
  <Record>
    <Field name="Dose">1.0</Field>
    <Field name="Response">99.0</Field>
  </Record>
  <Record>
    <Field name="Dose">2.0</Field>
    <Field name="Response">999.0</Field>
  </Record>
</Parent>
```

See also

[Hierarchy of Elements](#)

[Alphabetical List of Elements](#)

ParentDef

Contains information about a single parent field within the data.

Do not create unnecessary parent fields at the root. Parent fields should only be used when there is an actual hierarchical structure to the data. For example, an SDFfile would not have `ParentDef` elements, but would only have multiple `FieldDef` elements directly under the `Metadata` element. The actual data in the SDFfile follows this format by having multiple `Record` elements directly under the `Data` element.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
name	Yes	The name of the parent field. This is a simple name because the context of the parent in the hierarchy is implied by the XML structure.

Parent Element

[Metadata](#)

ParentDef

Child Elements

Element	Required	Description
FieldDef	Yes	Contains information about a single field within the data
ParentDef	No	Contains information about a single parent within the data. A parent within a parent creates a hierarchy.

Example

```
<?xml version="1.0"?>
<XDfile version="1.2" xmlns="http://www.mdl.com/XDfile/NS">
  <Dataset
    name="MySource">
    <Source>
      <DataSource>ACD99.1 Hview</DataSource>
      <ProgramSource>GCS 1.0</ProgramSource>
      <CreatorName>John Doe</CreatorName>
      <CreateDate dateOrder="M/D/Y">7/22/99</CreateDate>
      <CreateTime timeFormat="24">13:45</CreateTime>
    </Source>
    <Metadata>
      <FieldDef name="Molstructure" type="Structure"
        molFormat="Chime"/>
      <FieldDef name="Reaction Vessel ID" type="FixedText"
        maxLength="15"/>
      <FieldDef name="Library ID" type="FixedText"
        maxLength="8"/>
      <FieldDef name="Tag Id" type="fixedText"
        maxLength="3"/>
      <ParentDef name="XYZ Test Results">
        <FieldDef name="Dose" type="Double"/>
        <FieldDef name="Response" type="Double"/>
      </ParentDef>
    </Metadata>
  </Dataset>
</XDfile>
```

```
<Data>
  <Record>
    <Field name="Molstructure" xml:space="preserve">
      <![CDATA[7YALen$Ak1$QPbas35quasdfsdf38...]]></Field>
    <Field name="Reaction Vessel ID">SAR6077R-01A02</Field>
    <Field name="Library ID">SAR6077R</Field>
    <Field name="Tag Id">yes</Field>
    <Parent name="XYZ Test Results" totalRecords="2">
      <Record>
        <Field name="Dose">1.0</Field>
        <Field name="Response">99.0</Field>
      </Record>
      <Record>
        <Field name="Dose">2.0</Field>
        <Field name="Response">999.0</Field>
      </Record>
    </Parent>
  </Record>
  <Record>
    <Field name="Molstructure" xml:space="preserve">
      <![CDATA[7YALen$Ak1$QPbas35quasdfsdf38...]]></Field>
    <Field name="Reaction Vessel ID">SAR6077R-01A03</Field>
    <Field name="Library ID">SAR6077R</Field>
    <Field name="Tag Id">no</Field>
  </Record>
</Data>
</Dataset>
</XDfile>
```

See also

[Hierarchy of Elements](#)

[Alphabetical List of Elements](#)

ProgramSource

The program that created the data.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Source](#) on page 96

Child Elements

None

Example

```
<ProgramSource>MDL Core Interface 1.0</ProgramSource>
```

Metadata

Contains information that describes the fields in the data. Applications can use this information to allow automatic creation of database tables or better formatting of data. The hierarchical structure of the Metadata element must match the data model contained in the Data element.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Dataset](#) on page 87

Child Elements

Element	Required	Description
ParentDef	No	Contains information about a single parent within the data
FieldDef	No	Contains information about a single field within the data

Example

```
<Metadata>
  <FieldDef name="Molstructure"
    type="Structure"
    molFormat="Chime"/>
  <FieldDef name="Reaction Vessel ID"
    type="FixedText"
    maxLength="15"/>
  <FieldDef name="Library ID"
    type="FixedText"
    maxLength="8"/>
  <FieldDef name="Tag Id"
    type="FixedText"/>
  <ParentDef name="XYZ Test Results">
    <FieldDef name="Dose"
      type="Double"/>
    <FieldDef name="Response"
      type="Double"/>
  </ParentDef>
</Metadata>
```

Record

A record of data, and contains a set of Field elements.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Data](#) on page 86

[Parent](#) on page 92

Child Elements

Element	Required	Description
Field	No	A single field of data
Parent	No	Contains subrecords of data. A parent within a record creates a hierarchy.

Example

```
<Record>
  <Field name="Dose">1.0</Field>
  <Field name="Response">99.0</Field>
</Record>
```

Source

Contains information that describes the source of the data. Note that although each child element is optional, the order of the child elements is fixed. If a child element does not follow the order specified below, the element is discarded.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

None

Parent Element

[Dataset](#) on page 87

Child Elements

Element	Required	Description
DataSource	No	The source of the data
ProgramSource	No	The program that created the data
CreatorName	No	The person who created the data
CreateDate	No	The date the data was created. The default format is M-D-Y.
CreateTime	No	The time the data was created. The default format is 24-hour
Description	No	A description or comments about the data
Copyright	No	A copyright notice for the data

Example

```
<Source>
  <DataSource>ACD99.1 Hview</DataSource>
  <ProgramSource>Core Interface 1.1</ProgramSource>
  <CreatorName>John Doe</CreatorName>
  <CreateDate dateOrder="M/D/Y">7/22/99</CreateDate>
  <CreateTime timeFormat="24">13:45</CreateTime>
</Source>
```

XDfile

`XDfile` is the root element. XML data that uses the XDfile format must begin with `<XDfile>` and end with `</XDfile>`. It contains one or more `Dataset` elements.

See also [Alphabetical List of Elements](#) on page 83.

Attributes

The data type of all attributes is CDATA.

Attribute	Required	Description
<code>xmlns</code>	Yes	The default namespace for elements in the <code>XDfile</code> . The default is <code>http://www.mdl.com/XDfile/NS</code>
<code>version</code>	Yes	The version number of the XDfile data.

Parent Element

None

Child Elements

Element	Required	Description
Dataset	Yes	A single collection of data

Example

```
<?xml version="1.0"?>
<XDfile version="1.2" xmlns="http://www.mdl.com/XDfile/NS">
  <Dataset
    name="MySource">
    <Source>
      <DataSource>ACD99.1 Hview</DataSource>
      <ProgramSource>GCS 1.0</ProgramSource>
      <CreatorName>John Doe</CreatorName>
      <CreateDate dateOrder="M/D/Y">7/22/99</CreateDate>
      <CreateTime timeFormat="24">13:45</CreateTime>
    </Source>
    <Metadata>
      <FieldDef name="Molstructure" type="Structure"
        molFormat="Chime"/>
      <FieldDef name="Reaction Vessel ID" type="FixedText"
        maxLength="15"/>
      <FieldDef name="Library ID" type="FixedText"
        maxLength="8"/>
      <FieldDef name="Tag Id" type="fixedText"
        maxLength="3"/>
      <ParentDef name="XYZ Test Results">
        <FieldDef name="Dose" type="Double"/>
        <FieldDef name="Response" type="Double"/>
      </ParentDef>
    </Metadata>
    <Data>
      <Record>
```

```
<Field name="Molstructure" xml:space="preserve">
  <![CDATA[7YALen$Ak1$QPbas35quasdfsdf38...]]></Field>
<Field name="Reaction Vessel ID">SAR6077R-01A02</Field>
<Field name="Library ID">SAR6077R</Field>
<Field name="Tag Id">yes</Field>
<Parent name="XYZ Test Results" totalRecords="2">
  <Record>
    <Field name="Dose">1.0</Field>
    <Field name="Response">99.0</Field>
  </Record>
  <Record>
    <Field name="Dose">2.0</Field>
    <Field name="Response">999.0</Field>
  </Record>
</Parent>
</Record>
<Record>
  <Field name="Molstructure" xml:space="preserve">
    <![CDATA[7YALen$Ak1$QPbas35quasdfsdf38...]]></Field>
  <Field name="Reaction Vessel ID">SAR6077R-01A03</Field>
  <Field name="Library ID">SAR6077R</Field>
  <Field name="Tag Id">no</Field>
</Record>
</Data>
</Dataset>
</XDfile>
```


Appendix A: Stereo Notes

Parity values can appear in the atom blocks of CTfiles. See CFG (for stereo configuration) and sss (atom stereo parity) in the table for [Atom Block](#) on page 7.

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 Figure 1, atoms 1, 2, and 4 are all in the plane of the paper, and atom 3 is above the plane.

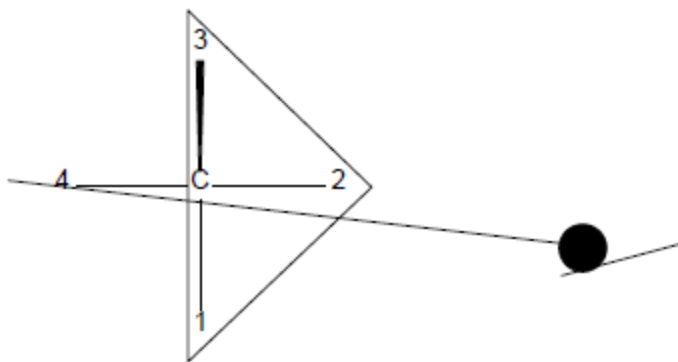


Figure 1

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.

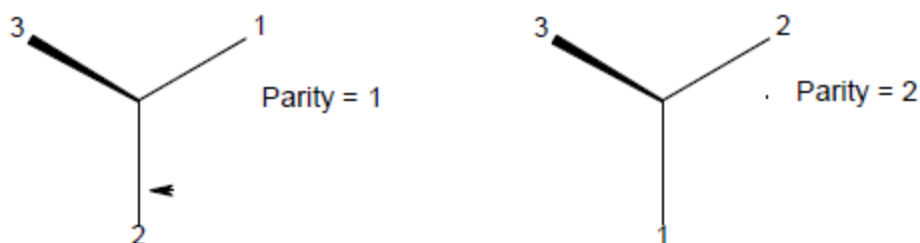


Figure 2

Note: 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 in Figure 2 has a parity value of 2.

For additional information about BIOVIA stereochemistry, see *BIOVIA Chemical Representation*, which has an appendix entitled 'Representation of Stereochemistry in BIOVIA Databases.'