Converting Chemical Names to Structures with Name=Struct

Abstract

Name=Struct is CambridgeSoft’s comprehensive algorithm for converting English chemical names into chemical structure diagrams.1 It is designed to be as practical as possible, interpreting chemical names as they are actually used by chemists. In addition to recognizing most of the official rules and recommendations of International Union of Pure and Applied Chemistry (IUPAC), the International Union of Biochemistry and Molecular Biology (IUBMB), and the Chemical Abstracts Service (CAS), Name=Struct also recognizes the shorthand, slang, and neologisms of everyday usage. It is extremely tolerant of deviations from the “official” rules in regard to spaces, parentheses, and punctuation. Both regular names (“chlorobenzene”) and inverted names (“benzene, chloro-”) are supported. In addition, it has an extensive algorithm for the identification of common “typos” (typing errors, such as “mehtyl”) to increase the odds of generating structures for the names it is given.

Name=Struct is available in two forms: a batch application, and an interactive version that is part of CS ChemDraw 8.0 Ultra.

With either version, Name=Struct sets the standard for coverage and accuracy in generating structures from chemical names.

Table of Contents

General Nomenclature Issues .................. 2
Capabilities ..................................... 8
   IUPAC and CAS Names .................. 8
   Supported Procedures .................. 9
Accuracy ..................................... 10
Speed ....................................... 10
Limitations .................................... 10
   Platform Limitations .................. 10
   Chemistry Limitations ................. 11
   Structural Limitations ................. 11
   Nomenclature Limitations ............. 12
Nomenclature Classes Handled by Name=Struct
   Chains .................................... 13
   Rings ..................................... 14
   Natural Products ....................... 15
   Functional Groups Containing Nitrogen 17
   Functional Groups Containing Oxygen 18
   Acids .................................... 20
   Ions and Radicals ...................... 23
   Salts ..................................... 24
   Isotopes ................................ 25
   Stereochemistry ....................... 25
   Miscellaneous .......................... 27

In general, Name=Struct is designed to be as smart as a real chemist—if a human chemist can understand what structure is intended by a given name, then Name=Struct should manage to do so as well. Chemical names come in many styles. Some names truly do conform to published nomenclature recommendations, most commonly from IUPAC, IUBMB, or CAS. Clearly, Name=Struct needs to recognize these names, but that’s only the start of the problem.

First, each of those organizations has changed their recommendations over time. There is no way to know which version of the recommendations were used to generate any given name, and so Name=Struct must recognize names produced by all versions.

### Changes in IUPAC Recommendations

- **IUPAC 1979 recommendations**
  - (methylthio)cyclohexane
  - 2-(methylamino)-1-phenylethanol

- **IUPAC 1993 recommendations**
  - (methylsulfanyl)cyclohexane
  - 2-(methylazanyl)-1-phenylethanol

Second, many chemical names use trivial forms that have long been forbidden by all of those nomenclature bodies. Nonetheless, these trivial names are used frequently enough that most chemists will recognize their meaning, and so Name=Struct should as well.

### Trivial and Structural Names

- **Structures for trivial names generated by Name=Struct**
  - paracetamol
  - acetaminophen

- **Structural name generated by Struct=Name**
  - N-(4-hydroxyphenyl)acetamide
Finally, even though those organizations have published nomenclature recommendations, the recommendations are extremely complex and difficult to understand. Even the best-intentioned chemist will often produce names that—technically or egregiously—violate the published norms. As long as the meaning of the name remains clear, Name=Struct should be able to handle it.

To achieve this goal, Name=Struct attempts to be as flexible as possible. Capitalization, font type, and font style are completely ignored. Most punctuation is ignored as well, regardless of whether it is used correctly as per the published recommendations or not. Spelling, similarly, is important only for clarity: Name=Struct will interpret many common misspellings correctly, but proper spelling is much more likely to be interpreted correctly. More recently, extensive typo recognition has been added, increasing the likelihood that names will be interpreted correctly even if they are not technically correct.

**Capitalization**

Capitalization is completely ignored when interpreting chemical names. “Benzene”, “benzene”, and “BENZENE” clearly all represent the same compound. There are a few nomenclature rules that call for a given capitalization. Possibly the most common of these is the “n” prefix. When used before an alkane, it indicates the straight-chain form (“n-butylamine”). When capitalized, it indicates that a ligand should be attached directly to a nitrogen atom (“N-chloroaniline”).

If capitalization is ignored, there is a potential for ambiguity (“n-butylaniline”). In our experience, this is rarely an issue. Not only are names of this type uncommon, but the intended structure is almost always intended to be “the straight chain form on the nitrogen” (“N-(n-butyl)aniline”). It is much more common to find names with completely non-standard capitalization than it is to find ones where the capitalization actually makes a difference.

Name=Struct interprets all of the following names identically:

- n-butyraniline
- n-Butylaniline
- n-BUTYLANILINE
- N-butyraniline
- N-Butylaniline
- N-BUTYLANILINE

\[\text{N-Butylaniline}\]
For the most part, Name=Struct ignores punctuation. It can be present, absent, or incorrect without affecting the interpretation of a chemical name. There are a few exceptions:

**Commas** must be present in CAS-style inverted names. “Benzene, chloro-” and “chlorobenzene” are interpreted identically. “Benzene chloro” without the comma cannot be interpreted.

**Parentheses, brackets, and braces** should be used for ambiguous names. “(Trichloromethyl)silane” is different from “trichloro(methyl)silane”. They also must be paired and nested appropriately, each open parenthesis with a matching close parenthesis and so on. Otherwise, parentheses, brackets, or braces may used interchangeably.

**Some punctuation** is required to separate digits. Traditionally, this duty is performed by a comma: “1,1-” vs. “11-”, but Name=Struct will recognize any reasonable separator—for example “1-1-” as acceptable. Other roles are traditionally performed by periods, colons, and hyphens, but Name=Struct will recognize any reasonable separator there as well.

**Spaces** are ignored whenever possible, but there are a few cases where the space performs a vital role. This is most commonly important for esters. The following names represent very different structures:

**Effect of spaces in related names**
**Baseline**

Superscripts are occasionally called for by various published nomenclature recommendations. Name=Struct handles them in a way consistent with its handling of other typographical issues. If the superscript consists of a number and immediately follows some other number, they must be separated in some fashion (parentheses are most often used). Thus “1\(^3,7\)" becomes “1(3,7)” and should not be represented as the confusing “13,7”. In all other cases, the superscripted characters should follow the previous characters with no added separation: “\(N^3,N^4,N^5\)-trimethyloctane-3,4,5-triamine”, for example, should be entered as “N3,N4,N5-trimethyloctane-3,4,5-triamine”.

\[ N^3,N^4,N^5\text{-trimethyloctane-3,4,5-triamine} \]

**Fonts**

Chemical names are not usually defined by the particular font used to display the name. There is one exception to this rule. Some chemical names are supposed to use Greek characters. Unfortunately, not only is this nuance lost on many people, but it is also impossible in many circumstances, such as in a text-only database. Accordingly, Name=Struct will recognize any reasonable representation of Greek characters. The following names are all interpreted identically:

\[ \alpha\text{-methylphenethylamine} \]

- a-methylphenethylamine
- alpha-methylphenethylamine
- .alpha.-methylphenethylamine
- &alpha;-methylphenethylamine

**Italicization**

All font styles, including italicization, are completely ignored when interpreting chemical names. This will never introduce any ambiguity.
Spelling

Spelling is critically important. There are many pairs of names that differ only by a single character—compare methylamine and menthylamine:

Similar spellings

However, in many cases, a human chemist might not even notice an incorrect spelling, and would interpret the name easily. Name=Struct should do the same. Many common misspellings, including “choro”, “cloro”, and “flouro” are automatically recognized (see “Automatic error recognition” below).

Elision

Elision refers to the elimination of one vowel that is directly followed by another, and is technically required or prohibited in many cases: the rules say to use “propanamine” rather than “propaneamine”. Name=Struct does not enforce those rules; vowels may be elided or not without penalty.

Automatic error recognition

Because it is designed to interpret real-life usage, Name=Struct also handles real-life misusage, and “typos” (typing errors) are incredibly pervasive. Even more to the point, errors within chemical names can be quite subtle, to the point where even a trained chemist won’t notice many “obvious” errors. As a result, without error recognition Name=Struct would fail to interpret many names that seemed quite reasonable. The following names, for example, each have a typing error. Name=Struct will find them, even if you didn’t.

- Phospine, tris(3-aminophenyl),oxide
- 4-(6-Bromoacenaphthen-5-yl)bezaldehyde
- (1R)-1,7,7-Trimethylbicyclo[2.2.1]heptan-2-one oxime
- 1-Piperazinecarboxylic acid, amide,thiono,N-cylohexyl-4-methyl
• 1-Methylethyl 2-[[ethoxy(1-methylethyl)amino]phosphinothioyl]oxy]benzoate
• 2-(N-ethoxybutyrimidoyl)-5-(2-ethoxythiopropyl)-3-hydroxy-2-cyclohexen-1-one
• 4,9-dihydro-4-(1-methyl-4-piperidinylidene)-10H-benzocyclohepta[1,2-b]thiopen-10-one

Obviously, there are a lot of ways that a name can be mis-typed. Name=Struct focuses on the four Damerau transformations of single-character additions, deletions, substitutions, and letter-pair inversions. These errors have historically been shown to account for the vast majority of errors in written text, an observation that appears to hold true even in the specific case of chemical names.

Order of substituents

Technically, the rules state that substituents must be in alphabetical order, that is “3-bromo-2-chloro-1-iodopentane” rather than “1-iodo-2-chloro-3-bromopentane” or any other ordering. Name=Struct does not enforce this rule. Substituents may be listed in any order without affecting the recognition of the chemical name.

Substitution order

Incorrect names

Correct name

Inverted names

Name=Struct can interpret CAS-style inverted names (“benzene, chloro” rather than “chlorobenzene”) without problems. That remains true even for more-complicated cases such as “1-propanaminium, 3-carboxy-2-hydroxy-N,N,N-trimethyl-, hydroxide, inner salt, (S)-”. (In its uninverted form, it

Complex inverted names

1-Propanaminium, 3-carboxy-2-hydroxy-N,N,N-trimethyl-, hydroxide, inner salt, (S)-

Language

Name=Struct is designed to interpret English chemical names only. Coincidentally, it can also interpret many German chemical names simply because the structure of the two languages is very similar. The Name=Struct algorithm could be extended to other languages without much difficulty. If you have a use for this functionality, please contact us.

Capabilities

Name=Struct is designed to be as complete, accurate, and fast as possible. It can be used with confidence to interpret one name or a million, whether those names follow any official published nomenclature recommendations or not. Name=Struct recognizes >90% of organic nomenclature recommendations. While the figure is somewhat lower for inorganic nomenclature, all general procedures and all recommendations that occur with any frequency in real-life usage are recognized. Testing over many different data sources has shown that with a typical database that is a combination of well-formed names, trade names, trivial names, and incorrect or misspelled names, Name=Struct will generate structures for about 70-90% of the names actually used. When running in batch mode, it can easily process over 20,000 names/minute, with an accuracy of >99%.

In most cases, Name=Struct will generate a reasonable structure. (See Limitations for types of structures that it cannot generate.) However, the structure it generates may not be the one you had in mind. In the case of unspecific (e.g. “methyl phenol”) or ambiguous (e.g. “2-chloroethylbenzene”) input, it will display only the single structure that it deems most likely. In cases such as these, the addition of locants (“3-methyl phenol”) or additional parentheses (“2-chloro(ethylbenzene)”) will help ensure that the structure generated will match the structure you had in mind.

IUPAC and CAS names

Name=Struct recognizes most IUPAC and CAS names. Note, however, that many—and perhaps most—chemical names in actual use violate the nomenclature rules published by those organizations. IUPAC names and CAS names represent only a small fraction of the chemical names that are actually being used. Name=Struct does interpret IUPAC and CAS names, but it also recognizes many types of nomenclature usage (and misusage) that are discouraged or even forbidden by the published rules.
Name=Struct supports basically all nomenclature procedures:

- **Substitutive names** describe the replacement of a hydrogen with some other ligand. This is by far the most common nomenclature procedure in organic chemistry, and includes names that list substituents as prefixes (“1,2-dichloroethane”), as suffixes (“ethanol”), and in combination (“1,2-dichloroethanol”).

- **Functional class names** have internal spaces that separate one or more parent structures from a functional class modifier. Possibly the most common examples are esters (methyl acetate), but names like methyl iodide and methyl ethyl ketone are also included. This procedure is much less common than it used to be, as names like “ethanol” are gaining popularity over the equivalent “ethyl alcohol”.

- **Replacement names** describe the replacement of individual carbon atoms by heteroatoms. These are most commonly seen in ring systems like “1,2,4-thiadiazole”, and are occasionally seen in a number of other circumstances.

- **Conjunctive names** are strictly limited to cases where a ring system is connected directly to the tail end of a chain, as in “benzenepropanol”. Names of this sort are more common in collections of highly systematic names, and especially collections of CAS names. Since this procedure is not well known, it is rarely seen in names generated by average chemists.

- **Additive names** describe names composed of multiple components where no component loses any atoms, including hydrogen. There are many different types of additive nomenclature, but salts (copper acetate, sodium chloride) are particularly well known.

- **Multiplicative names** exhibit the replication of a parent structure more than once. Names of this sort are generally limited to very symmetric compounds such as “2,2’-oxydiethanol”. As with conjunctive names, multiplicative names are most commonly seen from CAS and rarely seen in names created by average chemists.

Name=Struct handles all of the above IUPAC general nomenclature procedures more or less completely. That is, Name=Struct may fail to interpret any given name, but it would have problems because it failed to recognize some particular name fragment rather than because it couldn’t understand the principles of a substitutive name.

The only general procedure that Name=Struct fails to support completely is subtractive nomenclature.

- **Subtractive names** describe structures where one or more atoms are removed from a parent structure. By far the majority of usages of
subtractive nomenclature are in the “deoxy-” prefix for carbohydrates, nucleic acids, and amino acids, and the “nor-” prefix for natural products.

Name=Struct does interpret the common subtractive prefixes correctly. Technically, however, anything could be named using subtractive nomenclature. Methane could be named as “de(phenyl)toluene” or even recursively as de(phenyl)(phenyl(de(phenyl)toluene)). Name=Struct does not attempt to support subtractive nomenclature in the general case, but the general case is extremely obscure and rarely encountered.

**Accuracy**

Accuracy can be an extremely important question, especially when batch-converting thousands (or hundreds of thousands) of names. It’s very important that you are able to trust the output of any algorithm designed to run without supervision, and with Name=Struct, you can. In our extensive testing of many databases, including our own ChemFinder/Webserver and ChemACX, as well as many user-provided databases, we have found that the structures produced by Name=Struct are >99% accurate.

It would be nice if we could claim to be 100% accurate, but that’s never going to be realistic. The last percent includes a lot of names that are ambiguous in a variety of ways. Name=Struct is designed to interpret names in the most common and reasonable way possible. That’s usually the appropriate thing to do, but if someone intends to use a name in an unusual or unreasonable way, the structure generated by Name=Struct won’t match the structure that was intended (although it likely will match a name that could have been intended). Rather than arguing the correct behavior for these cases, we’re simply not claiming more than 99% accuracy.

**Speed**

The batch version of Name=Struct can convert:

- >20,000 names/minute on a 933 MHz Pentium III
- >30,000 names/minute on a 2.5 GHz Pentium 4

That works out to less than 5 milliseconds per name, on average. In other words, it is possible to convert over a million names in less than an hour. We know of only a few data sources with more than a million names, including the CAS and Beilstein databases.

**Limitations**

The only limitations of Name=Struct are ones that are mandated by the hardware, or by common sense chemistry, structural, or chemical nomenclature considerations.

**Platform limitations**

As a component of ChemDraw 8.0 Ultra, the interactive version of Name=Struct runs under the same configurations as ChemDraw Ultra, which means that it is available for most modern Windows and Macintosh...
computers. The system requirements for the latest version of ChemDraw are available on the CambridgeSoft web site.

The batch version is available for Windows only. It requires:

- Windows 95, 98, 2000, Me, XP, or Windows NT 4.0
- About 10 MB of free RAM
- About 3 MB of available disk space, plus enough disk space to receive any output files.

The Name=Struct algorithm itself is implemented in standards-compliant C++, and should be readily convertible to other systems. Please contact your CambridgeSoft sales representative if you think you need Name=Struct provided for some other operating system or in some other configuration.

There are very few limitations to Name=Struct in a chemical sense—no limitations to the length of name interpreted or the number of atoms in the resulting structure. Beyond a certain point, however, a structure diagram would simply get too large for a chemist to want to look at. In our opinion, that point would probably be in the order of a few thousand atoms. Still, if you ask for something larger, the interactive version of Name=Struct should be able to provide it for you.

The batch version of Name=Struct, which produces output in SDFile format, is necessarily restricted to the 999-atom and 999-bond heavy-atom limits imposed by that format. A heavy atom is, for the most part, an atom other than hydrogen. We have never seen a non-contrived name that generated a structure with greater than about 500 heavy atoms.

All elements from hydrogen to lawrencium are supported, even considering that some of them (such as helium) will rarely appear in chemical names simply because they form so few compounds.

Name=Struct does have some limitations in the types of structures it can generate. It is extremely difficult to generate meaningful or good-looking structural diagrams for several classes of substances.

**Proteins and enzymes** have extremely large structures that are defined principally by their biological activity and, occasionally, by their 3D coordinates. A 2D depiction of agarose, for example, would be neither recognizable nor useful. Name=Struct generally will not attempt to generate structures for substances of this sort.

**Highly bridged ring systems**, including cubane, fullerenes, and polyboranes, are extremely difficult to depict in an aesthetic fashion. The Name=Struct algorithm could interpret them without much difficulty, but support for these compounds has been disabled pending advances in the technology used to produce good-looking diagrams.
**Polymers** are not supported by Name=Struct. Interpretation of polymers has also been disabled pending advances in the technology used to produce good-looking diagrams.

More significant are the limitations inherent to chemical nomenclature itself. Many of the names in common use to describe various substances have no systematic component at all.

**Fully asystematic names** such as “Brilliant Green”, “Bis-Tris”, “C.I. 75660”, “Viagra™”, “compound #3”, and “brown sludge from Thursday’s reaction” convey no chemical meaning and can only be interpreted by direct database lookup. Name=Struct will recognize many of the most-common trivial and trade names, but its strength is not in that area and it is not intended to be a comprehensive interpreter of all trivial and trade names. CambridgeSoft can provide a number of other databases that are more appropriate for interpreting names of this type, but even the best database will have problems with a name like “compound #3”.

**Property-based descriptors** cannot be interpreted by Name=Struct. A prefix like “(+)-” refers to the sign of the optical rotation of the molecule, which describes a physical property that must be measured in the laboratory. In the best of circumstances physical properties can only be estimated, not predicted with absolute accuracy. Descriptors of this type are ignored by Name=Struct. If the remainder of the name can be interpreted successfully, the resulting structure will be accompanied by a warning that some descriptors were ignored.

**Ambiguity** is a constant concern in the interpretation of chemical names. Some “chemical names” actually describe a generic substance rather than a specific molecule. It is physically impossible to draw “the chemical structure of mineral oil”, for example. A name like “xylenes” makes its ambiguity explicit, but others are less obvious. “Quinhydrone” sounds like it ought to have a structure, but it, too, represents a mixture of several forms.

A name like “2-chloroethylbenzene” can readily be interpreted in at least two ways (parentheses added to clarify how each structure was generated):

\[
\text{2- chloroethylbenzene}
\]

\[
\begin{align*}
\text{2-chloro(ethylbenzene)} & \quad \text{(2-chloroethyl)benzene}
\end{align*}
\]

In developing Name=Struct, we have spent considerable time investigating how ambiguous names are actually being used by real chemists. As a result,
Name=Struct will produce the single structure that is most commonly intended by an ambiguous name—the right-hand structure in the specific example above. This will almost always generate the expected result, but we recognize that ambiguous names are ambiguous, and occasionally will be intended to represent one of the other possibilities. A future version of Name=Struct may address this problem in a more comprehensive manner.

Nomenclature classes handled by Name=Struct

The following pages show some examples of structures created with Name=Struct. These examples show the scope and flexibility of Name=Struct, but should not be taken as a list of preferred chemical names. Although some of these names do conform to published nomenclature recommendations, many others do not. All of these names, however, have appeared in various published sources, and Name=Struct produces a reasonable chemical structure for each of them.

In all examples, the name under the structure represents the exact text that was entered into Name=Struct.

Chains

Alkanes

hexane

2-ethenyl-pentane

1-Triisopropylsilyl-1-propyne

Heteroatomic parents

phenylsulfur pentafluoride

germathiane
Rings

Simple rings

4,4a-dihydronaphthalene

perhydrobenzene

Heterocycles

1,3,5-Dithiazine, 4,5-dihydro-5-methyl

1,3,5,7-Tetramethyltetravinylcyclotetrasiloxane

Ortho/peri-fused ring systems

pentacene

1,3-Diphenylbenzo[c]furan

Spiro-fused ring systems

1,6-Dioxaspiro[4.4]nonane-2,7-dione

9,9'-spirobifluorene
Rings, cont’d.

**Ring assemblies**

4,4’-biphenyldiol

[2,2′-Bi-1H-indene]-1,1′-dione, 2,2′,3,3′-tetrahydro-2,2′,3,3,3′,3′-hexahydroxy-

**Von Bayer ring systems**

bicyclo[3.3.1]tetrasiloxane

Tetracyclo[3.3.2.0(2,4).0(6,8)]dec-9-ene (1alpha,4alpha,6alpha,8alpha)-

**Natural Products**

**Amino Acids**

L-Arginine

para-iodo-D-phenylalanine hydroxamic acid
Natural Products, cont’d.

**Carbohydrates**

1,2:4,5-di-O-isopropylidene-beta-D-erythro-2,3-hexodiulo-2,6-pyranose

Erythraric acid

**Nucleic Acids**

(+)-3’,5’-O-(1,1,3,3-tetraisopropyl-1,3-disiloxanediyl)cytidine

Caution: Stereochemical terms discarded: +

**Steroids**

cholesterol

17-Hydroxy-19-nor-17alpha-pregna-4,9,11-trien-20-yn-3-one
Functional groups containing nitrogen

**Amines**

- diethylbutylamine
- N2,N2,6-trimethyl-1,3,5-triazine-2,4-diamine

**Imines**

- acetone n-phenylimine
- naphthalene-1,4-imine

**Azo**

- Hydrazobenzene
- 2,2'-Azobis(isobutyronitrile)

**Oximes, etc.**

- 2,3-Butanediione monoxime
- Acetic acid, (4-nitrophenyl)-oxo, ethyl ester, thiosemicarbazone
Functional groups containing Nitrogen, cont’d.

**Isocyanates, etc.**

- n-phenyl isocyanate
- 4-azidobenzoic acid

Functional groups containing oxygen

**Alcohols**

- 1,1-Diallylethanol
- p-methoxybenzeneselenol

**Ethers**

- diethyl ether
- diglycidyl ether of 2,2-bis(p-hydroxyphenyl)propane
- 18-Crown-6

**Ketones**

- 1,3-Cyclopentanedione
- 2-Benzylidene-1-tetralone
Functional groups containing oxygen, cont’d.

**Quinones**

- **Quinone**
- **2,3-Dichloro-1,4-naphthoquinone**

**Ketenes**

- **Butyl ethyl ketene**
- **Nitroketene dimethyl mercaptal**

**Oxides**

- **methionine oxide**
- **p-Mentha-1,8-diene 1,2-epoxide**

**Peroxides**

- **tert-Amyl peroxide**
- **(methylthio)benzene**

**Ketals**

- **Acetone diethyl ketal**
- **4-Aminobutyraldehyde Dimethyl Acetal**
Functional groups containing oxygen, cont’d.

**Glycols**

![Ethylene glycol](image1)

![Dibenzyl glycol](image2)

**Acids**

**Mono-, di-, and poly-carboxylic acids**

![Butyric acid](image3)

![Citric acid](image4)

![Acetothioic s-acid](image5)

**Aldehydes**

![Naphthalenedialdehyde](image6)

![Malonic semialdehyde](image7)

**Anhydrides**

![Acetic anhydride](image8)

![Succinic anhydride](image9)

![Butyric acid anhydride](image10)
Acids, cont’d.

**Imides**

- 2,3-naphthalimide
- 1,4-Diamino-2,3-anthracenedicarboximide

**Phenones**

- 3',5'-Dimethoxy-4'-hydroxyacetophenone
- Propanoic acid, piperazide, N-methyl-3,3,3-tris (4-chlorophenyl), hydrochloride

**Lactones**

- delta-caprolactone
- 1,8-Naphthalenesultam
Acids, cont’d.

**Esters**

- Ethylene ester of acetic acid
- 1α,5α-Tropan-3α-ol, atropate (ester)

**Oxyacids**

- Phosphorothioic acid, O,O-dimethyl S-carbomethoxymethyl ester
- Fluorosulfonic acid

**Acids as principal groups**

- Ethandiylbiscarbadithioic acid, calcium salt
- Formamidinesulfinic acid

**Other acid-like derivatives**

- p-Chlorobenzonitrile
- n-t-Butyl acetaldehyde
- Acetoin
Ions and Radicals

**Cations**

- cyclopropenium cation
- 6,7-dihydropyrido[1,2-a:2',1'-c]pyrazinediylium (Caution: A net charge appears to be present)

**Anions**

- (trimethylsilyl)acetylide (Caution: A net charge appears to be present)
- o-benzenedisulfonate dianion (Caution: A net charge appears to be present)

**Radicals**

- triphenylmethyl radical
- 2,2,4,5,5-Pentamethyl-3-imidazoline-1-oxyl, free radical
Salts

**Oxyacid salts**

- disodium phosphate
- sodium hypochlorite
- calcium permanganate

**Organic acid salts**

- diethylamine acetate
- ammonium acetate

**Organic oxide salts**

- sodium ethylate
- sodium phenolate

**Nitrogen, phosphorus cations**

- Hexane-1,6-bis[trimethylammonium chloride]
- phosphonium, tetrakis(methyl), acetate (salt)

**Transition metal complex anions**

- Potassium hexachloroiridate (IV)
- Hydrogen hexachloroosmium (IV)
Salts, cont’d.

Metal salts

\[
\begin{align*}
\text{As}^{+++} \, \text{Cl}^- \\
\text{Fe}^{+++} \\
\text{NH}_4^+ \, \text{O} \\
\end{align*}
\]

arsenic (III) chloride
ferric ammonium sulfate anhydrous

Isotopes

Isotopes

\[
\begin{align*}
\text{perdeuterobenzene} \\
\text{Dimethyl-d6 Phthalate} \\
\end{align*}
\]

Stereochemistry

(R) and (S)

\[
\begin{align*}
\text{(R)-glycidyl tosylate} \\
\text{(1S,2R)-2-Dimethylamino-1-phenylpropanol} \\
\end{align*}
\]
Stereochemistry, cont’d.

\((E)\) and \((Z)\)

\((2E,4Z)\)-Decadienal

\((E)-(+)\)-(S)-1-(4-chlorophenyl)-4,4-dimethyl-2-(1,2,4-triazol-1-yl)-pent-1-ene-3-ol

Caution: Stereochemical terms discarded: +

cis/trans for double bonds

(2-Ethyl-cis-crotonyl)urea

1,2-Dichloro-1,2-difluoroethylene (cis/trans)

cis/trans for ring substituents

4-Methyl-trans-perhydroquinolizine

trans-7,8-Dihydroxy-7,8-dihydrobenzo[a]pyrene

alpha/beta for ring substituents

Cyclopropa[d]naphthalene, 1,1a,4,4a,5,6,7,8-octahydro-2,4a,8,8-tetramethyl-, [1aS-(1alpha,4abeta,8aR*)]-
Stereochemistry, cont'd.

**meso**

1,2-Diphenylethane, alpha,beta-dinitro (meso)

Alanine, 3,3'-thiodi-, meso-

**Miscellaneous**

**Conjunctive nomenclature**

naphthalene-2-propanol

1,3-diaminopropane-ν,ν,ν',ν'-tetraacetic acid

**Replacement nomenclature**

3,5-diaza-tetradecane

2-azacyclobutanone

**Lambda convention**

1,4l4-Oxathiepane

2,4-Dioxa-3lamba2-plumbabicyclo[3.3.1]nonane
Bridging groups

9,10-o-Benzene-9,10-dihydroanthracene

cyclohexanecarbonitrile, 1,1’-hydrazobis-