Draft Specification for Revised Version of the Standard Molecular Data (SMD) Format

JOHN M. BARNARD‡

46 Uppergate Road, Stannington, Sheffield S6 6BX, England

Received December 5, 1989

The formal draft specification for the revised version of the Standard Molecular Data Format is presented.

INTRODUCTION

A paper describing the Standard Molecular Data Format, as developed by a group of European chemical and pharmaceutical companies, was published in 1989.[1] Copies of a technical description of the most recent version of it (version 4.3), dated February 4, 1987, have been distributed by the authors of that paper.[2]

An open meeting held in Frankfurt, FRG, in May 1988 discussed this version of the Format and identified several areas in which it could usefully be extended or improved. Under the auspices of the Chemical Structure Association, a series of Technical Working Groups was established to examine the Format in detail and to make recommendations for a revised version. Some preliminary accounts of the conclusions of these groups have appeared elsewhere.[3-5] Questions of standardization of chemical structure information, and the need for a suitable standard format for information exchange, are addressed in several contributions to a recent symposium.[6]

This paper presents the conclusions of the Working Groups in the form of a draft specification for version 5.0 of SMD Format. Comments are invited on the draft and should be sent to the Technical Secretary at the address shown above. In the light of any comments received, further revisions may be made before a definitive specification for the Format is presented at an open meeting to be held in association with the 2nd International Meeting on Chemical Structures at the Leeuwenhorst Congress Center, Noordwijkerhout, The Netherlands, June 3-7, 1990.

SMD Format is not intended as a universal database format, still less as an internal format for different systems. Rather, it is intended as an interchange format allowing different programs to exchange data with the minimum need for conversion routines.

Efforts are currently under way to form a new organization that will control the future development of the Format; it will also coordinate efforts to obtain recognition and approval for the Format from appropriate international authorities. Membership of the new body will be open to interested organizations and individuals.[7]

MEMBERSHIP OF TECHNICAL WORKING GROUPS

Five Technical Working Groups have so far been involved in considering different aspects of the Format, under the headings General Principles, Generic Structures and Query Structures, Reactions, Stereochemistry, and Layout Features.

The following individuals have participated in all or some of the meetings of these groups (Affiliations shown are those that were current at the time the person concerned attended Working Group meetings.): J. M. Barnard (Barnard Chemical Information Ltd., Sheffield) (Technical Secretary), H. Braun (F Hoffmann-La Roche AG, Basel), J. Buckingham (Chapman and Hall, London), C. Buse (Sandoz AG, Basel), B. Carrabin [Polygen (Europe) Ltd., Paris], A. P. F. Cook (Orac Ltd., Leeds), J. Dill (Molecular Design MDL AG, Basel), U. Hegi (Sandoz AG, Basel), U. Heigl (Molecular Design MDL AG, Basel), M. Hicks (Beilstein Institute, Frankfurt), K. Higgins (Orac Ltd., Leeds), P. Hoever (Bayer AG, Leverkusen), P. Huguet (Télésystèmes Questel, Paris), S. Hull [Fraser Williams (Scientific Systems) Ltd., Macclesfield], H. Jacob (Ciba-Geigy AG, Basel), A. Kos (Molecular Design MDL AG, Basel), Herr Krause (Information und Kommunikation, Freiburg), J. Mockus (Chemical Abstracts Service, Columbus), M. Ott (CAOS/CAMM Center, University of Nijmegen), Herr Posselt (Information und Kommunikation, Freiburg), B. Rhode (Ciba-Geigy AG, Basel), J. Römelt (Bayer AG, Leverkusen), G. Russell (Hampden Data Services Ltd., Nottingham), H. Saller (Chemodata Computer-Chemie GmbH, Gröbenzell), R. Schenck (Chemical Abstracts Service, Columbus), W. Sieber (Sandoz AG, Basel), R. Steppuhn (Schering AG, Berlin), J. Theodosiou (Molecular Design MDL AG, Basel), W. G. Town (Hampden Data Services Ltd., Abingdon), D. Watson (Cambridge Crystallographic Data Center, Cambridge), S. Welford (Beilstein Institute, Frankfurt), P. Youkaribache [Polygen (Europe) Ltd., Paris], C. Zirz (Bayer AG, Leverkusen).

REVISIONS MADE TO THE FORMAT

Though the revised format is based on version 4.3 of SMD Format, the problems and limitations of that version are such that it was decided not to attempt to retain backward compatibility with it. Many aspects of its basic philosophy, and some of its notational conventions, are, however, retained. In particular, the idea of a block structure, in which different types of information are given in different blocks, which can be skipped if desired, remains a central feature of the new version. The idea has in fact been extended, with the consequence that some of the blocks in version 4.3 (in particular the >CT block) have been subdivided into several different blocks in version 5.0. This allows better representation of charges, radical states, isotopes, and stereochemistry.

Other revisions include the ability to represent a variety of different bond conventions and more satisfactory handling of stereochemistry, "shortcut" superatoms, subconnection tables, and reaction descriptions. Better distinction is made between information about molecules themselves and information about how to display them. Extensions have been made to allow the representation of atom correspondences in reactions, generic structures, and chemical structure queries.

The FORTRAN bias of the original has been removed, and the specification of block length in the header line for each block has been replaced by the use of explicit terminator lines. Some adjustments have been made to the special characters used to distinguish between different types of line to make it easier to transfer SMD files between different character coding systems (e.g., ASCII and EBCDIC).

The revised version of the Format has been designed in such a way as to permit further extension while retaining backward compatibility. Among areas that the Format proposed here is unable to handle (or cannot handle fully) are inorganic and

‡Technical Secretary of the Technical Working Groups of the SMD Format Subgroup of the Chemical Structure Association.

0095-2338/90/1630-0081$02.50/0

organometallic compounds, π complexes, salts, biomacromolecules, polymers and polypeptides, and ancillary data (including text descriptors, property data, and fragment-, atom-, and bond-centered values). Preliminary ideas have already been developed by the Working Groups for some of these, though they are not included in the present draft specification. It is planned to convene meetings of additional Technical Working Groups to examine these problem areas and to propose further extensions to the Format. Particular attention will need to be given to the relationship between SMD Format and other standard formats which have been proposed, or are in use, for some of these problem areas.[8-11] In addition, it may be desirable to develop a compressed version of the Format for the representation of large amounts of data.

FORMAL SPECIFICATION OF REVISED FORMAT

Most of the remainder of this paper consists of a formal description of version 5.0 of SMD Format, as proposed by the Technical Working Groups. The bulk of the definition is contained in the set of 72 syntax diagrams and the notes associated with each. Some introductory paragraphs describe the general organization of lines in SMD files. Following the formal description, three examples of structures represented in the Format are given.

GENERAL ORGANIZATION OF SMD FILES

SMD files are ordinary line-structured text files. Each line can contain a maximum of 80 characters and is terminated by an end-of-line character. The lines within a file are organized in an hierarchical manner, as follows: **scope**, a group of sections, which are able to reference each other (for example, a reaction section referring to molecule sections); **section**, the description of a chemically relevant unit, such as a molecule or a reaction; **block**, a collection of data giving a particular kind of information (e.g., the nodes contained in a molecule; **subblock**, subordinate level of organization within a block. The divisions between these levels of organization are indicated by special header and terminator lines.

LINE TYPES AND TAG CHARACTERS

Nine different types of line can occur in SMD files; they are distinguished by the first character in the line, as follows:

"$" header or terminator line for a scope
"/" header or terminator line for a section
">" header or terminator line for a block
")" header or terminator line for a subblock
" " data line
"@" audit data line
"+" continuation line
"!" comment line
"#" filename specification line for "include" file

These tag characters allow irrelevant lines to be skipped during reading.

Header and Terminator Lines. The exact format for these is described in the syntax diagrams. Terminator lines always consist of the relevant tag character followed by "END".

Continuation Lines. These may be used where the previous line is longer than the permitted maximum of 80 characters, though the syntax minimizes the need for such continuation lines. The characters of the continuation line are regarded as being appended directly to the end of the previous line, any trailing spaces on the previous line being ignored. It is therefore possible to split long lines at any point, including in the middle of words.

Comment Lines. These can be inserted at any point and can contain any character string.

Table I. Alphabetical Cross-Reference Listing of the Syntax Diagrams

30	3D coordinates block	49	include file specification
40	alternative number	4	information section
48	atom change subblock	71	integer
10	atom subblock	70	integer set
13	atom symbol	25	isotope block
14	atom symbol set	45	mapping subblock
31	attachment block	58	minute
65	attachment number	8	molecule section
50	audit	54	month
51	audit data	9	node block
47	bond change subblock	37	node data picture block
38	bond data picture block	60	node identifier
33	bond environment block	64	node number
18	bond identifier	35	node picture block
36	bond picture block	63	node range
17	bond subblock	61	node set
22	charge subblock	43	path block
23	charge value	68	quoted string
28	CIP atom block	21	radical state
29	CIP bond block	20	radical subblock
62	component node set	41	reaction section
66	component number	72	real number
44	component subblock	32	ring connectivity block
5	content section	3	scope
15	convention block	59	second
16	convention identifier	46	site subblock
52	date	1	SMD file
55	day	26	stereo block
6	default section	27	stereo relationship
34	display coordinates block	67	string
7	fragment section	42	substep block
11	fragment subblock	56	time
2	global	24	valency block
57	hour	39	variable section
19	hydrogen count subblock	12	variable subblock
69	identifier	53	year

Include Files. These can be used to allow parts of an SMD file (for example, a set of standard shortcut fragments) to be held in a separate file. The format for the file specification line is shown in the relevant syntax diagram. Include file specification lines can occur at any point in the file, and the lines of the included file are treated exactly as if they had occurred in the main file at that point. Include files may be nested, and there is no requirement for the lines within any one included file to comprise any particular syntactic unit.

Audit Data Lines. Audit data are data about the origin of the SMD file (the program that wrote it, the date written, etc.). Because different parts of the file may be written at different times, or by different programs (especially, for example, when the data originally in the file are enhanced, perhaps to deduce CIP descriptors from another stereochemical description), audit data may optionally be included at any level of organization within the file.

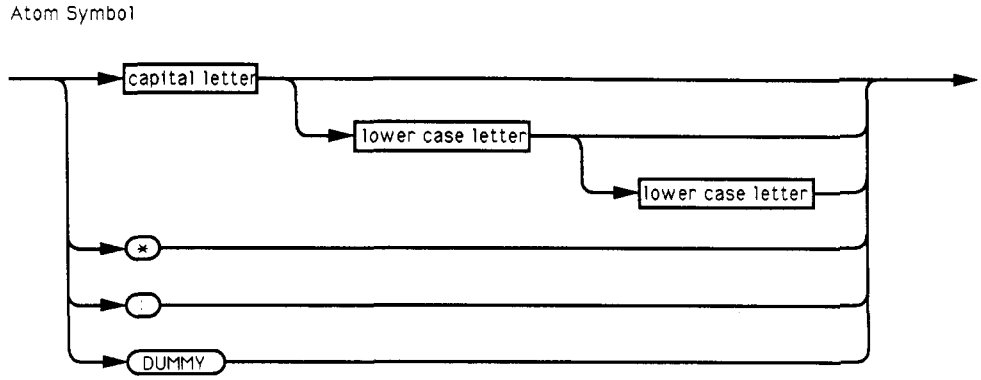
SMD SYNTAX

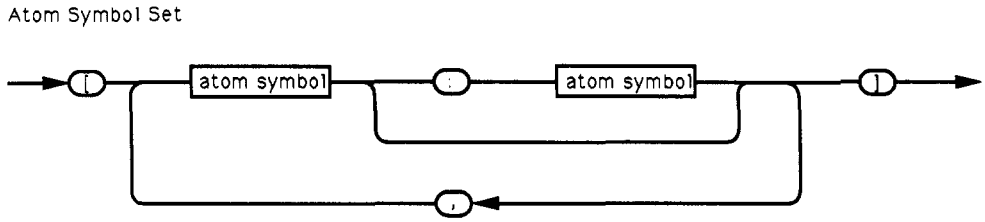
The syntax diagrams show the sequences of symbols that are permitted in SMD files. Within each diagram, words or symbols shown in ovals must be included exactly as they stand. The symbol "eol" enclosed in an oval is the end-of-line marker, and an empty oval indicates a space. Words in rectangular boxes are the names of other syntax diagrams, the contents of which are to be included at that point.

The arrows indicate the allowable sequences of symbols. In addition to those spaces explicitly indicated by empty ovals in the diagrams, one or more spaces must be used to separate every adjacent pair of words or symbols, with the exception that spaces are not required next to the following symbols:

eol [ ] , : . + - @ '

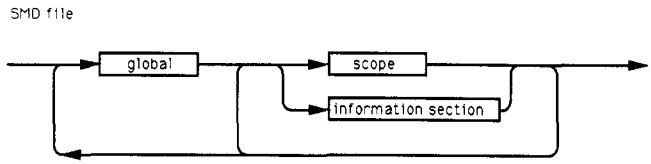
In addition, spaces are not required between the characters





of a string and must not occur between the digits of integers or real numbers, within dates and times, or between the letters, digits, and "_" symbols of identifiers.

(1) SMD File. This shows the overall composition of an SMD file. Global definitions are included in the initial global

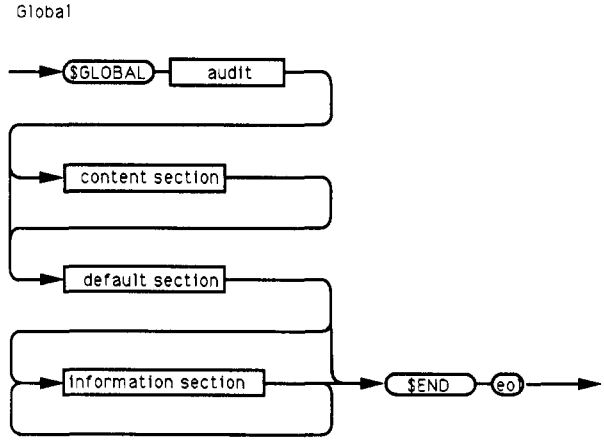


scope and may be referred to throughout the file. In the remainder of the file, only sections that occur within the same scope are able to refer to each other (for example, a /REACTION section referring to /MOLECULE sections for the participants in a reaction).

It is possible to have "unscoped" information sections, though such sections cannot refer to each other (though they can refer to sections defined in the global scope). Unscoped sections will normally be used where only a single section appears or perhaps for simple sets of molecules. The facility for allowing unscoped sections avoids the need to enclose all sections in otherwise redundant $SCOPE and $END lines.

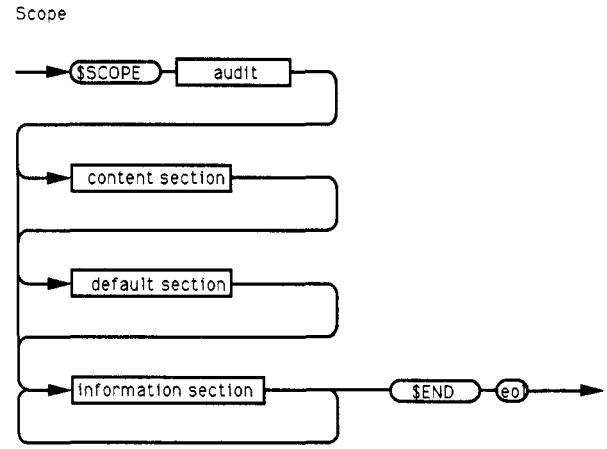
The entire syntax for an SMD file may be repeated, allowing SMD files to be appended without problems.

(2) Global. The definitions in a global scope apply throughout the file (unless superseded by another global scope).

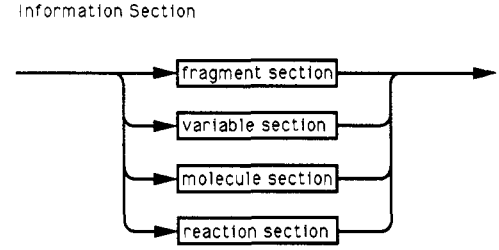


Global scopes must contain at least a /DEFAULT section, but need not contain any information sections.

(3) Scope. The information sections in a scope may refer to each other, as well as to the sections in the applicable global scope.



(4) Information Section. Sections containing actual chemical information may occur in any order, though any given

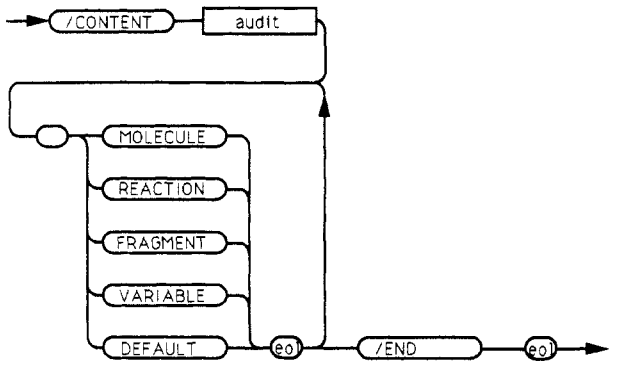


section within a scope is only able to refer to those sections that precede it.

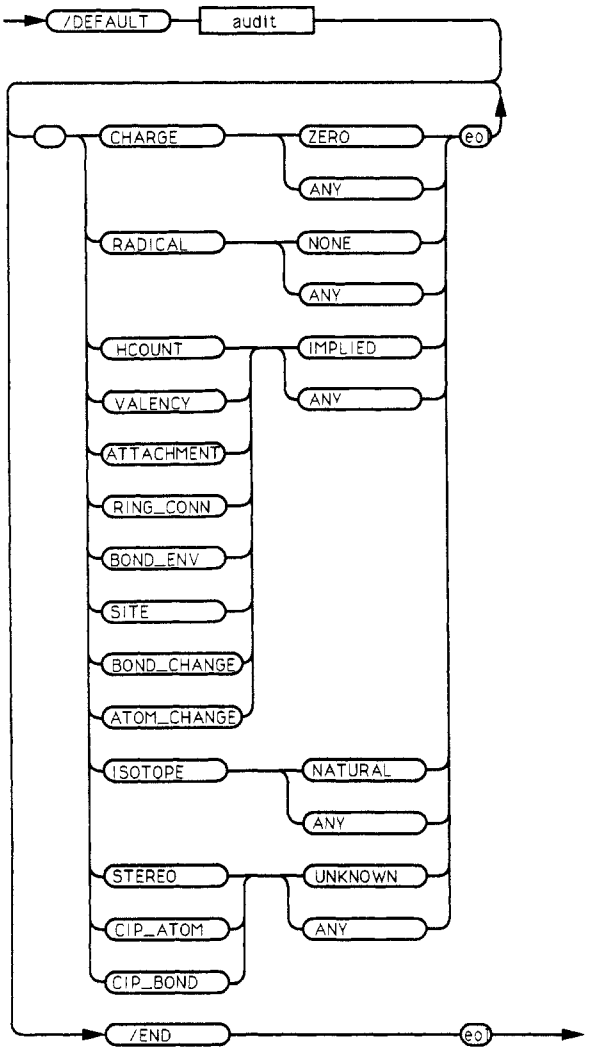
(5) Content Section. This lists the types of sections that are present in the current $SCOPE and, if present, must be the first section in the $SCOPE. It can be used to allow skipping of $SCOPEs that either do not contain sections of interest (e.g., /REACTIONS) or alternatively do contain sections that cannot be processed.

(6) Default Section. This is used to define the defaults to be assumed for missing data. The /DEFAULT section must

Content Section



Default Section



occur at the start of a $SCOPE, immediately following the /CONTENT section (if any).

The defaults that the section defines apply throughout the $SCOPE, though, of course, the defaults may always be overridden by explicitly given data. The /DEFAULT section in a $GLOBAL scope applies throughout the file (or until another $GLOBAL scope), though defaults may be redefined within any particular $SCOPE.

It is not compulsory to specify defaults for all (or even for any) of the possible types of data in a /DEFAULT section. Any that are not specified are reset to the "system default", which is the first-listed keyword in the syntax diagram in each case (i.e., ZERO, NONE, IMPLIED, NATURAL, and UNKNOWN). Thus, a valid /DEFAULT section can consist of

/DEFAULT
/END

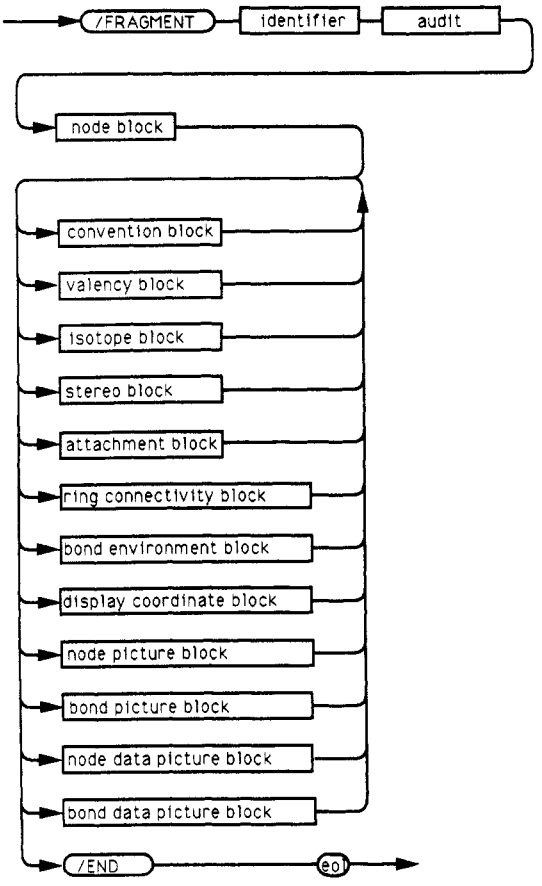
and has the effect of resetting all defaults to the system default.

The keyword IMPLIED, which is used for several defaults, indicates that the default values are the values implied by other data explicitly given. It should be noted that some of these values imply each other. The values given in a >VALENCY block, when taken with the connections shown in the)BOND subblock, imply the values in the)HCOUNT subblock and vice versa. If neither a >VALENCY block nor an)HCOUNT subblock is present, no default values can be assumed for either.

Default values other than the system defaults will normally only be required for the representation of query structures, where the keyword ANY in each case will give the most general possible query.

(7) Fragment Section. A fragment is a portion of a molecule, which is described separately, and is equivalent to the "Superatom" idea in version 4.3 of the Format. /FRAGMENT sections can be used to define shortcuts for commonly

Fragment Section



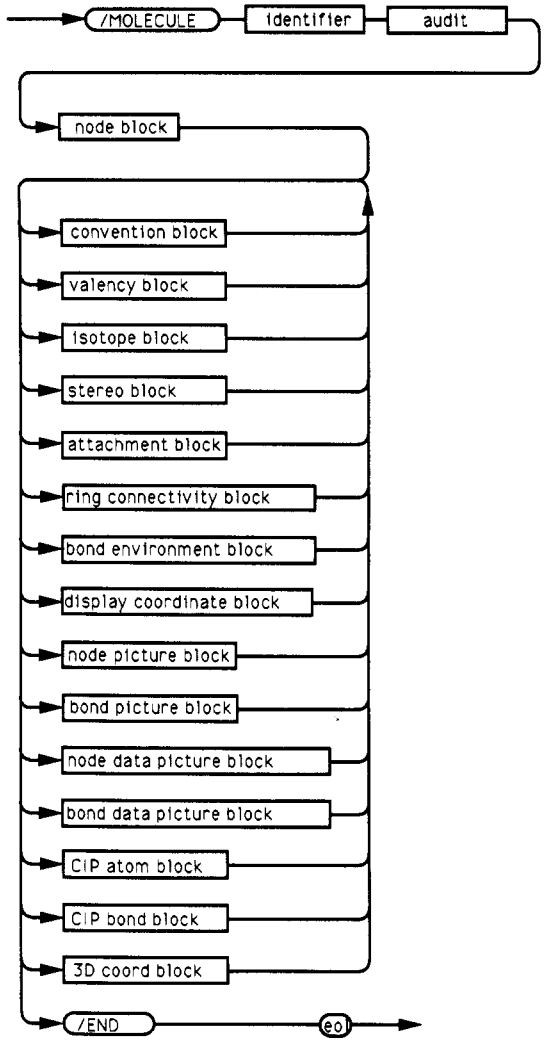
occurring groups and can be referred to by /MOLECULE sections and other /FRAGMENT sections. /FRAGMENT sections can be nested to any level. /FRAGMENT sections are also used to define the alternative values for variables.

A /FRAGMENT section can contain any of the same blocks as a /MOLECULE section, except for those blocks that are dependent upon the full structure of a molecule (>CIP_ATOM, >CIP_BOND and >COORD).

A /FRAGMENT section is referenced by its identifier, which must be unique within a $SCOPE.

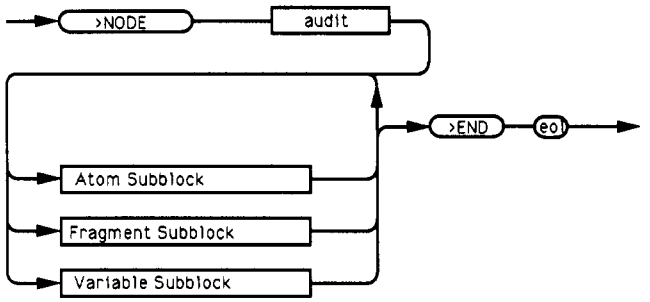
(8) Molecule Section. This describes a single molecule. It is referenced by its identifier, which must be unique within a $SCOPE.

Molecule section



(9) Node Block. This is compulsory in /FRAGMENT and /MOLECULE sections and must be the first block in the section. It lists the nodes present in the fragment or molecule

Node Block

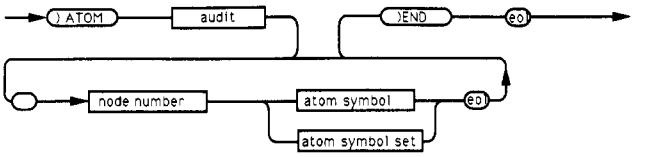


and is subdivided into three types of subblock corresponding to different types of node, each of which can occur any number of times.

The node numbers specified within a >NODE block are used to identify particular nodes. They must be unique throughout all the subblocks within a >NODE block and must form a continuous sequence starting at 1. However, they need not occur sequentially.

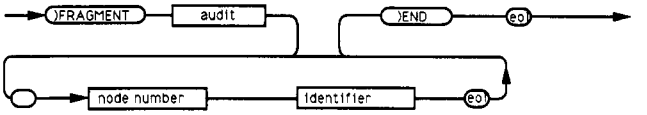
(10) Atom Subblock. This is used for nodes that are atoms; either a single atom symbol or a set of atom symbols where the atom type is variable is specified. Terminal hydrogens may be omitted, unless it is necessary to refer to them in another block (e.g., a >STEREO or >ISOTOPE block).

Atom Subblock



(11) Fragment Subblock. This is used for nodes that are fragments; an identifier is specified which refers to the

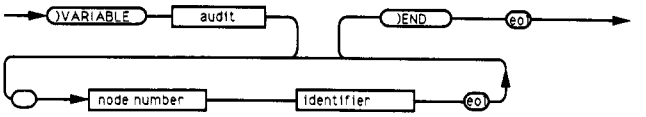
Fragment Subblock



/FRAGMENT section in question, which must occur earlier in the same scope or in the currently effective global scope.

(12) Variable Subblock. This is used for nodes that are variables; an identifier is specified which refers to the

Variable subblock



/VARIABLE section in question, which must occur earlier in the same scope or in the currently effective global scope.

(13) Atom Symbol. Any of the valid element symbols specified in the latest IUPAC recommendations[12] are permitted, including the three-letter symbols for elements beyond atomic number 103.[13] H should be used for explicitly cited hydrogen and its isotopes (which should be distinguished by use of the >ISOTOPE block).

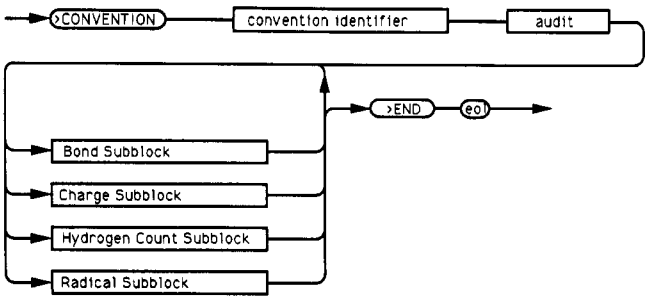
In addition, the symbol "*" can be used to show an attachment point in a fragment, the symbol ":" for an explicitly cited lone pair of electrons (which may be needed for reference in a >STEREO block), and the keyword DUMMY for special dummy nodes (which may be used for display and other purposes).

Special "generic element" symbols (e.g., Q, X, G1, etc.) are not permitted in the standard: atom symbol sets should be used instead.

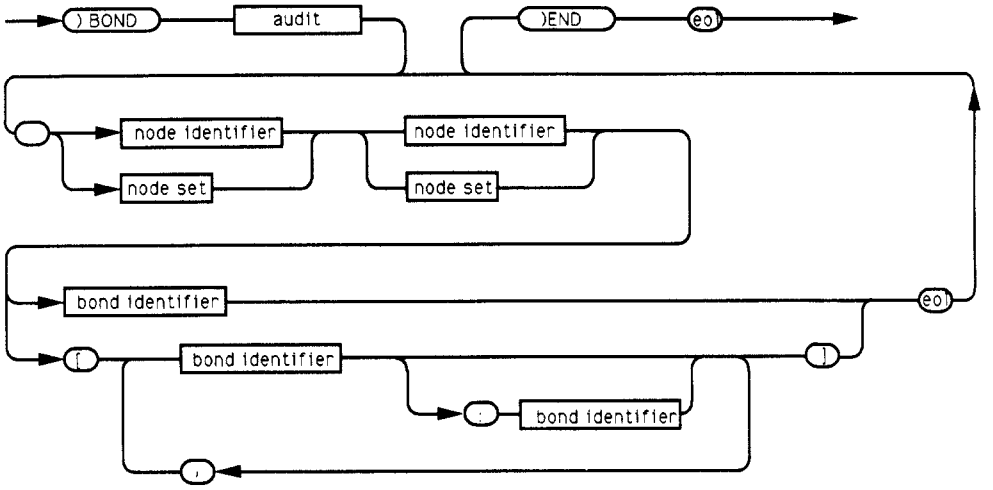
(14) Atom Symbol Set. This represents a set of atom symbols, which may be used for alternative atom types. The colon is a range constructor, and the comma separates items in the set. Only the valid IUPAC element symbols are permitted in ranges (though the special atom types may be included as individual items in the set). Within ranges, the first atom symbol must have a lower atomic number than the second, and the range implies inclusion within the set of all atom symbols from the first to the second inclusive, the implied ordering being by atomic number.

(15) Convention Block. A number of different bond representation conventions (for features such as aromaticity and

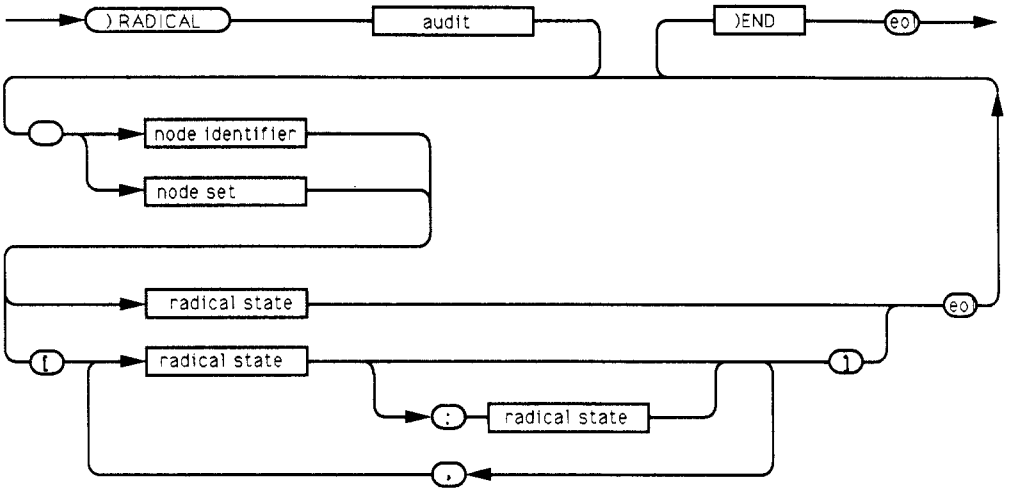
Convention Block



Bond Subblock



Radical Subblock



tautomerism) are commonly used in chemical information systems. SMD Format allows several different such conventions to be used within a /MOLECULE or /FRAGMENT section, each in its own >CONVENTION block. The >CONVENTION block includes subblocks for different types of data (bond orders, formal charges, hydrogen counts, and radical states) which may vary under different conventions.

(16) Convention Identifier. The convention being used is indicated by an identifier in the >CONVENTION block

Convention Identifier

header; for each convention only certain bond order identifiers may be used. Initially, two conventions will be permitted in the standard: The **Simple Convention** allows only single, double, and triple bonds, with no normalization of alternating bond, aromatic, or tautomeric systems.

convention identifier: SIMPLE
bond identifiers: SIN, DOU, TRI

The **Chemical Abstracts Registry System III Convention** also allows a normalized bond type, representing bonds that can be normalized by the procedures of Mockus and Stobaugh.[14]

convention identifier: CASREG3
bond identifiers: SIN, NOR, DOU, TRI

Additional conventions may be added to the standard in future revisions. Users may also define their own "private" conventions, using appropriate identifiers, but these will not be included in the standard unless the definitions of the various bond types (and any necessary normalization or aromaticity identification, etc., algorithms) are published in the open literature.

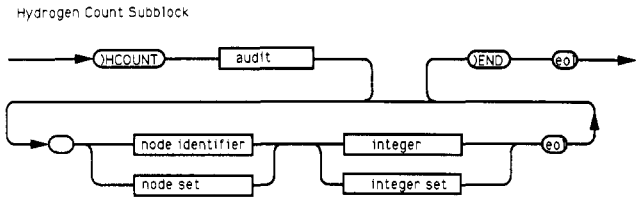
The >BOND_PIC block can be used to specify how bonds are to be drawn.

(17) Bond Subblock. The bond is described by the nodes that it connects. Sets of node numbers may be used to indicate variable-position bonds and sets of bond orders to indicate variable bond orders. In the construction of ranges of bond orders, the ordering is that specified in the definition of the relevant convention.

(18) Bond Identifier. The valid bond identifiers for the relevant convention may be used.

Bond Identifier

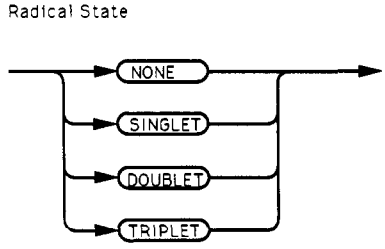
(19) Hydrogen Count Subblock. The value given for the hydrogen count is the sum of all attached hydrogen atoms and isotopes, including any that have been given explicit entries in the >NODE block. An integer set is used for variable hydrogen counts (as in query structures), and a node set may be used for delocalized hydrogens. It is not compulsory to give



a hydrogen count value for all nodes.

(20) Radical Subblock. A node set may be used for delocalized radicals and a set of radical state keywords for a variable radical state. It is not compulsory to specify a radical state for all nodes.

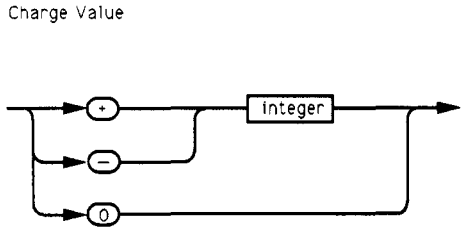
(21) Radical State. The keywords have the following meanings: NONE, no radical state (system default);



SINGLET, singlet biradical; DOUBLET, "ordinary" radical; TRIPLET, triplet biradical.

(22) Charge Subblock. A node set may be used for delocalized charges. A set of values for variable charges may be given (as in queries), in which case a range implies all charge values from the first to the second inclusive, including zero if applicable. The more negative charge must come first. It is not compulsory to specify a charge for all nodes.

(23) Charge Value. An explicit sign must be attached to all positive and negative values.



(24) Valency Block. This block specifies the valency taken by a particular atom in a particular context. The default valency assumed when the default keyword is IMPLIED is that implied by the bonds shown in the)BOND and)HCOUNT subblocks and the >ATTACHMENT block. There are no general default values for the valencies of individual elements.

(25) Isotope Block. A particular atomic mass or set of atomic masses may be specified for a node. The default isotope assumed is the naturally occurring mixture of isotopes.

(26) Stereo Block. The >STEREO block is the preferred form of stereochemical specification in SMD Format, and it shows the configuration or conformation at a stereo center by means of an ordered rotational list.

Each subblock describes a different geometry, and the geometries corresponding to each keyword are shown in Figure 1. The diagram shown for each geometry specifies a citation order for the nodes, and each data line in the subblock consists of a list ("tuple") of node identifiers, cited in the appropriate order. The number of node identifiers in the data line is determined by the geometry as defined in the subblock header.

Various permutations of the order of the node identifiers in the data line tuples can be used to describe the same configuration. Such equivalent permutations need to be taken

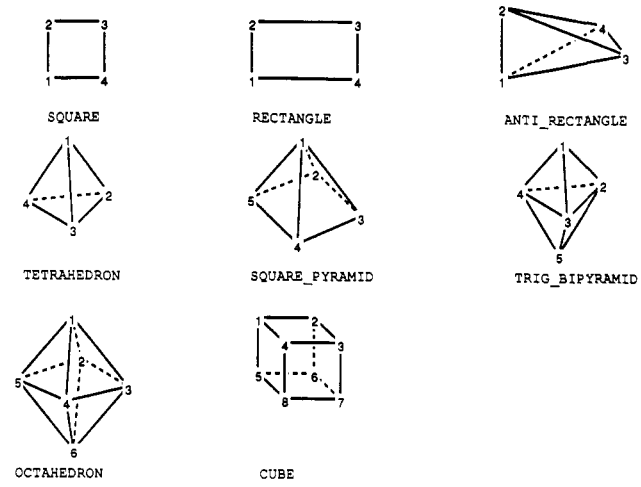
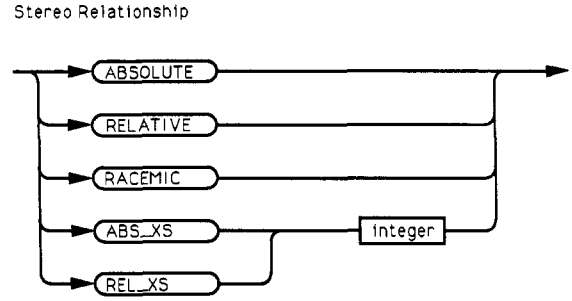


Figure 1. Set of allowed geometries in subblocks of a >STEREO block. The citation order for the nodes in each geometry is indicated (further explanation in the text).

into account when attempts are made to match stereo elements described by this method. The allowable permutations for a particular configuration depend on the symmetry of the geometry being described.

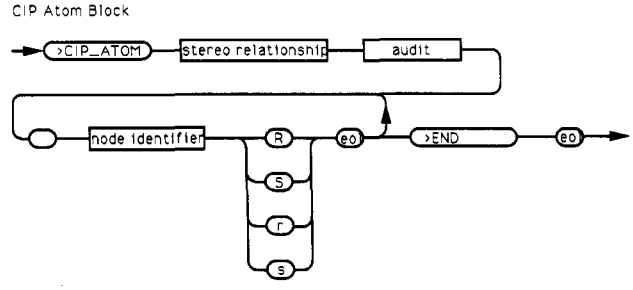
If hydrogen atoms or lone pairs are to be included in the stereochemical specification, they must be given explicit entries in the connection table. Stereochemical specification involving "free sites" in queries is possible: the free site is indicated by the special node identifier "0".

(27) Stereo Relationship. The meaning of the keywords is as follows: ABSOLUTE, the configuration of the stereo el-

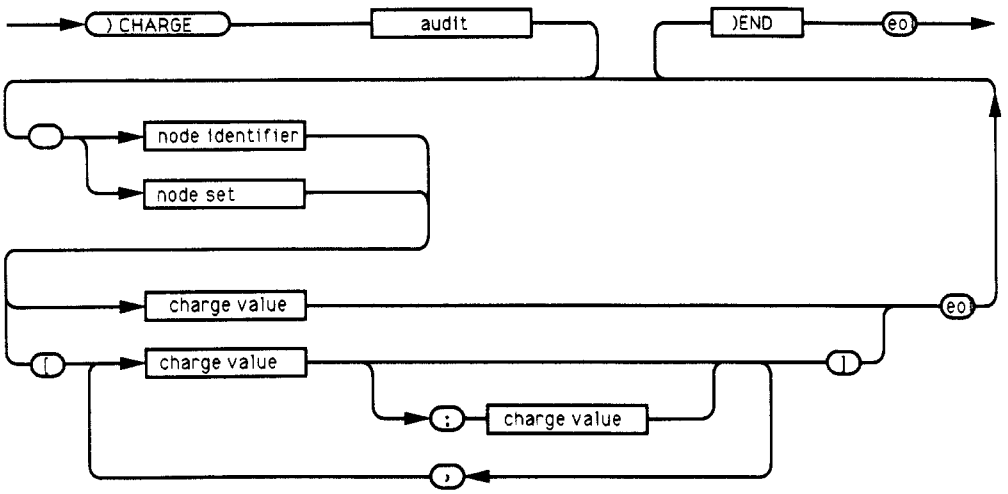


ement is exactly as described; RELATIVE, the configuration is only relative, and mirror reflection of all stereo centers in the block describes the same structure; RACEMIC, the block describes the relative configuration of the stereo centers, but the compound is a 50:50 mixture of both stereo isomers; ABS_XS, REL_XS, there is a mixture of two stereo isomers, one of which is present in enantiomeric excess. (For ABS_XS the predominant enantiomer is exactly that described; for REL_XS the predominant enantiomer is unknown, but is either that described or its mirror image. In both cases, the *E* value for the enantiomeric excess is given.)

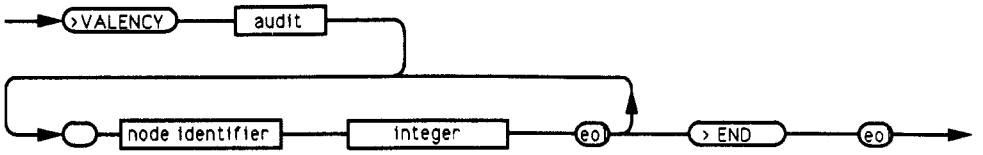
(28) CIP Atom Block. The Cahn–Ingold–Prelog stereo descriptor[15,16] specified by the keyword is applied to the node indicated.



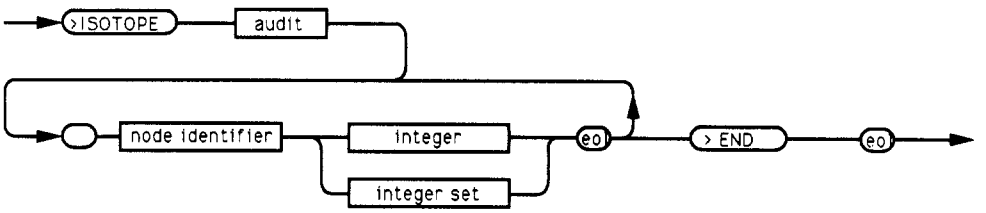
Charge Subblock



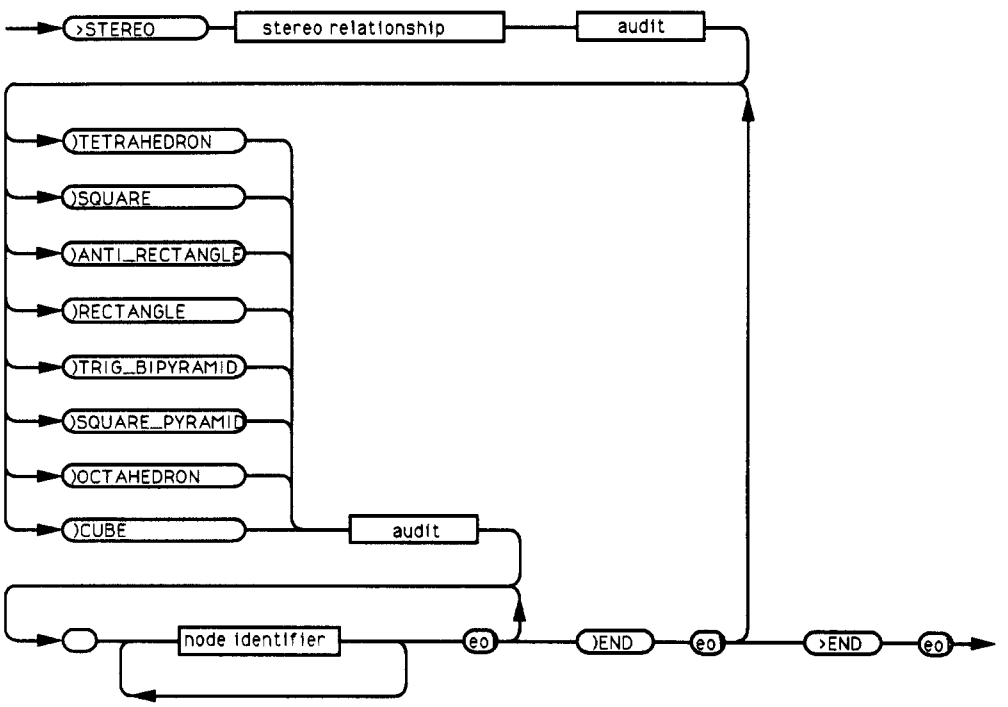
Valency Block



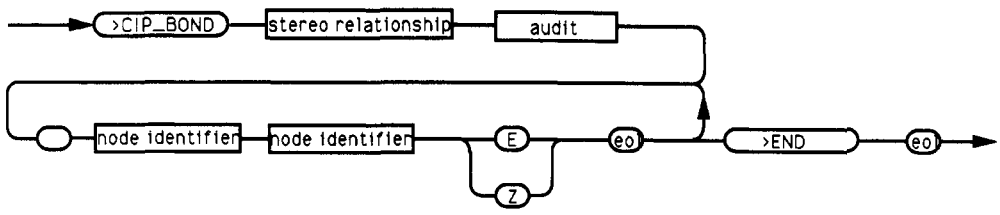
Isotope Block



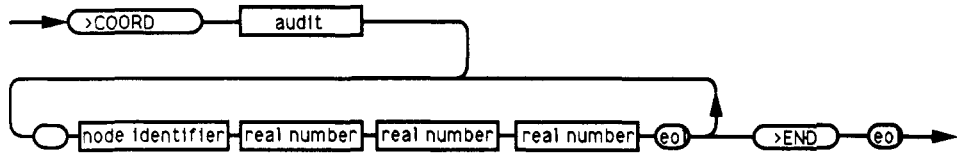
Stereo Block



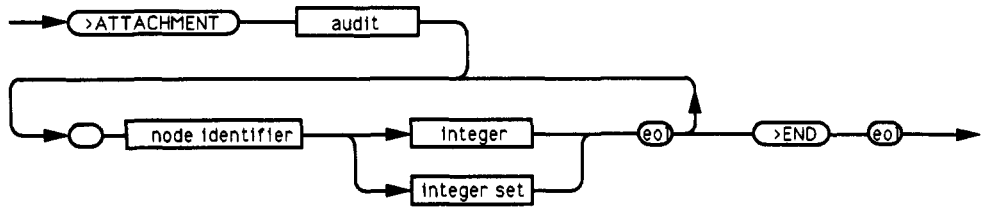
CIP Bond Block



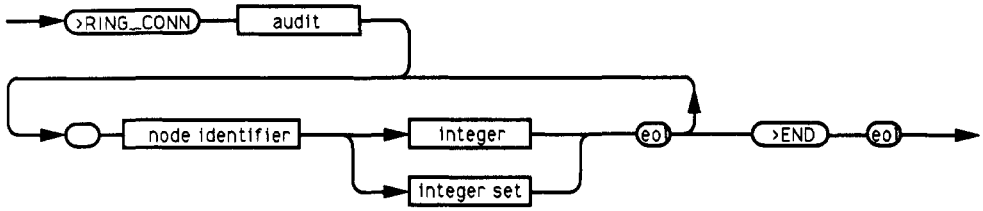
3D-Coordinates Block



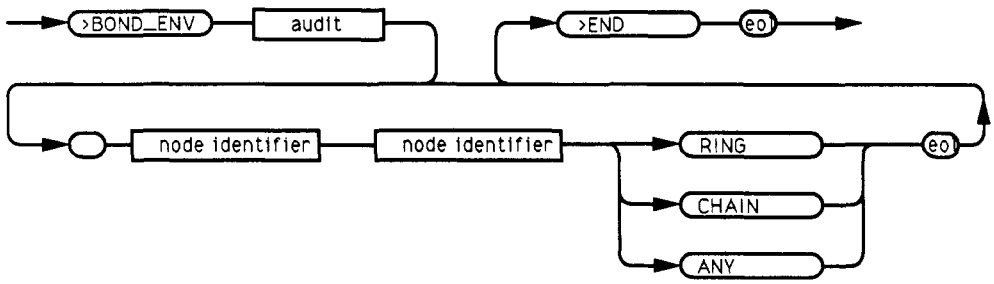
Attachment Block



Ring Connectivity Block



Bond Environment Block



(29) CIP Bond Block. The Cahn–Ingold–Prelog stereo descriptor[15,16] specified by the keyword is applied to the bond indicated.

(30) 3D Coordinates Block. Absolute three-dimensional spatial coordinates can be specified for the atoms in the molecule. The values are specified in the order *X* coordinate, *Y* coordinate, *Z* coordinate and are in angstrom units.

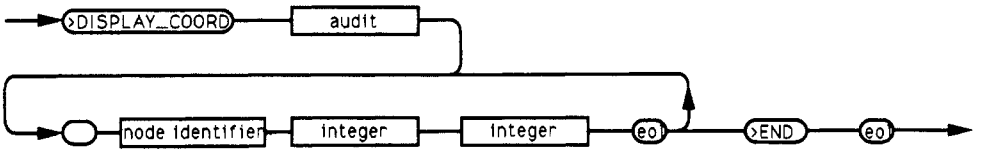
(31) Attachment Block. The values in this block specify the number of attachments at a node. This is defined as the total number of neighbors of the node, excluding all implicit and explicit hydrogen atoms and lone pairs.

The >ATTACHMENT block can be used to specify free sites in structure queries. The system default for >ATTACHMENT is IMPLIED (i.e., the attachments that are actually shown in the)BOND subblock). If this is changed to ANY, then all nodes become free sites. If left at IMPLIED, then individual nodes may be shown as free sites by giving an appropriate integer set value in the >ATTACHMENT block.

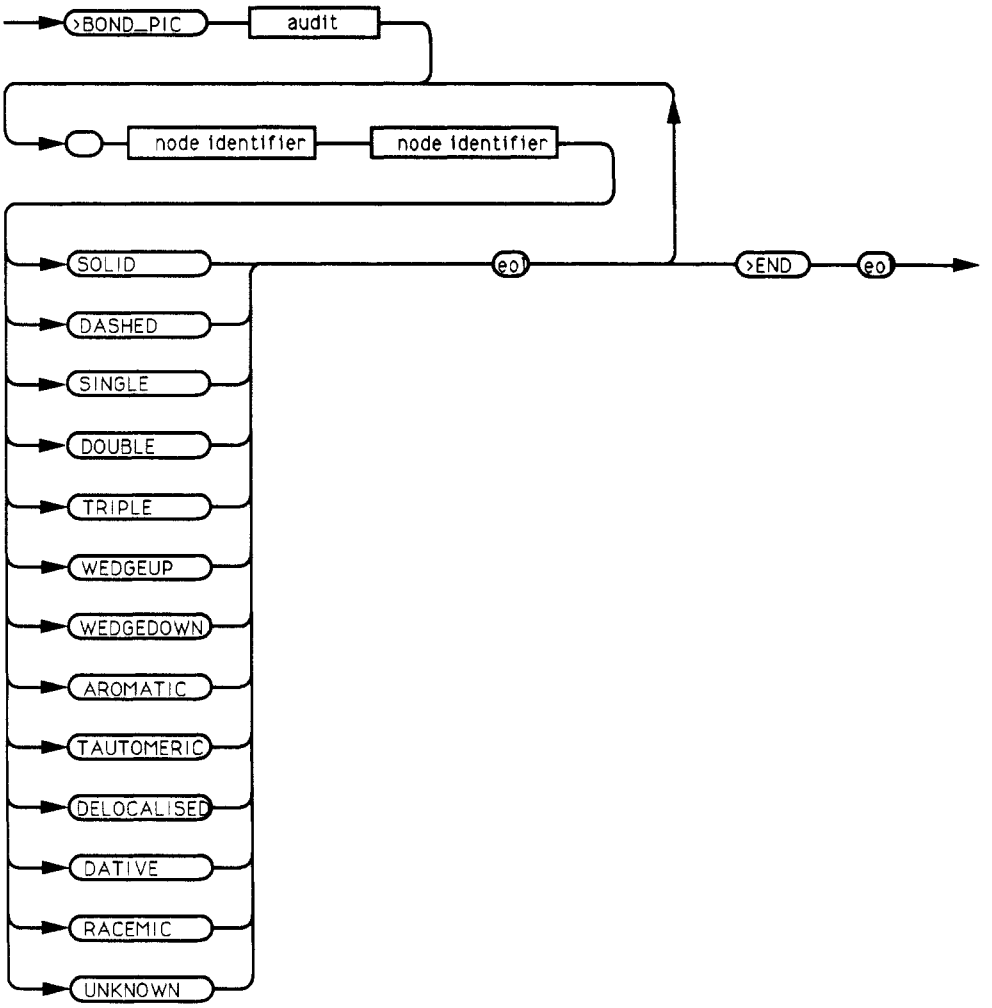
(32) Ring Connectivity Block. This block is analogous to the >ATTACHMENT block and specifies the "ring connectivity" for an atom (i.e., the number of ring bonds that it has). The block can be used in queries to specify whether or not any "free site" attachments should form parts of additional rings.

(33) Bond Environment Block. This block can be used in queries to define the ring/chain environment of bonds. The bond is identified by the nodes that it connects. Bonds that are self-evidently in a ring must have the attribute RING, but

Display Coordinates Block



Bond Picture Block



bonds that appear to be in a chain may be given any of the available attributes.

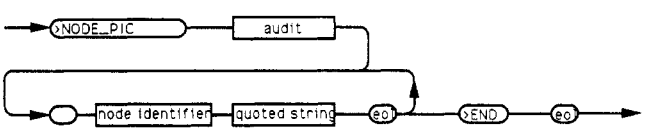
The system default (IMPLIED) is that bonds which appear to be chain bonds are chain bonds.

(34) Display Coordinates Block. This block gives the two-dimensional coordinates for display of a molecule or fragment. The display coordinates for a fragment may be (re)defined in a /MOLECULE section that refers to it.

The values specified may lie in the range 0–32 767 (15-bit unsigned integer). The origin is assumed to be at the bottom left of the screen, and a separate coordinate system is assumed for each section containing a >DISPLAY_COORD block.

(35) Node Picture Block. This block can be used to specify a string that is to be displayed as a node symbol. This may be especially useful when fragments within molecules are displayed.

Node Picture Block



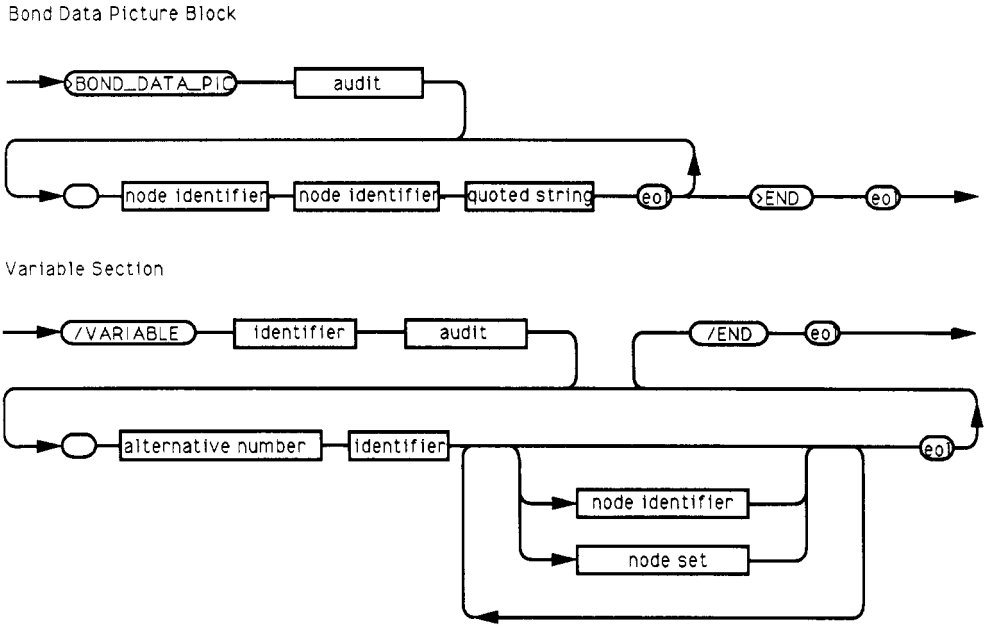
A >NODE_PIC block is not necessary for drawing: in many cases, the drawing program may simply use the information in the >NODE block to determine what to display.

(36) Bond Picture Block. This block can be used to indicate what is to be displayed for a bond. The keyword given is intended to be a "hint" to a drawing program, and SMD Format does not define any way of interpreting the meaning of the keyword. The two nodes specified do not necessarily have to have a bond between them specified in a)BOND subblock (for example, for hydrogen bonds, etc.)

The keywords WEDGEUP, WEDGEDOWN, and DATIVE indicate bond types that are directional. In these cases the order of the two node identifiers in the data line is significant; the bond goes from the first node to the second node.

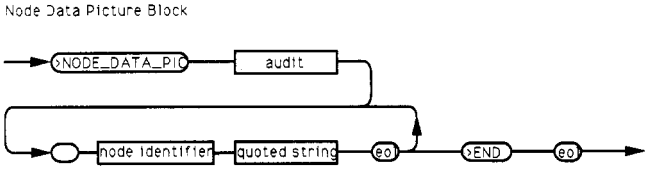
A >BOND_PIC block is not necessary for drawing: in many cases, the drawing program may simply use the information in the)BOND block to determine what to display.

Though programs may use the WEDGEUP, WEDGEDOWN, etc. keywords to deduce stereochemistry, SMD



Format does not define any rules for such interpretation. It is possible that such rules may be introduced in a future version of the Format, perhaps on the basis of the proposals by Maehr.[17]

(37) Node Data Picture Block. This block specifies data to be displayed next to a particular node symbol (e.g., an atom number).



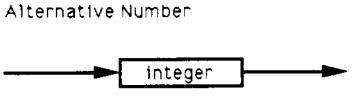
(38) Bond Data Picture Block. This block specifies data to be displayed next to a particular bond symbol.

(39) Variable Section. A /VARIABLE section describes a set of alternative fragments and may be used in generic structure and structure query representation. The header line for the section gives an identifier for the name of the variable. The /VARIABLE section is not divided into blocks or subblocks but consists simply of a set of data records, each describing one alternative value for the variable.

Each data line contains an identifier, which references a /FRAGMENT section, and a list of node identifiers or node sets which specify the attachment points in that fragment. The /FRAGMENT sections specified in a /VARIABLE section may themselves contain variables, to any level of nesting.

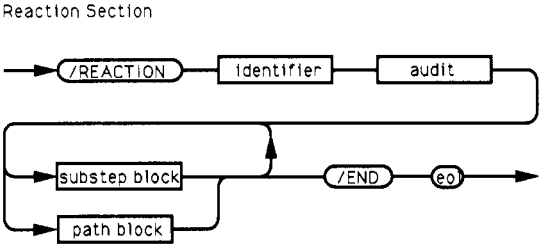
The number of attachment points given is the number of connections that the variable has. There is no limit on the number of connections a variable may have. All the data lines in a particular /VARIABLE section must have the same number of attachment points. The order of citation of these attachment points is significant and is referred to by the attachment numbers in node identifiers. Where any of the attachment points is at a variable position, a node set is used.

(40) Alternative Number. The alternatives are numbered sequentially from one. The alternative numbers may be used,



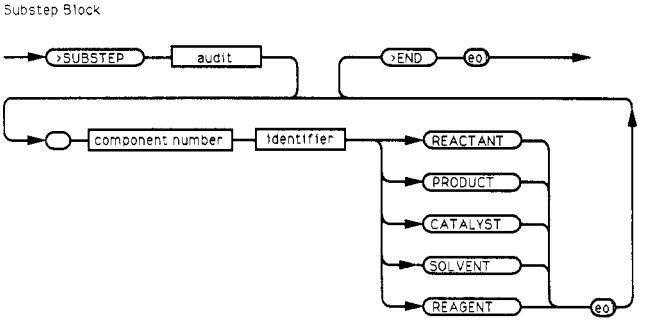
in a future enhancement to the format, in some sort of logic specification for the relationship between values of different variables.

(41) Reaction Section. A /REACTION section describes a single chemical reaction by reference to /MOLECULE



sections describing the molecules that participate in the reaction. Future enhancements to the Format may introduce additional sections for the description of reaction sequences and schemes.

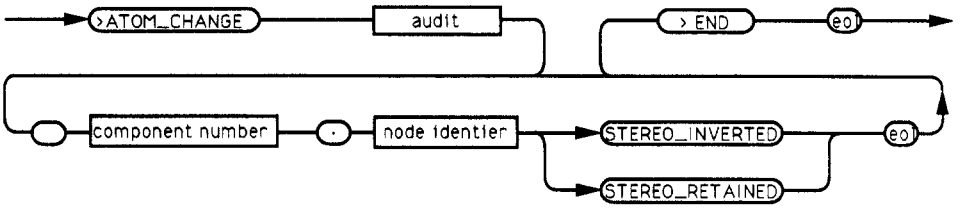
(42) Substep Block. Each >SUBSTEP block within a /REACTION section lists the molecules participating in the reaction substep in question. In most cases there will be only one >SUBSTEP block; more will be required in situations



where there are several stages to the reaction [for example, an additional reagent added partway through the reaction, but product(s) only identified after the final substep].

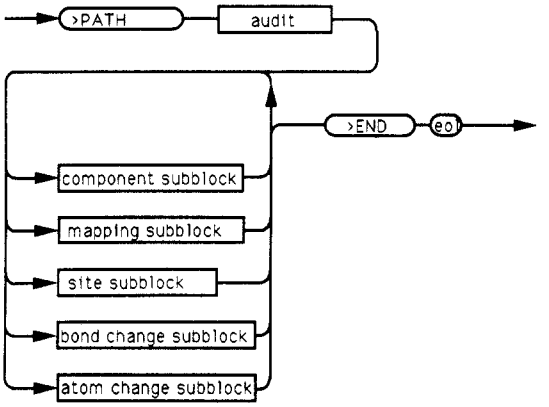
Each data record in a >SUBSTEP block refers to a /MOLECULE block by its identifier. Component numbers are allocated sequentially throughout all >SUBSTEP blocks in the whole /REACTION section. The keyword indicates the role played by the component in the reaction as a whole. It is important to note that >SUBSTEP blocks are not used to represent different steps in multistep syntheses; the latter are shown by using several /REACTION sections, which may refer to some of the same /MOLECULE sections.

Atom Change Subblock



(43) Path Block. A >PATH block describes the relationship between reaction components. Several >PATH blocks

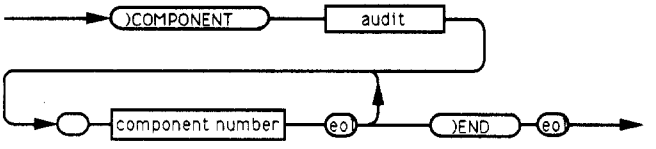
Path Block



may be used in a single /REACTION section to describe, for example, byproducts and reacting mixtures. The information in a >PATH block may be given in a number of different ways, each shown in a different subblock.

(44) Component Subblock. This subblock simply lists the component numbers (as defined in the >SUBSTEP blocks)

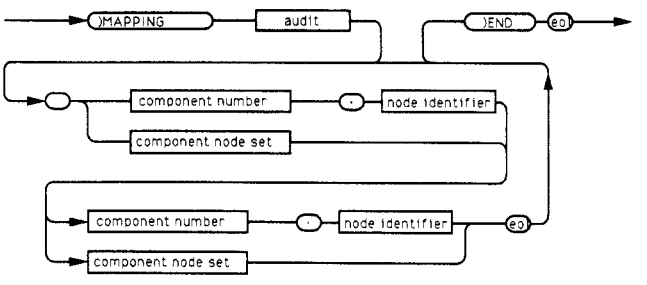
Component Subblock



of the molecules which participate in this step. It will normally be used only when no)MAPPING subblock is present.

(45) Mapping Subblock. This subblock indicates the atom–atom mapping between REACTANT and PRODUCT components. The first-cited component number must be a

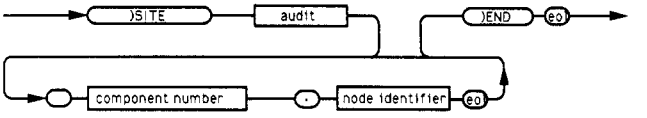
Mapping Subblock



REACTANT and the second a PRODUCT. Sets of nodes may be used if the exact mapping is unknown or variable (as in esterifications). Correspondences need not be given for all reactant and product atoms and may be given at the fragment or variable identifier level.

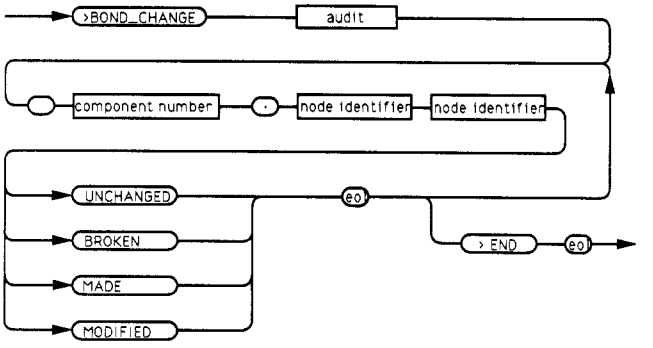
(46) Site Subblock. This subblock lists the nodes that are part of the reaction site. The identification of these is subjective.

Site Subblock



(47) Bond Change Subblock. This subblock can be used in query structures to define bond changes taking place. If a

Bond Change Subblock

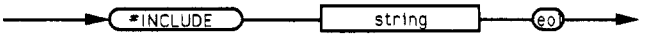


)MAPPING subblock is given, the)BOND_CHANGE subblock is redundant. Because both atoms specified must be in the same component, the component number need only be given for the first node.

(48) Atom Change Subblock. This subblock can be used to specify stereochemical changes at particular nodes in reaction queries. If a)MAPPING subblock is given, it is redundant.

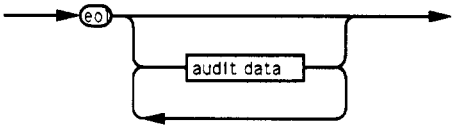
(49) Include File Specification. The string gives the file name for the file to be included. Its exact form will of course depend upon the computer operating system in question.

Include file specification



(50) Audit. Audit data are not compulsory, but may follow any header line on a separate line or lines. The data apply

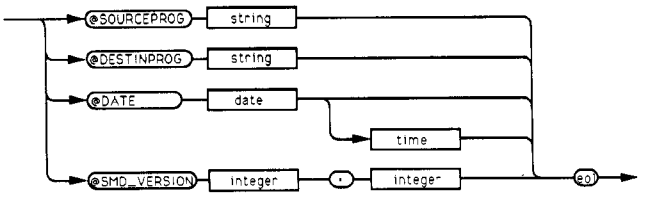
Audit



throughout the syntactical unit in question, unless locally redefined for a subordinate syntactical unit.

(51) Audit Data. The meanings of the keywords are as follows: @SOURCE_PROG, the program that wrote this part of the file; @DESTIN_PROG, the intended destination

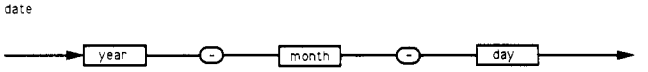
Audit Data



program; @DATE, the date and time at which this part of the file was written; @SMD_VERSION, the SMD version number being used.

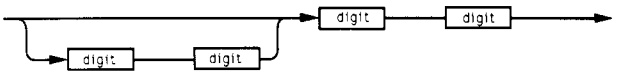
(52) Date. The date must conform to the "extended format" of the ISO standard for representation of dates.[18]

date



(53) Year. The digits for the century are optional.

year

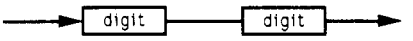


(54) Month. Leading zeros must be included.

month

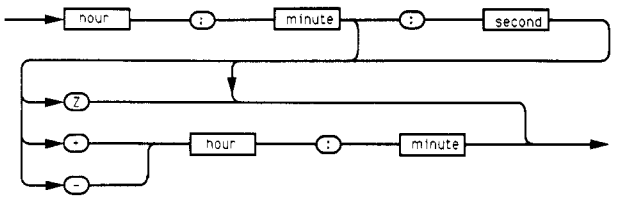
(55) Day. Leading zeros must be included.

Day



(56) Time. The time must conform to the extended format of the ISO standard for the representation of times.[18] The

Time



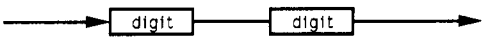
"T" separator between the date and time should not be used. If no time-zone suffix is given, the time is assumed to be local. The suffix "Z" designates Coordinated Universal Time (UTC), which is frequently, though inaccurately, referred to as Greenwich Mean Time (GMT). Other time-zone suffixes indicate the difference from UTC.

(57) Hour. Leading zeros must be included.

Hour

(58) Minute. Leading zeros must be included.

Minute

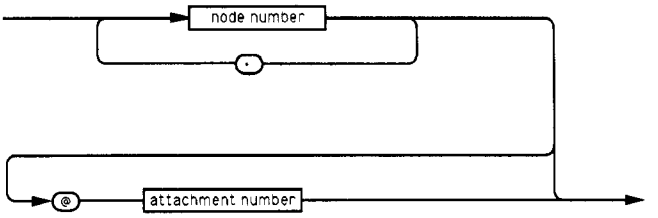


(59) Second. Leading zeros must be included.

Second

(60) Node Identifier. The initial sequence of node numbers allows reference to nodes in nested fragments. Only nodes

Node Identifier

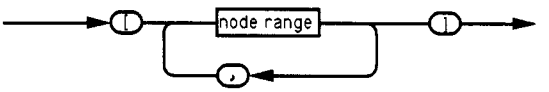


that are fragments may be followed by periods. Thus, the node identifier 4.3.6 refers to node 6 in the fragment that occurs as node 3 of the fragment that occurs as node 4 in the section being referred to.

If the node ultimately identified by the first part of the node identifier is a variable node, then one of the nodes in the alternative values for the variable can be identified by means of the attachment number. Only a node that is a variable may be followed by the "@" symbol. Because several alternative nodes are possible when reference is made to a variable, further identification of nodes within the value for a variable is not possible.

(61) Node Set. A node set allows a set of nodes to be specified. The elements of the set are separated by commas.

Node Set

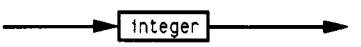


(62) Component Node Set. This is analogous to a node set, except that a component number must also be given.

(63) Node Range. This defines a range of node numbers, which may be used as a shorthand in a node set. The colon implies inclusion of all node numbers (or attachment numbers) from the first specified to the second specified inclusive. Ranges may only be defined within the most nested fragment or variable.

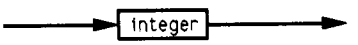
(64) Node Number. Node numbers are defined in the >NODE block of a /FRAGMENT or /MOLECULE section. They are unique within a section.

Node Number



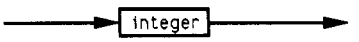
(65) Attachment Number. The attachment number refers to a position in the sequence of attachment points listed in a /VARIABLE section.

Attachment number



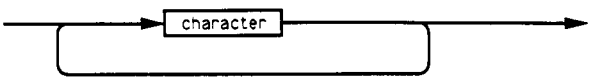
(66) Component Number. Component numbers are defined in the >SUBSTEP block of a /REACTION section. They are unique within a /REACTION section.

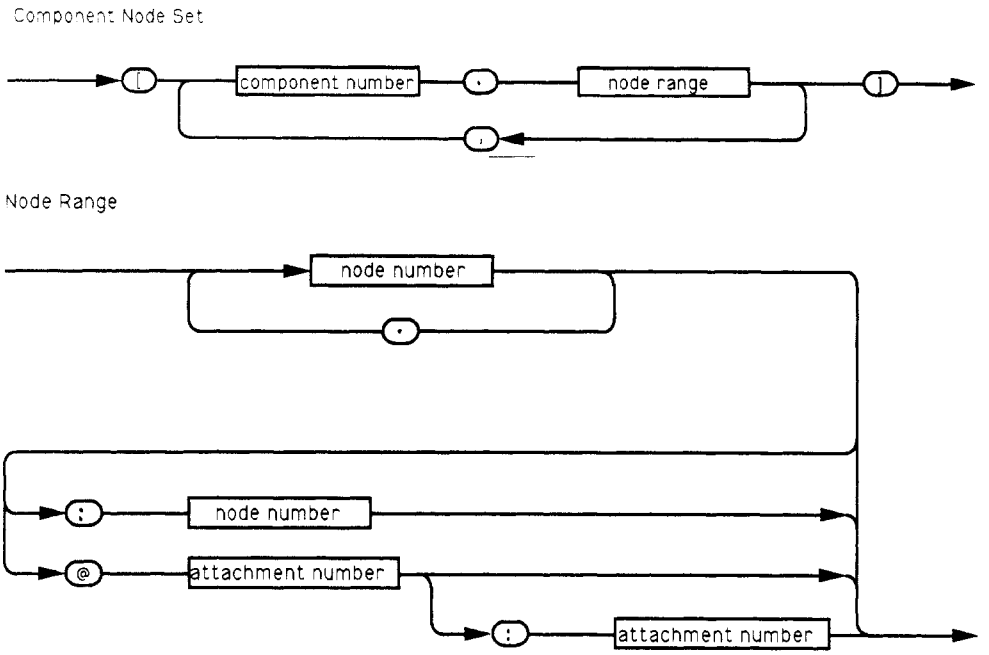
Component number



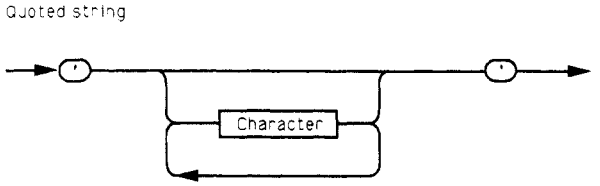
(67) String. Any characters from the character set in use are permitted.

String

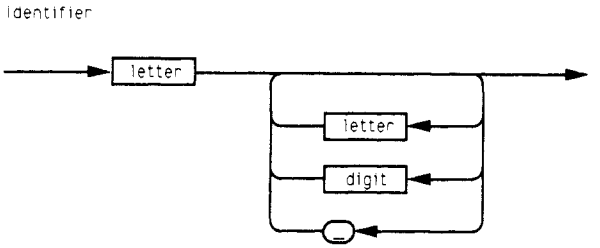




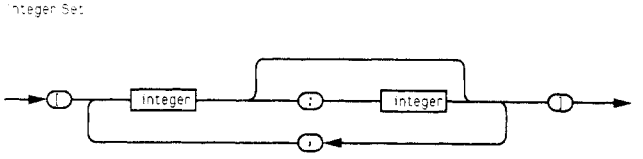
(68) Quoted String. If the quote character itself is to be included in the string, it should be shown twice.



(69) Identifier. Identifiers must start with a letter. Although they may be of any length, they must be unique in the first 16 characters.

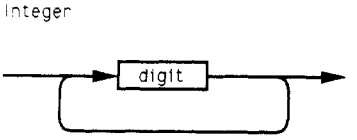


(70) Integer Set. This defines a set of integers. The colon is a range constructor, and the comma separates elements in

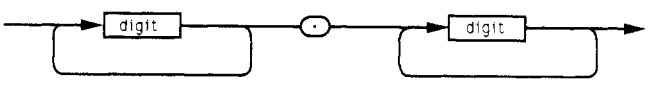


the set. Within ranges, the first integer must be less than the second, and the range implies inclusion within the set of all integers from the first to the second inclusive.

(71) Integer. Only positive integers are permitted. Negative values for charges are accounted for in the syntax for charge value.



(72) Real Number. Only floating-point representations are permitted; there must be at least one digit on either side of the decimal point.




```
)GLOBAL
/DEFAULT
/END
)END
)SCOPE
@SOURCE_PROG MANUAL
@DATE 89-11-22
/FRAGMENT phenyl
>NODE
)ATOM
 1 C
 2 C
 3 C
 4 C
 5 C
 6 C
)END
>END
>CONVENTION SIMPLE
)BOND
 1  2 DOU
 1  6 SIN
 2  3 SIN
 3  4 DOU
 4  5 SIN
 5  6 DOU
)END
>END
/END
/MOLECULE Ex_1
>NODE
)ATOM
  1 N
  2 C
  3 C
  4 C
  5 C
  6 C
  7 C
  8 C
 10 C
 11 N
 12 O
)END
)FRAGMENT
 9 phenyl
)END
>END
>CONVENTION SIMPLE
)BOND
  1  2   SIN
  1  6   DOU
  1  7   SIN
  2  3   DOU
  3  4   SIN
  3 10   SIN
  4  5   DOU
  5  6   SIN
  7  8   SIN
  8  9.1 SIN
 10 11   SIN
 10 12   DOU
)END
)HCOUNT
  1  0
  2  1
  3  0
  4  1
  5  1
  6  1
  7  2
  8  2
 9.1 0
 9.2 1
 9.3 1
 9.4 1
 9.5 1
 9.6 1
 10  0
 11  2
 12  0
)END
)CHARGE
 1 +1
)END
>END
>ISOTOPE
 11 15
>END
>DISPLAY_COORD
  1 20400 19400
  2 23100 20600
  3 23100 23000
  4 20400 24200
  5 17700 23000
  6 17700 20600
  7 20400 17000
  8 17700 15800
  9 15000 17000
 10 25800 24200
 11 28500 23000
 12 25800 26600
>END
>NODE_PIC
 9 'Ph'
>END
>NODE_DATA_PIC
 11 15
>END
/END
)END
```

Figure 2. SMD Format description for example 1 (discussion in the text).

```
)GLOBAL
/DEFAULT
 CHARGE ANY
 RADICAL ANY
 HCOUNT ANY
 VALENCY ANY
 ATTACHMENT ANY
 RING_CONN ANY
 BOND_ENV ANY
 ISOTOPE ANY
 STEREO ANY
 CIP_ATOM ANY
 CIP_BOND ANY
/END
)END
/MOLECULE Ex_2
>NODE
)ATOM
 1 C
 2 C
 3 N
 4 [O, S]
)END
>END
>CONVENTION SIMPLE
)BOND
 1 2 SIN
 2 3 SIN
 2 4 [SIN, DOU]
)END
>END
>ATTACHMENT
 1 [1:4]
 2 3
 3 [1:3]
 4 [1:2]
>END
>RING_CONN
 1 2
>END
>BOND_ENV
 1 2 ANY
 2 4 ANY
 2 3 CHAIN
>END
/END
```

Figure 3. SMD Format description for example 2 (discussion in the text).

```
)GLOBAL
/DEFAULT
/END
)END
)SCOPE
/CONTENT
 FRAGMENT
 VARIABLE
 MOLECULE
 REACTION
/END
/FRAGMENT methyl
>NODE
)ATOM
 1 C
 2 *
)END
>END
>CONVENTION SIMPLE
)BOND
 1 2 SIN
)END
)HCOUNT
 1 3
)END
>END
/END
/FRAGMENT ethyl
>NODE
)ATOM
 1 C
 2 C
 3 *
)END
>END
>CONVENTION SIMPLE
)BOND
 1 2 SIN
 1 3 SIN
)END
)HCOUNT
 1 2
 2 3
)END
>END
/END
```

```
/VARIABLE R1
 1 methyl 1
 2 ethyl  1
/END
/MOLECULE carboxylic_acid
>NODE
)VARIABLE
 1 R1
)END
)ATOM
 2 C
 3 O
 4 O
)END
>END
>CONVENTION SIMPLE
)BOND
 1@1 2 SIN
 2   3 DOU
 2   4 SIN
)END
)HCOUNT
 2 0
 3 0
 4 1
)END
>END
/END
/MOLECULE alcohol
>NODE
)ATOM
 1 C
 2 C
 3 O
)END
>END
>CONVENTION SIMPLE
)BOND
 1 2 SIN
 2 3 SIN
)END
)HCOUNT
 1 3
 2 2
 3 1
)END
>END
/END
```

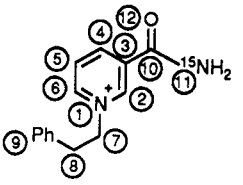
```
/MOLECULE ester
>NODE
)VARIABLE
 1 R1
)END
)ATOM
 2 C
 3 O
 4 O
 5 C
 6 C
)END
>END
>CONVENTION SIMPLE
)BOND
 1@1 2 SIN
 2   3 DOU
 2   4 SIN
 4   5 SIN
 5   6 SIN
)END
)HCOUNT
 2 0
 3 0
 4 0
 5 2
 6 3
)END
>END
/END
/REACTION esterification
>SUBSTEP
 1 carboxylic_acid REACTANT
 2 alcohol REACTANT
 3 ester PRODUCT
>END
>PATH
)MAPPING
 1.1        3.1
 1.2        3.2
 1.3        3.3
 [1.4, 2.3] 3.4
 2.1        3.6
 2.2        3.5
)END
>END
/END
)END
```

Figure 4. SMD Format description for example 3 (discussion in the text).

EXAMPLES

Figures 2–4 show examples of structure descriptions using SMD version 5.0.

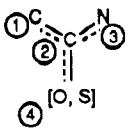
Example 1 (Figure 2). This shows the following molecule, part of which (a phenyl group) is separately shown as a fragment. Allocated node numbers are shown in circles.



The initial $GLOBAL scope contain a /DEFAULT section that sets all the defaults to the system default. The remainder of the file is a single $SCOPE containing the phenyl /FRAGMENT section and a /MOLECULE section that refers to it. The audit data given apply to both of these.

The /FRAGMENT section defines only the nodes and bonds in the phenyl group itself; information about how it is connected to the rest of the molecule is given in the /MOLECULE section. The /MOLECULE section contains both an)ATOM subblock and a)FRAGMENT subblock in the >NODE block, the latter referring to the phenyl fragment. One of the entries in the)BOND subblock shows that node 8 of the molecule is connected to node 1 of the fragment at node 9 by a single bond. Hydrogen count values are specified not only for the nodes in the main part of the molecule but also for the atoms in the phenyl fragment. The charge on node 1 and the nitrogen isotope at node 11 are specified, and display coordinates are given for the nodes in the molecule. The string "Ph" is defined as the display string for node 9, and the >NODE_PIC block shows that the isotope label for node 11 is to be displayed. The display program may, of course, display isotope labels in any case, but the use of the >NODE_PIC block emphasizes the need to display it in this case.

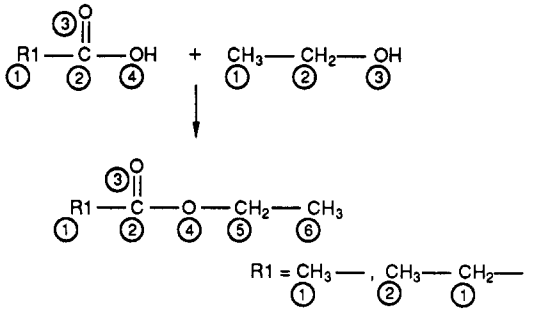
Example 2 (Figure 3). This shows the following simple substructure search query. Allocated node numbers are shown in circles.



The /DEFAULT section sets all the defaults to "ANY", thus minimizing the restrictions on answers to the query. The query substructure itself is expressed as an unscoped /MOLECULE section (no $SCOPE is required as the query does not need to refer to nor be referred to by any other sections).

Atom 4 is shown to be either oxygen or sulfur as alternatives, and all the bonds are shown as single or double as alternatives. The >ATTACHMENT block shows that atoms 1, 2, and 4 are free sites, but that atom 2 cannot be further substituted. The >RING_CONN block shows that atom 1 must be in a ring, but must not be a ring fusion point, since exactly two ring bonds are specified. The applicable defaults indicate that the other atoms may have any number of ring bonds consistent with other specifications. The BOND_ENV block shows that the bonds between atoms 1 and 2 and atoms 2 and 4 may be either ring or chain but that the bond between atoms 2 and 3 must be a chain bond. Note that this does not preclude the possibility of atom 3 being in a ring.

Example 3 (Figure 4). This shows the following simple chemical reaction, in which one of the reactants and the product are expressed as generic structures. Allocated node numbers are shown in circles.



Initially, the two alternative values for the variable group are defined in /FRAGMENT sections. For each of them the special dummy node "*" is used to show the way in which the group is connected to the main structure. A /VARIABLE section then defines the variable R1 by means of references to the two alternative fragments. For each, the node number in the fragment at which attachment is made is specified.

The next three sections in the $SCOPE describe the molecules participating in the reaction; two of these include references to the variable R1, and the)BOND subblocks use the "@" symbol to specify the attachment number in the /VARIABLE section. (In fact, as the variable has only one attachment point, this does not convey any further information, but the rules of the Format require it to be given.)

The final section in the $SCOPE is for the reaction itself. The roles played by each molecule are identified in the >SUBSTEP block, and the >PATH block contains a)MAPPING subblock that shows the atom correspondences between the reactant and product molecules. Because (at least for the purpose of this example) the mechanism of the reaction is unknown (or not specified), there are two alternative correspondents for node 4 of the product (component 3), and this is shown by means of a component node set which includes node 4 of component 1 and node 3 of component 2.

ACKNOWLEDGMENT

The text of this draft specification has been prepared by Dr. J. M. Barnard, as Technical Secretary of the SMD Subgroup

of the Chemical Structure Association. Assistance in preparing the syntax diagrams and checking the text has been provided by A. P. F. Cook (Orac Ltd.) and Karina Gale. Thanks are due to Dr. W. A. Warr (ICI PLC), Dr. J. D. Rayner, and Dr. M. Lord (Hull University) for helpful comments on the manuscript. Thanks are also due to the following organizations, which have provided facilities for meetings of the technical working groups: Télésystèmes Questel, Paris (June 27–29, 1988), Cambridge Crystallographic Data Center, Cambridge (July 25, 1988), Beilstein Institute, Frankfurt (November 21–24, 1988), and Molecular Design MDL AG, Basel (October 2–5, 1989). Financial support for the development of SMD Format during 1988 and 1989 has been provided by the following organizations: Bayer AG (FRG), CAOS/CAMM Center, University of Nijmegen (The Netherlands), Chemical Design Ltd. (U.K.), Chemical Abstracts Service (USA), Chemical Structure Association (U.K.), Chemodata Computer-Chemie GmbH (FRG), Ciba-Geigy AG (Switzerland), Dialog Information Services Inc. (USA), European Communities Joint Research Center (Italy), Finnigan Corporation (USA), Fisons PLC Pharmaceutical Division (U.K.), Fraser Williams (Scientific Systems) Ltd. (U.K.), Glaxo Group Research (U.K.), Hampden Data Services Ltd. (U.K.), Imperial Chemical Industries PLC (U.K.), Institute for Scientific Information (U.K.), Molecular Design MDL AG (Switzerland), Orac Ltd. (U.K.), Pfizer Central Research (U.K.), Polygen (Europe) Ltd. (France), F Hoffmann-La Roche AG (Switzerland), Sadtler Research Laboratories (USA), Sandoz AG (Switzerland), Schering AG (FRG), Smith Kline and French Research Ltd. (U.K.), Télésystèmes Questel (France).

REFERENCES AND NOTES

(1) Bebak, H.; Buse, C.; Donner, W. T.; Hoever, P.; Jacob, H.; Klaus, H.; Pesch, J.; Römelt, J.; Schilling, P.; Woost, B.; Zirz, C. The Standard Molecular Data Format (SMD Format) as an Integration Tool in Computer Chemistry. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 1–5.
(2) Bebak, H. The SMD File Format. Version 4.3. Copies will be provided on request to Dr. W. T. Donner, Bayer AG, ZF-DID, Geb. Q18, D5090 Leverkusen 1, FRG.
(3) Barnard, J. M. Towards a Standard Interchange Format for Chemical Structure Data. In *Proceedings of the 12th International Online Information Meeting*; Learned Information: Oxford, 1988; pp 605–609.
(4) Barnard, J. M. Standard Representations for Chemical Information. In *Proceedings of the Montreux 1989 International Chemical Information Conference, Montreux, Switzerland, 26–28 September 1989*; Collier, H., Ed.; Springer-Verlag: Heidelberg, 1989.
(5) Barnard, J. M.; Cook, A. P. F.; Rohde, B. Storage and Searching of Stereochemistry in Substructure Search Systems. In *Beyond the Structure Diagram* (Proceedings of a Conference held at the College of St. Hild and St. Bede, University of Durham, U.K., 17–20 July 1989); Bawden, D., Mitchell, E., Eds.; Ellis Horwood: Chichester, U. K. (in press).
(6) Warr, W. A., Ed. *Chemical Structure Information Systems. Interfaces, Communication and Standards*; ACS Symposium Series 400; American Chemical Society: Washington, DC, 1989.
(7) Information concerning membership of the new organization intended to oversee the development of SMD Format is available from the Administrative Secretary, Dr. Vivienne Winterman, 80 Linton Ave., Borehamwood, Hertfordshire WD6 4QY, U.K..
(8) Brown, I. D. Standard Crystallographic File Structure. *Acta Crystallogr.* **1983**, *A39*, 216–224.
(9) George, D. W.; Mewes, H. W.; Kihara, H. A Standardized Format for Sequence Data Exchange. *Protein Sequences Data Anal.* **1987**, *1*, 27–39.
(10) Gund, P.; Barry, D. C.; Blaney, J. M.; Cohen, N. C. Guidelines for Publications in Molecular Modeling Related to Medicinal Chemistry. *J. Med. Chem.* **1988**, *31*, 2230–2234.
(11) Gasteiger, J.; Hendriks, B. M. P.; Hoever, P.; Jochum, C.; Somberg, H. JCAMP-CS. A Standard Exchange Format for Chemical Structure Information in Computer Readable Form. *Appl. Spectrosc.* (in press).
(12) International Union of Pure and Applied Chemistry, Commission on the Nomenclature of Inorganic Chemistry. Nomenclature of Inorganic Chemistry, 2nd ed. *Pure Appl. Chem.* **1971**, *28*, 1–110.
(13) International Union of Pure and Applied Chemistry, Commission on the Nomenclature of Inorganic Chemistry. Recommendation for the Naming of Elements of Atomic Number Greater than 100. *Pure Appl. Chem.* **1979**, *51*, 381–384.
(14) Mockus, J.; Stobaugh, R. E. The Chemical Abstracts Registry System. VII. Tautomerism and Alternating Bonds. *J. Chem. Inf. Comput. Sci.* **1980**, *20*, 18–22.
(15) Cahn, R. S.; Ingold, C. K.; Prelog, V. Specification of Molecular Chirality. *Angew. Chem., Int. Ed. Engl.* **1966**, *5*, 385–415, 511.
(16) Prelog, V.; Helmchen, G. Basic Principles of the CIP System and Proposals for a Revision. *Angew. Chem., Int. Ed. Engl.* **1982**, *21*, 567–583.
(17) Maehr, H. A Proposed New Convention for Graphic Presentation of Molecular Geometry and Topology. *J. Chem. Educ.* **1985**, *62*, 114–120.
(18) International Standards Organization. Data Elements and Interchange Formats—Information Interchange. Representation of Dates and Times; ISO Standard 8601, 1988.

COMPUTER SOFTWARE REVIEWS

Two FORTRAN Compilers for Microcomputers: Ryan-McFarland and Microsoft

AVI MARANI‡

USDA, ARS, Systems Research Laboratory, BARC-West, Beltsville, Maryland 20705-2350

Received October 6, 1989

FORTRAN is one of the oldest computer languages, mainly used for mathematical and engineering applications. It is also favored by many chemists and biologists. Recently, while many scientists are using microcomputers more frequently than main-frames, FORTRAN has been replaced in many cases by other languages such as Pascal or C. There are several good reasons, however, for using FORTRAN in microcomputers: (1) FORTRAN has an established standard (FORTRAN 77 or ANSI X3.9-1978), so that programs written for mainframes or minicomputers can be easily transported to microcomputers and vice versa. (2) FORTRAN is the preferred language for "number-crunching" jobs, and the new 286 and 386 microprocessors (with added numerical coprocessors) can now handle this kind of job. (3) Advanced programming techniques, such as structuring, modularity, and object-oriented

‡Permanent address: Hebrew University School of Agriculture, P.O. Box 12, Rehovoth 76100, Israel.