



OpenEye
Scientific Software

LEXICHEM

Release 2.0.0

OpenEye Scientific Software, Inc.

January 07, 2010

CONTENTS

1	Front Matter	1
2	Introduction	3
2.1	Input Name Representation	3
2.2	Output Name Representation	3
2.3	Output Name Styles	4
2.4	Examples of Name Style Differences	4
3	Installation and Platform Notes	7
3.1	Licenses	7
3.2	General Installation	7
4	nam2mol	11
4.1	Command Line Interface	11
4.2	Command Line Options	11
5	mol2nam	13
5.1	Command Line Interface	13
5.2	Command Line Options	14
6	translate	17
6.1	Command Line Interface	17
6.2	Command Line Options	17
7	Release Notes	19
7.1	Lexichem 2.0.0	19
7.2	Lexichem 1.9	19
7.3	Lexichem 1.8	19
7.4	Lexichem 1.7	20
7.5	Lexichem 1.6	20
7.6	Lexichem 1.5	21
7.7	Lexichem 1.4	21
7.8	Lexichem 1.3	22
7.9	Lexichem 1.2	22
7.10	Lexichem 1.2	22
7.11	Lexichem 1.1	22
7.12	Lexichem 1.0	23
7.13	Lexichem 1.0.b3	23
7.14	Lexichem 1.0.b2	23
7.15	Lexichem 1.0.b1	26

7.16 Indices and tables	26
Bibliography	27
Index	29

FRONT MATTER

Copyright 1997-2009 OpenEye Scientific Software, Santa Fe, New Mexico. All rights reserved.

All rights reserved. This material contains proprietary information of OpenEye Scientific Software. Use of copyright notice is precautionary only and does not imply publication or disclosure.

The information supplied in this document is believed to be true but no liability is assumed for its use or the infringement of the rights of others resulting from its use. Information in this document is subject to change without notice and does not represent a commitment on the part of OpenEye Scientific Software.

This package is sold/licensed/distributed subject to the condition that it shall not, by way of trade or otherwise, be lent, re-sold, hired out or otherwise circulated without OpenEye Scientific Software's prior consent, in any form of packaging or cover other than that in which it was produced. No part of this manual or accompanying documentation, may be reproduced, stored in a retrieval system on optical or magnetic disk, tape, CD, DVD or other medium, or transmitted in any form or by any means, electronic, mechanical, photocopying recording or otherwise for any purpose other than for the purchaser's personal use without a legal agreement or other written permission granted by OpenEye.

This product should not be used in the planning, construction, maintenance, operation or use of any nuclear facility nor the flight, navigation or communication of aircraft or ground support equipment. OpenEye Scientific software, shall not be liable, in whole or in part, for any claims arising from such use, including death, bankruptcy or outbreak of war.

Windows is a registered trademark of Microsoft Corporation. Apple, OS X, and Macintosh are registered trademarks of Apple Computer, Inc. AIX and IBM are registered trademarks of International Business Machines Corporation. UNIX is a registered trademark of the Open Group. RedHat is a registered trademark of RedHat, Inc. Linux is a registered trademark of Linus Torvalds. SPARC is a registered trademark of SPARC International Inc.

SYBYL is a registered trademark of TRIPOS, Inc. MDL is a registered trademark and ISIS is a trademark of MDL Information Systems, Inc. SMILES, SMARTS, and SMIRKS may be trademarks of Daylight Chemical Information Systems. Macromodel is a trademark of Schrodinger, Inc.

Python is a trademark of the Python Software Foundation. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the U.S. and other countries.

Other products and software packages referenced in this document are trademarks and registered trademarks of their respective vendors or manufacturers.

INTRODUCTION

The OpenEye Lexichem product contains three applications, **nam2mol**, **mol2nam** and **translate**.

1. The *nam2mol* program is an application for converting compound names into chemical structures. The program currently converts text files containing a single name per line, in either American or British English, into database of molecules, in a choice of file formats, including MDL SD file, SMILES, SLN or Tripos .mol2. This program does not require the input compound name to be the preferred IUPAC name of a compound, and it will work with a variety of traditional names and/or alternate IUPAC forms.
2. The *mol2nam* program is an application to perform the opposite conversion, translating chemical structures into a reasonable compound name.
3. Finally, the *translate* utility program converts chemical names from one language to another.

2.1 Input Name Representation

The *oeiupac* library currently processes NUL (zero) terminated ASCII character strings, therefore Greek characters, symbols, fonts and superscripts must be transliterated into the printable subset of ASCII. When parsing compound names, the *oeiupac* library considers both spaces and tab characters as interchangeable, and any number of consecutive 'whitespace' characters are treated as a single space.

Currently, the name parsing is case insensitive, allowing arbitrary mixing of upper and lower case characters, *e.g.* initial letter capitalization.

Greek characters are understood in a number of different representations. The strings '\$a', '{\$a}', 'alpha', '.alpha.', 'α', '#945;' and '#x3B1;' are all understood to represent the Greek character *alpha*, (α).

There is no special representation for italic characters. Compound names such as '*tert*-butyl' and '*p*-aminobenzamidine' are represented as 'tert-butyl' and 'p-aminobenzamidine'. Both the long and short forms of prefixes can be used, allowing the above examples to also be written as 't-butyl' and 'para-aminobenzamidine'.

2.2 Output Name Representation

Unrecognized functional groups, linkers or ring systems are denoted in the generated name as the string '**BLAH**'. As much of the name possible is generated resulting in compound names such as 'dichloroBLAHcarboxylic acid'. Generated compound names are entirely lower case, with no initial capitalization. Upper case characters are generated for locants and as described above, for BLAH.

When generating Greek characters in compound names, the *oeiupac* library currently uses the dollar character followed by single letter representation. In this formalism, '\$a' represents the Greek character alpha, α , '\$b' the Greek character beta, β , '\$g' the Greek character gamma, γ and '\$l' the Greek character lambda, λ .

When generating superscripts, the *oeiupac* library currently uses the caret and curly braces representation. Hence ‘ λ^5 ’ represents the Greek character lambda followed by a superscript five, *i.e.* λ^5 . Similarly, ‘pentacyclo[4.2.0.0^{2,5}.0^{3,8}.0^{4,7}]octane’ would be the von Baeyer system name for cubane, *i.e.*

pentacyclo [4.2.0.0^{2,5}.0^{3,8}.0^{4,7}] octane.

Multiple components in a disconnected molecule, apart from common salts and counter ions, are separated from each other by a semicolon followed by a space. Mixtures containing salts are written ordering the cations, before the compound name, followed by anions, finally followed by any common neutral molecules (*e.g.* hydrate or hydrochloride).

2.3 Output Name Styles

The *Lexichem* compound naming functionality supports the generation of several *styles* of compound names. The currently predefined name styles are *OpenEye* (the default), IUPAC, CAS, Traditional and Systematic. *OpenEye* names loosely correspond to the kinds of names familiar to a medicinal chemist. These names are intended to be a subset of the IUPAC 2005 standard’s acceptable names, but not necessarily the PIN (Preferred IUPAC Name). These correspond to the types of names found in a Sigma-Aldrich catalog or a Journal of Medicinal Chemistry article for example.

IUPAC names are intended to follow the IUPAC 2005 recommendations for the Preferred IUPAC Name (PIN). Unfortunately, this functionality is relatively recent, so the best that can be hoped for these names is that they are more IUPAC-like than the default *OpenEye* name style. Future releases of *Lexichem* may further refine this definition to provide IUPAC2005, IUPAC93 and IUPAC79 name styles that reflect the corresponding standard’s preferred name.

The *Lexichem* CAS name style is intended to follow the Chemical Abstracts Service’s naming conventions, where they differ from IUPAC’s. Once again, as this functionality is relatively recent, the effect is to generate names that are more CAS-like than the default *OpenEye* name style.

The Traditional name style corresponds to forms of compound naming that are now no longer acceptable to the IUPAC rules. The boundary between whether a trivial/common name is considered *OpenEye* or Traditional when it is acceptable to IUPAC but not preferred is blurred, with *OpenEye* attempting to follow the more prevalent usage.

Finally, Systematic names correspond to the fully systematic IUPAC names that the IUPAC preferred names are slowly converging towards.

2.4 Examples of Name Style Differences

Some of the concepts explained in the previous section are probably best clarified through some real examples.

2.4.1 Example *OpenEye* vs. IUPAC vs. Systematic Differences

The SMILES string O is called ‘water’ by the *OpenEye* name style, but ‘oxidane’ by the IUPAC and Systematic name styles.

The SMILES C#C is called ‘acetylene’ by the *OpenEye* and IUPAC name styles, but ‘ethyne’ by the Systematic name style.

The SMILES prefix *Nc1ccccc1 is called ‘anilino’ by the *OpenEye* and IUPAC name styles, but ‘phenylamino’ by the Systematic name style.

The SMILES prefix *O[N+]#[C-] is called ‘fulminato’ by the *OpenEye* name style, but ‘isocyanooxy’ by the IUPAC and Systematic name styles.

The SMILES prefix *C(=O)C is called ‘acetyl’ in the *OpenEye* and IUPAC name styles, but ‘ethanoyl’ in the Systematic name style.

The SMILES string CC(=O)C is called ‘acetone’ in the *OpenEye* name style, but ‘propan-2-one’ in the IUPAC and Systematic name styles.

The SMILES string C12C3C4C1C5C4C3C25 is called ‘cubane’ in the *OpenEye* name style, but is currently named ‘BLAH’ in the IUPAC and Systematic name styles as we currently fail to name it as the preferred IUPAC2005 PIN: ‘pentacyclo[4.2.0.0^{2,5}.0^{3,8}.0^{4,7}]octane’.

The SMILES string C(=O)O is called ‘formic acid’ in the *OpenEye* and IUPAC name styles, but ‘methanoic acid’ in the Systematic name style.

2.4.2 Example OpenEye/IUPAC vs. CAS Differences

The SMILES string c1ccccc1CCCCC is named as ‘1-phenylheptane’ by the *OpenEye* and IUPAC name styles, but as ‘heptylbenzene’ by the CAS name style.

The SMILES prefix *[BH2] is called ‘boranyl’ by the *OpenEye* and IUPAC name styles, but as ‘boryl’ by the CAS name style.

2.4.3 Example OpenEye/IUPAC vs. Traditional Differences

The SMILES prefix *S is called ‘sulfanyl’ by the *OpenEye* and IUPAC name styles, but as ‘mercapto’ by the Traditional name style.

The SMILES string CCCCCCCC(=O)O is called ‘nonanoic acid’ by the *OpenEye* and IUPAC name styles, but as ‘pelargonic acid’ by the Traditional name style.

INSTALLATION AND PLATFORM NOTES

3.1 Licenses

To run LEXICHEM you will need to obtain a license file for LEXICHEM from OpenEye Scientific Software (business@eyesopen.com). The license file should be in a file pointed to by the **OE_LICENSE** environment variable.

3.2 General Installation

By default, all OpenEye applications are installed into a single distribution directory tree on the specified machine. The default location for this tree is platform specific and will be detailed below.

The root of the tree (i.e. the `openeye` directory) contains the following subdirectories:

- admin** This directory is intended to contain any administrative scripts and tools associated with the installed applications. Currently, this directory is simply a placeholder on all platforms except for Microsoft Windows, where it contains the uninstaller executables.
- arch** This directory contains the collection of platform specific subdirectories. Each subdirectory contains the actual installed executables and support libraries for the associated platform. In the platform specific subdirectory, there will be a subdirectory for each application and within that will be another subdirectory for each version of that application.
- bin** This directory contains a startup script for each application that has been installed. This script determines at run-time what the current platform is and then calls the appropriate executable in the `arch`. This script enables the easy co-existence of multiple platforms and versions of any OpenEye application in the same distribution tree.
- data** This directory contains all of the associated data for the installed applications. There will be a subdirectory for each installed application and within that subdirectory there will be another subdirectory for each specific version of that application.
- docs** This directory contains all of the documentation associated with the installed applications. There will be a subdirectory for each installed application and within that subdirectory there will be another subdirectory for each specific version of that application.
- examples** This directory contains all of the examples associated with the installed applications. There will be a subdirectory for each installed application and within that subdirectory there will be another subdirectory for each specific version of that application.

The startup script discussed in the section on the `bin` directory above will have the same name as the installed executable with which it is associated. When the script is called, it will attempt to determine the current platform and

run the appropriate executable if installed. If an appropriate executable cannot be found, the script will report that information as well as a list of the currently installed platforms. The auto-detection can be overridden by setting one of two environment variables:

- **OE_ARCH** can be used to specify a colon separated list of compatible distributions for the current platform such as:

```
redhat-RHEL5-x64:redhat-RHEL4-x64
```

Specification of this environment variable overrides the auto-detection process if it is present. If none of the compatible distributions listed are found, the script will fall back to the auto-detection process.

- **APPNAME_OE_ARCH** can be used to specify a colon separated list of compatible distributions for a specific application (as specified by changing the **APPNAME** text in the environment variable name) just like **OE_ARCH** as detailed above.

Specification of this environment variable overrides the **OE_ARCH** environment variable as well as the auto-detection process. If none of the compatible distributions listed are found, the script will fall back to the **OE_ARCH** list first and then to the auto-detection process.

Specifying this variable provides a simple way to customize the behavior for individual applications on non-standard platforms.

The startup script also supports a few commandline arguments including:

- | | |
|---------------------|--|
| -path | Specifying this argument will output the full path of the executable to be run. The executable will not be started if this argument is present. |
| -print_arch | Specifying this argument will output the details of the current platform as detected by the script as well as which platform-version of the executable is being run. The executable will be started if this argument is present. |
| -use_version | Specifying this argument followed by a specific version number allows the user to control which released version of the executable to run. |

3.2.1 Linux/Unix

Linux/Unix distributions are provided as a gzipped tarball of the distribution tree described above. Installation is performed by untarring the file in the desired location. Multiple distributions can be installed in the same location without any challenge.

To ensure that the installed applications can be called from the command line, be sure to add the full path of the `openeye/bin` subdirectory to the **PATH** environment variable. For instance, if the distribution was installed into `/usr/local/openeye`, the **PATH** environment variable should contain: `/usr/local/openeye/bin`.

3.2.2 Windows

Windows distributions are provided as a standard EXE installer. By default, all OpenEye applications will install into the `C:\OpenEye` directory.

An OpenEye group with an application specific subgroup will be added to the *Start* menu. The application specific subgroup will contain links to the documentation, the uninstaller, as well as to a Windows command shell which has the appropriate **PATH** settings already defined to allow the user to simply type the executable name at the prompt without concern for where the executable is actually installed.

For GUI applications, a link to the application will be created on the desktop as well as in the application specific subgroup of the *Start* menu.

3.2.3 Mac OS X

Mac OS X distributions are provided as a standard *pkg* installer delivered as a *dmg* disk image. By default, all OpenEye applications will install into the `/Applications/OpenEye` directory.

To ensure that the installed applications can be called from the command line in the *Terminal*, be sure to add `/Applications/OpenEye/bin` to the **PATH** environment variable.

For GUI applications, an application bundle which can be clicked on to start, will be present in the `/Applications/OpenEye` directory. This bundle cannot be moved independent of the `OpenEye` directory. For instance, the entire `OpenEye` directory can be moved as one piece, but moving the application bundle or the contents of any of the subdirectories in the `OpenEye` directory may cause the application to not start. However, the bundle can still be dragged into the Dock and run from there without any problem.

NAM2MOL

OpenEye Scientific Software's *nam2mol* application converts chemical compound names into molecular structures.

4.1 Command Line Interface

A description of the command line interface can be obtained by executing **nam2mol** with no arguments.

```
prompt> nam2mol --help
```

will generate the following output:

```
nam2mol - Name to Structure Conversion
OpenEye Scientific Software
  Version: 2.0.0
  Built: <build date>
  Platform: <platform>
```

Help functions:

```
nam2mol --help simple      : Get a list of simple parameters
nam2mol --help all        : Get a complete list of parameters
nam2mol --help defaults   : List the defaults for all parameters
nam2mol --help <parameter> : Get detailed help on a parameter
nam2mol --help html       : Create an html help file for this program
```

4.1.1 Required Parameters

-in <filename>

The input file on the command line is assumed to be compound name file in ASCII text format, and the optional output filename is treated as the output molecule file. If no output file is specified, molecules will be written to stdout in SMILES format.

4.2 Command Line Options

-out <filename>

The file format of the output file is automatically determined from the file extension. The extensions *.smi*, *.can* and *.ism* may be used to specify SMILES format; *.sdf*, *.mdl* and *.mol* can be used to specify MDL connection

table file formats; *.oeb* for OEBinary; *.mmod* for Macromodel; *.sln* for Sybyl line notation; *.mol2* for Tripos *.mol2* files; *.pdb* for PDB format files, *etc...*

File type	Extension
SMILES	.smi .ism .can
SDF	.sdf .mol .sdf.gz .mol.gz
MOL2	.mol2 .mol2.gz
PDB	.pdb .ent .pdb.gz .ent.gz
MacroModel	.mmod .mmod.gz
OEBinary	.oeb .oeb.gz

-language <lang>

Parse the input file based on the specified language. The default is English. Can use either the full name or synonym. `-language dutch` and `-language nl` are equivalent.

Language	Option	Synonyms	Language	Option	Synonyms
English	american	english us	Italian	italian	it
British	british	uk	Japanese	japanese	jp ja
Chinese	chinese	zh cn	Polish	polish	pl
Danish	danish	dk da	Portuguese	portuguese	pt
Dutch	dutch	nl	Romanian	romanian	ro
French	french	fr	Russian	russian	ru
German	german	de	Slovak	slovak	sk
Greek	greek	el	Spanish	spanish	es
Hungarian	hungarian	hu	Swedish	swedish	se sv
Irish	irish	ie ga	Welsh	welsh	cy

[default = american]

-empty

Write an empty connection table whenever unable to parse an input line. This function is useful for keeping track of which names were converted, as the output file is guaranteed to have the same number of connection tables as there were lines in the input file.

-dots

Use dots to show program progress.

-depict

When writing output files that contain co-ordinates, this command line option can be used to generate suitable 2D co-ordinate depictions, provided that the OpenEye *ogham* toolkit is appropriately licensed.

-tag

Specify the tag field name to be used when writing MDL SD files. Normally, the original name is recorded in the title field of each output connection table. However, for MDL SD files the title is truncated to a maximum of 80 characters. This option allows the full name to additionally be written as a field to the file. For example, `-tag name` writes the name in the `<name>` data field.

MOL2NAM

OpenEye Scientific Software's *mol2nam* application converts molecular structures into reasonable chemical names.

5.1 Command Line Interface

A description of the command line interface can be obtained by executing **mol2nam** with the *-help* option.

```
prompt> mol2nam --help
```

will generate the following output:

```
mol2nam - Structure to Name Conversion
OpenEye Scientific Software
  Version: 2.0.0
  Built: <build date>
  Platform: <platform>
```

Help functions:

```
mol2nam --help simple      : Get a list of simple parameters
mol2nam --help all         : Get a complete list of parameters
mol2nam --help defaults    : List the defaults for all parameters
mol2nam --help <parameter> : Get detailed help on a parameter
mol2nam --help html        : Create an html help file for this program
```

5.1.1 Required Parameters

-in <filename>

The input file can be in any of a number of popular connection table formats. If no output file is specified, the program writes a name per line, for each connection table in the input file to standard output, stdout. If an output file is specified with *-out*, it is treated as the output molecule file, and each of the input molecules is written to it, with the title of each record set to the assigned name.

The file format of the input file is automatically determined from the file extension. The extensions *.smi*, *.can* and *.ism* may be used to specify SMILES format; *.sdf*, *.mdl* and *.mol* can be used to specify MDL connection table file formats; *.oeb* for OEBinary; *.mmod* for Macromodel; *.sln* for Sybyl line notation; *.mol2* for Tripos *.mol2* files; *.pdb* for PDB format files; *etc...*

File type	Extension
SMILES	.smi .ism .can
SDF	.sdf .mol .sdf.gz .mol.gz
MOL2	.mol2 .mol2.gz
PDB	.pdb .ent .pdb.gz .ent.gz
MacroModel	.mmod .mmod.gz
OEBinary	.oeb .oeb.gz

To read from stdin (via a pipe), use the file extension that corresponds to the input stream. For example, to pipe in SMILES, use: `-in .smi`.

5.2 Command Line Options

-out <filename>

Output molecule file. The title of each molecule will be set to the generated name. Output formats are the same as available for `-in`.

-style <name style>

Determines the name style of generated names. By default, **mol2nam** uses the *OpenEye* name style. Styles include:

autonom Attempt to generate MDL/Beilstein AutoNom-like names. MDL's AutoNom normally generates capitalized names, which can be controlled via the `-capitalize` command line option.

cas Attempt to generate CAS-like names, as used by the Chemical Abstracts Service (CAS).

casidx Attempt to generate CAS permuted index-like names, as used by the Chemical Abstracts Service (CAS).

iupac Attempt to generate the Preferred IUPAC Name (PIN) of a compound as defined by the IUPAC200x standard.

iupac79 Attempt to generate an IUPAC 1979-style name.

iupac93 Attempt to generate an IUPAC 1993-style name.

openeye Attempt to generate an OpenEye-style name. [default]

traditional Attempt to generate traditional, common or archaic names for a compound.

systematic Attempt to generate (fully) systematic names.

[default = openeye]

-capitalize

Capitalize the appropriate letter of the generated name.

[default = false]

-language <lang>

Determines the output language of generated names. The default is English. Can use either the full name or synonym. `-language dutch` and `-language nl` are equivalent.

Language	Option	Synonyms	Language	Option	Synonyms
English	american	english us	Italian	italian	it
British	british	uk	Japanese	japanese	jp ja
Chinese	chinese	zh cn	Polish	polish	pl
Danish	danish	dk da	Portuguese	portuguese	pt
Dutch	dutch	nl	Romanian	romanian	ro
French	french	fr	Russian	russian	ru
German	german	de	Slovak	slovak	sk
Greek	greek	el	Spanish	spanish	es
Hungarian	hungarian	hu	Swedish	swedish	se sv
Irish	irish	ie ga	Welsh	welsh	cy

[default = american]

-charset <charset>
-encoding <charset>

ascii Encode the output using ASCII.

eucjp Encode the output using EUC-JP to represent Japanese characters. This is normally used in conjunction with the *-language japanese* command line option.

html Encode the output using HTML markup to represent Greek characters, foreign characters and superscripts.

sjis Encode the output using Shift-JIS to represent Japanese characters. This is normally used in conjunction with the *-language japanese* command line option.

utf8 Encode the output using UTF-8.

[default = ascii]

-delim

By default, the connection tables written to the output file have their title replaced with the generated compound name. However, if the *-delim* option is given followed by a delimiter string, the name is appended to the original title separated by the specified delimiter.

-tag

When the output file is to be written in MDL SD file format, also write the compound name in the specified data tag.

TRANSLATE

OpenEye Scientific Software's *translate* utility converts chemical names from one language to another.

6.1 Command Line Interface

A description of the command line interface can be obtained by executing **translate** with the *-help* option.

```
prompt> translate --help
```

will generate the following output:

```
translate - Structure to Name Conversion
OpenEye Scientific Software
  Version: 2.0.0
  Built: <build date>
  Platform: <platform>
```

Help functions:

```
translate --help simple      : Get a list of simple parameters
translate --help all         : Get a complete list of parameters
translate --help defaults    : List the defaults for all parameters
translate --help <parameter> : Get detailed help on a parameter
translate --help html        : Create an html help file for this program
```

6.1.1 Required Parameters

-in <filename>

The input file on the command line is assumed to be compound name file in text format, and the optional output filename *-out* is treated as the output molecule file. If no output file is specified, translated names will be written to stdout.

6.2 Command Line Options

-from <language>

-from_language <language>

Parse the input file based on the specified language. The default is English. Can use either the full name or synonym. *-from dutch* and *-from nl* are equivalent.

Language	Option	Synonyms	Language	Option	Synonyms
English	american	english us	Italian	italian	it
British	british	uk	Japanese	japanese	jp ja
Chinese	chinese	zh cn	Polish	polish	pl
Danish	danish	dk da	Portuguese	portuguese	pt
Dutch	dutch	nl	Romanian	romanian	ro
French	french	fr	Russian	russian	ru
German	german	de	Slovak	slovak	sk
Greek	greek	el	Spanish	spanish	es
Hungarian	hungarian	hu	Swedish	swedish	se sv
Irish	irish	ie ga	Welsh	welsh	cy

-to <language>

-to_language <language>

Generate output based on the specified language. The default is English. Can use either the full name or synonym. `-to dutch` and `-to nl` are equivalent. Language choices are shown in the table above.

-out <filename>

Output text file for names. If not specified, names are written to *stdout*.

-charset <charset>

-encoding <charset>

ascii Encode the output using ASCII.

eucjp Encode the output using EUC-JP to represent Japanese characters. This is normally used in conjunction with the `-language japanese` command line option.

html Encode the output using HTML markup to represent Greek characters, foreign characters and superscripts.

sjis Encode the output using Shift-JIS to represent Japanese characters. This is normally used in conjunction with the `-language japanese` command line option.

utf8 Encode the output using UTF-8.

[default = ascii]

RELEASE NOTES

7.1 Lexichem 2.0.0

- The applications have a new, standardized command line interface. Please have a look at the updated documentation for *mol2nam*, *nam2mol* and *translate*.
- This release includes the ability to parse stereo on input names. Previously it was read and ignored.
- Fixed a bug where, in rare cases, the output name depended on input atom ordering.
- Fixed a crash in determining CIP stereo for very large, pathological molecules.

7.2 Lexichem 1.9

- On a benchmark of 250251 compounds in the NCI00 database, *mol2nam* is able to convert 234297 structures (93.62%) to names without BLAH. Of these 234297 names, *nam2mol* is able to convert 231566 (98.83%) back into structures.
- This release includes a significant number of improvements to both name generation and name parsing. Several bugs have also been fixed. The name parsing conversion rate for the 71367 compound names in the 2003 Maybridge catalog is now up to 95.24%.
- Several improvements have been made to the specification of CIP stereochemistry during name generation. For example, previously linking groups such as *amidino*, *carbamimidoyl* and *diazenyl* would forget to specify E/Z descriptors if they contained a chiral double bond with specified stereochemistry. We would also fail to place some chiral prefixes such as (E)-*styr1* and (Z)-*cinnamyl* in brackets which can lead to ambiguity when interpreting the generated name.

7.3 Lexichem 1.8

- On a benchmark of 250251 compounds in the NCI00 database, *mol2nam* is able to convert 234296 structures (93.62%) to names without BLAH. Of these 234296 names, *nam2mol* is able to convert 228102 (97.36%) back into structures.
- This release includes a significant number of improvements to both name generation and name parsing. Several bugs have also been fixed. The name parsing conversion rate for the 71367 compound names in the 2003 Maybridge catalog is now up to 95.12%.
- One of the major parsing improvements in this release is the much improved support for handling von Baeyer ring nomenclature. We can now parse names such as:

- '1,4-dithioniabicyclo[2.2.2]octane',
- 'bicyclo[4.2.0]octa-1(6),2,4-triene' and
- '2,4-diazaspiro[4.4]nonane-1,3-dione'.

7.4 Lexichem 1.7

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 234155 structures (93.57%) to names without BLAH. Of these 234155 names, `nam2mol` is able to convert 223246 (95.34%) back into structures.
- This release includes a significant number of improvements to both name generation and name parsing. Several bugs have also been fixed. The name parsing conversion rate for the 71367 compound names in the 2003 Maybridge catalog is now up to 93.81%.
- A new function has been added to the *Lexichem* toolkit API. This function converts the input chemical name to lower-case, whilst preserving the case sensitive aspects of IUPAC names. This functionality allows uppercase and mixed case names to be translated into English, as the `OEFROM<FOO>` functions assume their input is lowercase. For example, this feature allows AGUA to be recognized via `.`
- A new function has been added to the *Lexichem* toolkit API. This function attempts to reorder the given permuted index name into a form that can be handled by the function. For example, this will convert the string 'benzene, chloro-' into 'chloro-benzene'.
- A number of improvements and bug fixes have been made to *Lexichem's* naming styles. For example, AutoNom and CAS permuted index styles are now far more AutoNom-like and CAS-like respectively. Naming of metallocenes and fullerenes is much improved.
- Some dramatic improvements have been made with foreign language support. On the 250251 compounds in the NCI00 database mentioned above, we now round-trip 100% to German and back without any differences. Japanese, Spanish and Swedish rates are all currently above 99%. Support for Hungarian and Polish has been dramatically improved.

7.5 Lexichem 1.6

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 233010 structures (93.11%) to names without BLAH. Of these 233010 names, `nam2mol` is able to convert 221331 (94.99%) back into structures.
- This release includes a significant number of improvements to both name generation and name parsing. For example, both name generation and parsing now do a much better job on ring fusion nomenclature, for names like '5,6,7,8-tetrahydro[1,2,4]triazolo[4,3-a]pyridine'. There's also much improved handling of charged ring systems. The name parsing conversion rate for the 71367 compound names in the 2003 Maybridge catalog is now 93.25% in v1.6, up from 80.80% in v1.5.
- In name generation, new naming styles have been added for MDL/Beilstein AutoNom style names, for CAS permuted index style names (and there are new placeholder styles for IUPAC79 and IUPAC93 naming). A large number of improvements have been made to names generated using the 'traditional' naming style. A new API function is available to capitalizing the appropriate first letter of a generated name, such as 'p-tert-Butylbenzoic acid'.
- Several bug fixes have been made to the Cahn-Ingold-Prelog (CIP) chirality perception implementation.

- The function is now able return supplementary locant annotations for each atom. This function now stores an integer locant code/identifier in the integer atom type field of each atom, which may be retrieved using the method and converted into a readable/displayable string using the recently exposed function. This functionality is a recent addition (obviously), and most but not all supported ring systems and parents have locant annotations in this initial release.
- Finally, for the adventurous, new APIs for translating compound names from foreign languages into English are available as the experimental , and functions. Additionally, a function is available for converting UTF-8 encoded strings into the escaped sequences expected by these functions (effectively the inverse of).

7.6 Lexichem 1.5

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 223066 structures (89.14%) to names without BLAH. Of these 223066 names, `nam2mol` is able to convert 192487 (86.29%) back into structures.
- This release includes a significant number of improvements to both name generation and name parsing. For example, `nam2mol` now supports more numbered locants, such as ‘N1-methylaniline’ and for ‘Maybridge-style’ locant names such as N’ 1 (interpreted as the more common N1’). These and similar changes have increased the conversion rate for the 71367 compound names in the 2003 Maybridge catalog, from 69.51% in v1.4 to 80.80% in v1.5.
- This release includes the ability to generate compound names in Japanese, and much improved Spanish and Polish naming support. In order to better support internationalization, APIs are now available to map from the default ISO-8859-1 output to either 7-bit ASCII, UTF-8, HTML and for Japanese locales, Shift-JIS or EUC-JP.
- Although impossible in the general case, several improvements have been made to *Lexichem*’s compound naming such that the assigned names are now more stable under arbitrary input ordering of atoms and bonds.

7.7 Lexichem 1.4

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 221254 structures (88.41%) to names without BLAH. Of these 221254 names, `nam2mol` is able to convert 192345 (86.93%) back into structures.
- *Lexichem* v1.4 is predominantly a maintenance to provide a version of the *oeiupac* library that is compatible with *OEChem* v1.4. However, there have been a number of significant improvements to name parsing, and minor improvements to name generation since last month’s v1.3 release.
- This release also includes the ability to generate compound names in several languages. In addition, to British spellings, *Lexichem* can now generate German, Italian, French, Spanish, Swedish, Dutch and Polish names. Whilst the translations for German, Italian, Swedish and Polish are quite comprehensive, those for French, Spanish and Dutch are less complete.
- A potential ambiguity with the ring names ‘oxazole’ and ‘thiazole’ has also been resolved. The IUPAC documentation states that it is permissible to omit locants from Hantzsch-Widman names when the locants are consecutive, *i.e.* ‘1,2,3,4-tetrazole’ may be written as ‘tetrazole’, and ‘1,2-oxazirene’ is preferred as ‘oxazirene’. Unfortunately, this conflicts with the traditional interpretations of ‘oxazole’ as meaning ‘1,3-oxazole’ and ‘thiazole’ as ‘1,3-thiazole’. Instead the traditional names ‘isoxazole’ and ‘isothiazole’ denote the ‘1,2-’ forms. This ambiguity, that affected IUPAC-style (but not OpenEye-style) names, has been resolved by preserving the locants, so that the IUPAC names ‘1,2-oxazole’, ‘1,3-oxazole’, ‘1,2-thiazole’ and ‘1,3-thiazole’ are now generated for ‘isoxazole’, ‘oxazole’, ‘isothiazole’ and ‘thiazole’ respectively.

7.8 Lexichem 1.3

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 221205 structures (88.39%) to names without BLAH. Of these 221205 names, `nam2mol` is able to convert 183444 (82.93%) back into structures.
- The major announcement of this release is the support for stereochemistry in compound naming. The CIP rules for assigning R/S descriptors to tetrahedral chiral centers, and E/Z descriptors to double bonds are used during name generation.

7.9 Lexichem 1.2

On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 220949 structures (88.29%) to names without BLAH. Of these 220949 names, `nam2mol` is able to convert 182438 (82.57%) back into structures.

7.10 Lexichem 1.2

On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 220949 structures (88.29%) to names without BLAH. Of these 220949 names, `nam2mol` is able to convert 182438 (82.57%) back into structures.

7.11 Lexichem 1.1

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 220924 structures (88.28%) to names without BLAH. Of these 220924 names, `nam2mol` is able to convert 177145 (80.18%) back into structures.
- A new API has been added so allow applications to check whether *Lexichem*'s parsing and naming functionality can safely be used.

7.11.1 OEParseIUPACName Improvements

The *Lexichem* name parsing routines now handle a small number of structural abbreviations when parsing names. For example, it can now handle names like '3-CF3-5-NO2-benzoic acid'. The usual improvements in name parsing, including more entries for common names in the Lexichem dictionary. Support for names containing multiple explicit hydrogen locants, such as 'pyrimidine-2,4(1H,3H)-dione' and '2,4(1H,3H)-pyrimidinedione'.

7.11.2 OECreatelUPACName Improvements

A serious bug that could cause a core dump when naming thioperoxoic acids has been fixed. The performance of compound naming has been improved. The usual improvements in the names generated (following the IUPAC standards more closely).

7.12 Lexichem 1.0

On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 220922 structures (88.28%) to names without BLAH. Of these 220922 names, `nam2mol` is able to convert 177032 (80.13%) back into structures.

7.12.1 OEParseIUPACName Improvements

In addition to a great many other improvements to the name parsing code, the *Lexichem* parser now contains an internal dictionary allowing the recognition of common non-systematic names, such as ‘ranitidine’ and ‘zantac’.

7.12.2 OECreatelUPACName Improvements

In addition to a great many improvements to the name generation code, the *Lexichem* naming functionality now allows the specification of a naming style, allowing the compound to be named in either a traditional, OpenEye, IUPAC, CAS or systematic naming style.

7.13 Lexichem 1.0.b3

On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 220854 structures (88.25%) to names without BLAH. Of these 220854 names, `nam2mol` is able to convert 144821 (65.57%) back into structures.

7.14 Lexichem 1.0.b2

- On a benchmark of 250251 compounds in the NCI00 database, `mol2nam` is able to convert 208378 structures (83.27%) to names without BLAH. Of these 208378 names, `nam2mol` is able to convert 139100 (66.75%) back into structures.

7.14.1 OEParseIUPACName Improvements

- Support for parsing substituents (name fragments) by generating a wildcard atom in SMILES, for example, ‘methyl’ returns ‘*C’ and ‘hydroxy’ returns ‘*O’.
- Adds support for parenthesized indicated hydrogens, *e.g.* ‘naphthalen-1(2H)-one’.
- Tweaks to support names, such as ‘9,10-anthracene-dicarboxylic acid’, where the hyphen after the stem would cause the parser to expect a list of locants.
- Fixes to multipliers of ‘oxy’ (and related) linkers, so that ‘dineopentyloxybenzene’ is correctly considered two copies of the prefix ‘neopentoxy’.
- Added support for ‘...ic aldehyde’ and ‘...ic acid aldehyde’ as synonyms of ‘...aldehyde’. Added support for both ‘yl’ and ‘oyl’ variants of traditional acid stems, *i.e.* both ‘crottonoyl’ and ‘crotonyl’, and both ‘acryloyl’ and ‘acrylyl’. Adds support for ‘...amido’ as a traditional linker, *e.g.* ‘propanamido’, ‘acrylamido’ and ‘benzamido’.
- Added support for traditional bivalent imides, *e.g.* ‘succinimide’, including N-substituted forms, *e.g.* ‘N-iodosuccinimide’.
- Fixes parsing of salt multipliers, *e.g.* ‘benzene trihydrate’.

- Fixes nitrogenous linker processing, *e.g.* ‘N,N-dimethylsulfamoyl chloride’.
- Adds support for locants alpha, beta and gamma (as written in English), allowing us to handle ‘alpha,alpha,alpha-trichlorotoluene’ and ‘beta-mercaptoethanol’.
- Improvements to handle unspecified locants, *e.g.* ‘chlorophenyl’ and ‘pentachlorophenyl’.
- Support for substitutions on ‘benzoyl’ and ‘benzamido’ prefixes, *e.g.* ‘4-nitrobenzoyl chloride’.
- Handling of names using di, tri etc. incorrectly when bis, tris etc. should have been used, *e.g.* ‘2,4-di(*t*-butylperoxy)hexane’.
- Improvements to ring locant numbers on ‘benzophenone’ and ‘benzidine’, and support for primed locants on ring system parents.
- Adds N and N’ locants to the hydrazine stem, *e.g.* ‘4-(N’,N’-dihydroxyhydrazino)benzoic acid’.
- Treat ‘imidamide’ as a synonym for ‘amidine’, and ‘hydrazinyl’ as a synonym for ‘hydrazino’.
- Adds support for the element ‘columbium’ (more commonly known as ‘niobium’).
- Adds support for the prefixes ‘keto’ (synonym for ‘oxo’), ‘allophanoyl’, ‘hydantoyl’, ‘ureido’, ‘carbamido’, ‘lauryl’, ‘myrsityl’, ‘palmityl’, ‘stearyl’, ‘carbamimidoyl’, ‘sulfinamoyl’, ‘thiocarbamoyl’, ‘carbamothioyl’, ‘carbamoyl’, ‘guanyl’, ‘morpholino’, ‘oxycyano’, ‘sulfinyl’, ‘sulfonyl’ and ‘fulminato’.
- Added support for ‘acetonyl’ and ‘phenacyl’ as vinyl-like prefixes, including their ‘.idene’ and ‘.idyne’ variants.
- Added support for the traditional stems ‘vanillic acid’, ‘isovanillic acid’, ‘syringic acid’, ‘arachidic acid’, ‘benhenic acid’, ‘carboceric acid’, ‘cerinic acid’, ‘ceromelissic acid’, ‘ceroplastic acid’, ‘cerotic acid’, ‘daturic acid’, ‘enanthic acid’, ‘geddic acid’, ‘gheddic acid’, ‘japanic acid’, ‘lacceric acid’, ‘lignoceric acid’, ‘margaric acid’, ‘melissic acid’, ‘montanic acid’, ‘pelargonic acid’, ‘psylic acid’, ‘thapsic acid’, ‘brassylic acid’ and ‘pyruvic acid’.
- Added support for the stems ‘ketene’, ‘thioketene’, ‘vanillin’, ‘isovanillin’, ‘rhodanine’, ‘allophanic acid’, ‘hydantonic acid’, ‘picoline’, ‘borate’ [BH₄-], ‘fulvene’, ‘isobutene’, ‘isoprene’, ‘alloxane’, ‘barbituric acid’, ‘hydantoin’, ‘cytosine’, ‘guanine’, ‘hydroxylamine’ and the common amino acids.
- Adds support for the ring systems ‘benzimidazole’, ‘benzoimidazole’, ‘benzothiophene’, ‘benzoxazole’, ‘benzooxazole’, ‘benzothiazole’, ‘benzotriazole’, ‘benzotrioxazole’, ‘pyrene’, ‘perylene’, ‘as-indacene’, ‘s-indacene’, ‘pyrrolizine’ and ‘quinolizine’.
- Added support for the ring suffix ‘carbonyl chloride’ (and other acid halides).
- Added support for the suffixes ‘thioketone’, ‘sulfide’, ‘nitrite’, ‘azanium’, ‘thial’, ‘diazonium’, ‘arsonic acid’, ‘peroxoic acid’, ‘carboperoxoic acid’, ‘carbothioamide’, ‘carboximidamide’, ‘carboxamidine’, ‘aldehyde oxime’ ‘one oxime’.
- Added support for the linkers ‘mercuri’, ‘carbamimidoyl’, ‘sulfinamoyl’, ‘sulfamoyl’, ‘thiocarbamoyl’, ‘carbamothioyl’, ‘carbonimidoyl’, and ‘thioyl’ (as in ‘ethanethioylbenzene’).
- Added support for the salts ‘hydrobromide’, ‘hydrofluoride’, ‘hydroiodide’, ‘triiodide’, ‘hydrotriiodide’, ‘sulfite’, ‘peroxide’, ‘perchlorate’, ‘perbromate’, ‘periodate’, ‘hydrate’, ‘nitrite’, ‘hypochlorite’, ‘chlorite’, ‘chlorate’, ‘bromate’, ‘iodate’, ‘nitrate’, ‘cyanide’ and ‘cyanate’ (including iso and thio variants).
- Added support for ‘hydrochloric acid’, ‘hydrobromic acid’, ‘hydrofluoric acid’, ‘hydroiodic acid’, ‘hydrotriiodic acid’, and ‘tetric acid’.

7.14.2 OECreatelUPACName Improvements

- Added support for spiro, bicyclo and large simple heterocycle naming, all with hetero replacement nomenclature. No longer elide ring system prefix locants, which fixes the ambiguity with ‘N-tetralinylacetamide’ which should be ‘N-tetralin-1-ylacetamide’.
- Improvements to indicated hydrogen perception (a nitrogen with three ring bonds doesn’t require an indicated hydrogen), improving our naming of ‘indolizine’, ‘pyrrolizine’ and ‘quionlazine’.
- Improvements to naming cycloalkane rings attached via an oxygen linker, *i.e.* ‘cyclopropoxy’ instead of ‘cyclopropyloxy’.
- Improvements in the handling of substituted linkers, ‘carbamoyl’, ‘thiocarbamoyl’, ‘carbamidoyl’, ‘sulfamoyl’ and ‘sulfnamoyl’.
- Improvements to alkyl chain termination, for example ‘carboxymethyl’ instead of ‘2-hydroxy-2-oxo-ethyl’ and ‘cyanomethyl’ instead of ‘nitridoethyl’.
- Improvements to amide, thioamide and sulfonamide atom typing to allow ‘*C(=O)N=C*’ to be considered an N-substituted amide.
- Added support for more phosphane variants; ‘*=P-*’, ‘*=PH’ and ‘*#P’.
- Add ‘oxamide’ as a contraction of ‘oxalamide’.
- Fixes in multiple suffix processing, *e.g.* ‘cyclohexane-1,4-diamine’.
- Fixes to the prefixes ‘acryloyl’ and ‘proprioloyl’ (and their acid halides) which were previously incorrectly named ‘acrylyl’ and ‘propiolyl’.
- Fixed handling of atom typed metals, fixing ‘trichloromagnesium’.
- Corrected names for bivalent acid halides, *e.g.* ‘malonyl dichloride’.
- Made ‘benzoyl chloride’ the preferred name for ‘benzenecarbonyl chloride’ and likewise for other benzoic acid halides.
- Fix spelling typos in ‘acenaphthene’ and ‘isothiazolidine’.
- Added support for the parents ‘heptalene’, ‘octalene’, ‘hydroxylamine’, ‘hexahydropyrimidine’ (formerly ‘1,2-diazinane’), and ‘hexahydropyridazine’ (formerly ‘1,3-diazinane’).
- Added support for the prefixes ‘acetyl’, ‘aceonylidene’, ‘phenacyl’, ‘phenacylidene’, ‘ureido’, ‘anilino’, ‘hydantoyl’, ‘allophanoyl’, ‘amidino’, ‘acetoxy’, ‘isopropoxy’, ‘isobutoxy’, ‘sec-butoxy’, ‘tert-butoxy’, ‘morpholino’ (formerly ‘morpholin-4-yl’), ‘oxycyano’, ‘sulfinyl’ (*S=O), ‘sulfonyl’ (*S(=O)=O), and ‘methylene’ (formerly ‘methylidene’).
- Added support for the linkers ‘carbonimidoyl’, ‘mercuri’ and ‘benzoyl’ (replacing ‘phenylcarbonyl’).
- Add support for the suffixes ‘amidine’, ‘ohydrazide’, ‘carbohydrazide’, ‘...one oxime’, ‘...al oxime’, ‘aldehyde oxime’ and ‘nitrile oxide’ (note we consider both ‘*C#N=O’ and ‘*C#[N+][O-]’ nitrile oxides).
- Added support for the salts ‘triiodide’, ‘hydrotriiodide’, ‘hydrogen triiodide’, ‘perbromate’, ‘periodate’, ‘sulfide’, ‘sulfite’, ‘peroxide’, ‘iodate’, ‘nitrite’ and ‘hypochlorite’.
- Added support for the molecule ‘fulvene’.

7.15 Lexichem 1.0.b1

On a benchmark of 250,251 compounds in the NCI00 database, `mol2nam` is able to convert 201004 structures (80.32%) to names without BLAH. Of these 201004 names, `nam2mol` is able to convert 111442 (55.44%) back into structures.

7.16 Indices and tables

- *Index*
- *Search Page*

BIBLIOGRAPHY

- [Brecher-1999] J. Brecher, **Name=Struct: A Practical Approach to the Sorry State of Real-Life Chemical Nomenclature**, *Journal of Chemical Information and Computer Science*, Vol. 39, pp. 943–950, 1999.
- [Bünzli-Trepp-2007] Ursula Bünzli-Trepp, **Systematic Nomenclature of Organic, Organometallic and Coordination Chemistry: Chemical-Abstracts Guidelines with IUPAC Recommendations and Many Trivial Names**, EPFL Press, April 2007.
- [Cahn-1966] R.S. Cahn, C.K. Ingold and V. Prelog, **Specification of Molecular Chirality**, *Angew. Chem. Int. Ed. Engl.*, Vol. 5, pp. 385–414, 1966. Errata Vol. 5, p. 511, 1966.
- [Gernot-2006] Gernot A. Eller, **Improving the Quality of Published Chemical Names with Nomenclature Software**, *Molecules*, Vol. 11, pp. 915–928, 2006.
- [Robert-2001] Robert B. Fox and Warren H. Powell, **Nomenclature of Organic Compounds: Principles and Practice**, Oxford University Press publishers, 2001.
- [Garfield-1961] E. Garfield, **Chemico-linguistics: Computer Translation of Chemical Nomenclature**, *Nature*, Vol. 192, pp. 192–194, 1961.
- [Goebels-1991] L. Goebels, A.J. Lawson and J.L. Wisniewski, **AUTONOM: System for Computer Translation of Structural Diagrams into IUPAC-Compatible Names: 2. Nomenclature of Chains and Rings**, *Journal of Chemical Information and Computer Science*, Vol. 31, pp. 216–225, 1991.
- [Hellwinkel-2006] D. Hellwinkel, **Die Systematische Nomenklatur der Organischen Chemie: Eine Gebrauchsanweisung**, Springer publishers, 2006 (German).
- [Hellwinkel-2001] D. Hellwinkel, **Systematic Nomenclature of Organic Chemistry: A Directory to Comprehension and Application of its Basic Principles**, Springer-Verlag publishers, 2001.
- [Nyitrai-1998] József Nyitrai and József Nagy, **Útmutató a szerves vegyületek IUPAC-nevezéktanához**, Magyar Kémikusok Egyesülete, Budapest, 1998 (Hungarian).
- [Labute-1996] Paul Labute, **An Efficient Algorithm for the Determination of Topological RS Chirality**, *Journal of the Chemical Computing Group*, On-line, November 1996.
- [Leigh-2001] G.J. Leigh, H.A. Favre and W.V. Metanomski, **Principes de Nomenclature de la Chemie: Introduction aux recommandations de l'IUPAC**, DeBroeck Université publishers, 2001 (French).
- [Levine-1992] John R. Levine, Tom Mason and Doug Brown, **lex & yacc**, 2nd Edition, UNIX Programming Tools, O'Reilly and Associates publishers, 1992.
- [Mitchell-1948] A.D. Mitchell, **British Chemical Nomenclature**, Edward Arnold & Co. publishers, London, 1948.

- [Peterson-1987] W.R. Peterson, **Formulacion Y Nomenclatura Quimica Organica**, 15th Edition, EDUNSA publishers, Barcelona, **1993**. (Spanish).
- [Polskie-1992] Polskie Towarzystwo Chemiczne, **Nomenklatura Zwiasków Organicznych**, Panstwowe Wydawnictwo Naukowe publishers, Warsaw, **1992**. (Polish).
- [Polskie-1994] Polskie Towarzystwo Chemiczne, **Przewodnik Do Nomenklatury Zwiasków Organicznych**, Narodowy Komitet Miedzynarodowej Unii Chemii Czystej I Stosowanej publishers, Warsaw, **1994**. (Polish).
- [Prelog-1982] V. Prelog and G. Helmchen, **Basic Principles of the CIP-System and Proposals for a Revision**, *Angew. Chem. Int. Ed. Engl.*, Vol. 21, pp. 567–583, **1982**.
- [Sayle-2009] Roger Sayle, **Foreign Language Translation of Chemical Nomenclature by Computer**, *Journal of Chemical Information and Modeling*, Vol. 49, pp. 519–530, **2009** (online: <http://pubs.acs.org/doi/abs/10.1021/ci800243w>)
- [Thurlow-2006] K.J. Thurlow, **Chemical Nomenclature**, Kluwer Academic Publishers, September **1998**.
- [Unicode-2006] The Unicode Consortium, **The Unicode Standard, Version 5.0**, Fifth Edition, Addison-Wesley Professional publishers, October **2006**.
- [Wikman-2004] Susanne Wikman, **Organisk-Kemisk Nomenklatur**, Studentlitteratur Publishers, Lund, **2004**. (Swedish).
- [Williams-1990] Anthony Williams and Andrey Yerin, **The Need for Systematic Naming Software Tools for Exchange of Chemical Information**, *Molecules*, Vol. 4, pp. 255–263, **1999**.
- [Wisniewski-1990] J.L. Wisniewski, **AUTONOM: System for Computer Translation of Structural Diagrams into IUPAC-Compatible Names: 1. General Design**, *Journal of Chemical Information and Computer Science*, Vol. 30, pp. 324–332, **1990**.

INDEX

Symbols

- capitalize
 - mol2nam command line option, 14
- charset <charset>
 - mol2nam command line option, 15
 - translate command line option, 18
- delim
 - mol2nam command line option, 15
- depict
 - nam2mol command line option, 12
- dots
 - nam2mol command line option, 12
- empty
 - nam2mol command line option, 12
- encoding <charset>
 - mol2nam command line option, 15
 - translate command line option, 18
- from <language>
 - translate command line option, 17
- from_language <language>
 - translate command line option, 17
- in <filename>
 - mol2nam command line option, 13
 - nam2mol command line option, 11
 - translate command line option, 17
- language <lang>
 - mol2nam command line option, 14
 - nam2mol command line option, 12
- out <filename>
 - mol2nam command line option, 14
 - nam2mol command line option, 11
 - translate command line option, 18
- style <name style>
 - mol2nam command line option, 14
- tag
 - mol2nam command line option, 15
 - nam2mol command line option, 12
- to <language>
 - translate command line option, 18
- to_language <language>
 - translate command line option, 18

A

APPNAME_OE_ARCH, 8

E

environment variable

- APPNAME_OE_ARCH, 8
- OE_ARCH, 8
- OE_LICENSE, 7
- PATH, 8, 9

M

mol2nam command line option

- capitalize, 14
- charset <charset>, 15
- delim, 15
- encoding <charset>, 15
- in <filename>, 13
- language <lang>, 14
- out <filename>, 14
- style <name style>, 14
- tag, 15

N

nam2mol command line option

- depict, 12
- dots, 12
- empty, 12
- in <filename>, 11
- language <lang>, 12
- out <filename>, 11
- tag, 12

O

- OE_ARCH, 8
- OE_LICENSE, 7

P

PATH, 8, 9

T

translate command line option

- charset <charset>, 18
- encoding <charset>, 18
- from <language>, 17
- from_language <language>, 17
- in <filename>, 17
- out <filename>, 18
- to <language>, 18
- to_language <language>, 18