IUPAC International Chemical Identifier (InChI)

InChI version 1, Software version 1.05

Release Notes

Last revision date:January 31, 2017

This document is a part of the release of the IUPAC International Chemical Identifier with InChIKey, version 1, software version 1.05.

CONTENTS

Summary

This document introduces the release of InChI Software version 1.05 (Winter 2017).

The current version of InChI Identifier is 1; the current status of the InChI software is 1.05 release. Previously released versions are now considered obsolete.

InChI Software v. 1.05 includes several significant additions to previous versions. The main novel features are as follow:

- support for chemical element numbers 113-118 was newly added;
- experimental support of InChI/InChIKey for regular single-strand polymers was implemented;
- experimental support of large molecules containing up to 32767 atoms was added;
- ability to read input files in Molfile V3000 format was added; it is necessary for treating large molecules (previous versions supported only V2000 format limited to not more than 1000 atoms). Provisional support for extended features of Molfile V3000 was also added;
- InChI API Library was significantly updated, in particular:

  a novel API procedure for direct conversion of Molfile input to InChI has been added, as well as a demo program illustrating its use;

  a whole new set of API procedures for both low and high-level operations (InChI extensible interface, IXA) has been added, as well as a demo program illustrating its use;

- the source code was significantly changed in order to ensure multi-thread execution safety of the InChI Library; also, several minor bugfixes/changes (including refactoring) were made to the sources;

- several convenience options were added to the inchi-1 executable;

- documentation was updated.

New and updated features

Support of new chemical elements

The previous releases of InChI Software supported chemical elements up to dubnium to copernicium (element 112). In the current release, support was added for six elements adopted since v. 1.04 release. Of these, elements 114 (flerovium, Fl) and 116 (livermorium, Lv) were finally adopted in 2012. The elements 113 (nihonium, Nh), 115 (moscovium, Mc), 117 (tennessine, Ts), and 118 (oganesson, Og) were adopted in 2016 provisional recommendations; it is expected that they also will be finally approved in 2016.

The data for elements 105-112 (introduced in previous Software v. 1.04 release) were updated.

Source publications are:

J. Meija, T.B. Coplen, M.Berglund, W.A. Brand, P.De Bièvre, M. Gröning, N. E. Holden, J. Irrgeher, R.D. Loss, T. Walczyk and T. Prohaska. Atomic weights of the elements 2013 (IUPAC Technical Report). Pure Appl. Chem., Vol. 88, No. 3, pp. 265–291, 2016 and

L. Ohrstrom, J. Reedijk. Names and Symbols of the Elements with Atomic Numbers 113, 115, 117 and 118. Pure Appl. Chem., May 1, 2016, Manuscript ID PAC-REC-16-05-01. Available at http://iupac.org/cms/wp-content/uploads/2016/06/names-and-symbols-of-elements.pdf

InChI Software requires that an average atomic mass and the atomic mass of the most abundant isotope (as integer numbers) are tabulated. As all the added elements are artificial and have no stable isotopes, the atomic mass of the longest-lived isotope was used in all cases, as listed in Table 1 below.

The standard valences of newly added elements 114 and 116 were set to the formal value of 1.

Table 1. Data for added/updated chemical elements 105-118.

Element	Name	Element number	Atomic mass
Db	Dubnium	105	270
Sg	Seaborgium	106	269
Bh	Bohrium	107	270
Hs	Hassium	108	270
Mt	Meitnerium	109	278
Ds	Darmstadtium	110	281
Rg	Roentgenium	111	281
Cn	Copernicium	112	285
Nh	Nihonium	113	278
Fl	Flerovium	114	289
Mc	Moscovium	115	289
Lv	Livermorium	116	293
Ts	Tennessine	117	297
Og	Oganesson	118	294

Support of regular single-strand polymers

General

InChI for polymers was implemented following the principal recommendations from the IUPAC InChI Working Party on polymers.

Both structure-based and source-based representation and encoding of polymers are supported, see below.

Executable inchi-1 supports reading input Molfile files containing polymer description lines[1]. This support is also built into the API procedure `MakeINCHIFromMolfileText()` and the demo program mol2inchi included in this distribution. To generate InChI from molecular data stored in other formats, one may use InChI API Library polymer-aware procedures (`GetINCHIEx()` and new IXA calls) which accept specifically polymer-extended input data structures[2].

Limitations and known issues of InChI for polymers are summarized in a dedicated section below.

Note that support of polymers is an experimental feature. To emphasize this, InChI/InChIKey for a polymer uses the 'B' flag character (for "Beta"), instead of 'S' or 'N' for standard/non-standard InChI. It is supposed that this flag will be replaced by common standard/non-standard conventions if and when InChI for polymers is finally adopted. Also, by default the executable inchi-1 ignores polymer-specific data (which also ensures compatibility with the behaviour of previous versions); to allow treatment

---

[1] Properties block of V2000 format, lines: "M STY", "M SAL", "M SBL", "M SST", "M SCN", "M SLB", "M SDI", "M SMT"; see [CTFile Formats. Accelrys, December 2011. http://accelrys.com/products/collaborative-science/biovia-draw/ctfile-no-fee.html]

[2] See API Reference and the file "inchi_api.h". The source code of the IXA API/demo program test_ixa included in this distribution internally uses GetINCHIEx() and may serve as an example of using this extended procedure.

of polymers, one should explicitly use the new command line option `Polymers` (`-Polymers` under Linux or `/Polymers` under Windows).

Polymer ("modification") InChI layer, '/z'

An additional optional "modification" layer has been added to the InChI layout to encode polymeric structures.

This layer starts from two symbols '/z' and is located immediately before the stereo sub-layer (if any) of the main InChI layer.

That is, for InChI including all possible layers and sub-layers up to the polymer layer, the sequence is as follows:

`InChI=1B/.../c.../h.../q.../p...`**`/z`**`.../b.../t.../m.../s...(other layers)`

Note that, for metal-containing structures, in InChI created with RecMet option a polymer layer may appear twice (first in the metal-disconnected and second in the metal-reconnected part).

Quick examples:

InChI for styrene-butadiene block copolymer, source-based representation (entries 11 and 12 in Table 2 below):

`InChI=1B/C8H8.C4H6/c1-2-8-6-4-3-5-7-8;1-3-4-2/h2-7H,1H2;3-4H,1-2H2/z200-9-12;200-1-8;330-1-12`

`InChIKey=MTAZNLWOLGHBHU-ZNVYRHKRBA-N`

InChI for polycaprolactam, structure-based representation (entry 37 in Table 2):

`InChI=1B/C6H10O2/c7-6-4-2-1-3-5-8-6/h1-5H2/z101-1-8(1,2,1,3,2,4,3,5,4,6,5,8,6,8)`

`InChIKey=PAPBSGBWRJIAAV-CMRMDLKMBA-N`

InChI for coordination polymer, zinc-containing fungicide Zineb, structure-based representation produced with RecMet option (entry 64 in Table 2):

InChI=1B/C4H8N2S4.Zn/c7-3(8)5-1-2-6-4(9)10;/h1-2H2,(H2,5,7,8)(H2,6,9,10);/q;+2/p-2/z101-1-11(1,5)/rC4H6N2S4Zn/c1-2-6-4-9-11(10-4)7-3(5-1)8-11/h5-6H,1-2H2/z101-1-11(1,5)

InChIKey=AMHNZOICSMBGDH-ZPQHTIIXBA-L

Source-based and structure-based representations

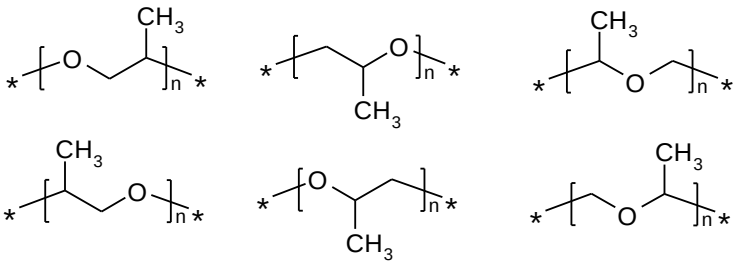
Source-based representation of polymers is based on the chemical structures of the starting material(s) with a special indication that the structure represents a polymer.

InChI encoding of source-based representation of polymers enhances general InChI encoding with a polymer layer used to specify polymer nature, type of polymer and the role and order of the components where needed. Provision is made for indicating the nature of copolymers - block, random, and alternating.

The structure-based representation of polymers is based on the structure of structural repeating units (SRU; sometimes called constitutional repeating units, CRU) enclosed in polymer brackets with possible indication of end-groups or “star” atoms.

InChI encoding of a structure-based representation includes, in the polymer layer,information on SRUs present.

For SRUs with “star” atoms as end groups, InChI encoding accounts for possible different ways to draw the same SRU. For example, the seemingly different structures below do represent (for polymer chemists) the same polymer:



In other words, there exist different options for slicing a repeating unit from the same “infinite” sequence formed by SRUs in the above picture: a so-called “phase shift”.

As a solution, one may require that some *canonical*[3] structure is selected, and all the others are converted to the canonical form before generating an identifier.

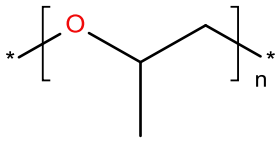
InChI for polymers uses a slightly different approach. It stores all the possible options for attachment of “star” atoms to the SRU backbone, leaving a choice of canonical form to the future (if any) step of restoring the structure from InChI, inchi2struct conversion.

To exemplify, all the structures above produce the same InChI and InChIKey:

InChI=1B/C3H6O/c1-3-2-4-3/h3H,2H2,1H3/z101-1-4(2,3,2,4,3,4)

InChIKey=GOOHAUXETOMSMM-KUWDYTNTBA-N

On inchi2struct conversion, this single InChI produces a single *canonical* structure:



Note that this canonical SRU is exactly the same as the preferred SRU recommended by IUPAC rules, see [*Kahovec, J.; Fox, R. B.; Hatada, K. Nomenclature of Regular Single-Strand Organic Polymers (IUPAC Recommendations 2002). Pure and Applied Chemistry 2002, 74, 1921–1956.*]. Though this behavior is not always the case, special care in InChI algorithm was taken to account for at least the basic IUPAC criteria (for more details see *InChI Software Technical Manual* included in the distribution).

[3] Here the term *canonical* is used in the polymer chemistry sense, which bears no relation to InChI internal canonicalization.

As for source-based representation, InChI encoding of copolymers accounts for the information about the type of polymer and roles of components, which is stored in the ‘/z\ layer.

Relation between source- and structure-based InChI encoding

Source- and structure-based representations and their InChI encodings are independent and, in general, no procedures are implied for algorithmic conversion and relation from one type to the other.

However, InChI for polymers is designed in such a way that, in many practically important cases, InChIs and InChIKeys for source- and structure-based representations have the same first part. For example, polypropylene (entries 72-74 in Table 2) is encoded in source-based form as:

InChI=1B/C3H6/c1-3-2/h3H,1H2,2H3/z200-1-3

InChIKey=QQONPFPTGQHPMA-DJNVOPQRBA-N

and in structure-based form as:

InChI=1B/C3H6/c1-3-2/h3H,1H2,2H3/z101-1-3(1.3)

InChIKey=QQONPFPTGQHPMA-KDIWSDSHBA-N

As is seen, InChI strings up to the polymer layer are identical, as well as the first block of InChIKey. Moreover, removing the ‘/z’ layer in both cases results in obtaining InChI for a monomer, propylene:

InChI=1S/C3H6/c1-3-2/h3H,1H2,2H3

InChIKey=QQONPFPTGQHPMA-UHFFFAOYSA-N

Another example is poly(caprolactam), nylon 6 (entries 75-77 in Table 2) which is encoded in source-based form as:

InChI=1B/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)/z200-1-8

InChIKey=JBKVHLHDHHXQEQ-ZMQGHSLKBA-N

and in structure-based form as:

InChI=1B/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)/z101-1-8(1,2,1,3,2,4,3,5,4,6,5,7)

InChIKey=JBKVHLHDHHXQEQ-DZWZRWJOBA-N

Again, the first parts of the identifier are identical, and they are also identical to InChI of monomeric caprolactam:

InChI=1S/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)

InChIKey=JBKVHLHDHHXQEQ-UHFFFAOYSA-N

However, such a convenient feature of polymer encoding is not always available. It is enough to notice that, generally, the same polymer with the same structure-based representation may be obtained from different starting materials, that is, have different source-based representations. A general relation between source- and structure-based InChIs and InChIKeys can be established via procedures close to the InChI resolver protocol developed for discrete structures. For polymers this task is simple due to the significantly smaller number of polymers as compared with the number of discrete structures.

Limitations and known issues

Note that InChI for polymers is in experimental stage, so there are a number of limitations, as described below.

In particular:

- Polymers other than single-strand (cross-linked, etc.) are not supported (though specification of “CRO”, the cross-linking nature of a copolymer component, is allowed in source-based representation).
- Polymers with mixed-style (both source-based and structure-based) representation are not supported.
- Only either zero (source-based representation) or two (structure-based representation) crossing bonds per SRU is allowed; ladder polymers are not supported.

- Mixing “star” atom with normal-atom end-groups for the same SRU is not supported.
- Polymers with an explicitly drawn hydrogen end group are not supported.
- Polymers with exactly two triple-bonded atoms in the SRU chain between the “star” atoms, *-X≡Y-* are not supported.
- Possibility of simplification of SRU (e.g., polyethylene to polymethylene, etc.) is not considered.
- No provision for possible “phase shift” is made in the case of structure-based SRUs with “non-star” atom end-groups. Also, “phase shift” may be not recognized in metal-disconnected InChI for metal-containing SRUs.

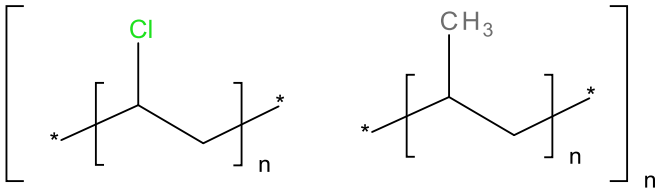
Preparing input data and drawing rules

It is assumed that input data for inchi-1 (winchi-1) are files in Molfile V2000 format as described in [CTFile Formats. Accelrys, December 2011. http://accelrys.com/products/collaborative-science/biovia-draw/ctfile-no-fee.html]. Such files may be easily prepared and saved by using many available chemical drawing programs (for example, Accelrys Draw, ACD ChemSketch, etc.). The polymer data structures which are necessary to call API procedures are presented in API Reference and file “inchi_api.h”: they closely follow Molfile data layout.

Also, there are some rules which should be followed, to ensure correct functioning of the software. When drawing (or preparing related data structures):

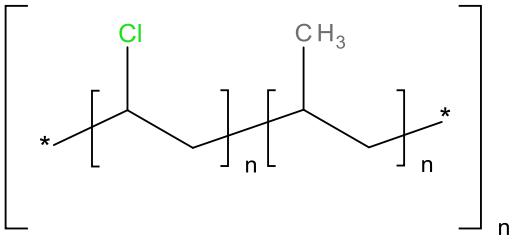
- Always place brackets around polymer SRUs or monomers, as well as the whole copolymer.
- For copolymers, use the representations with disconnected SRUs (I not II below).

Correct (disconnected presentation of copolymer, all SRUs are in brackets, star atom connections in component SRU)



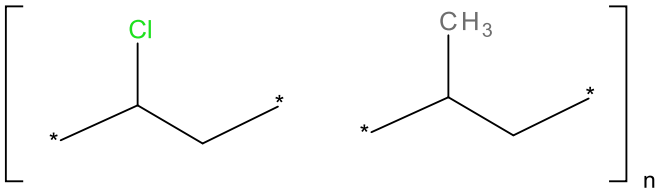
I

Incorrect (connected presentation of copolymer, mix of star atom and specific end group in connections of same SRU)



II

Incorrect (no brackets at components)



III

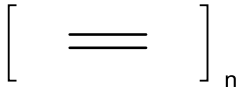
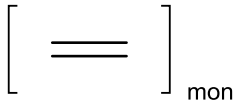
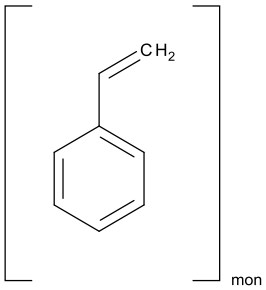
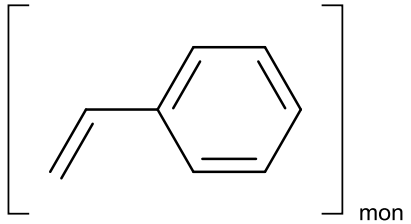
Please also consult Table 2 below (and accompanying file INCHI-1-TEST /PEX.SDF) to check for valid representation in various complex cases.

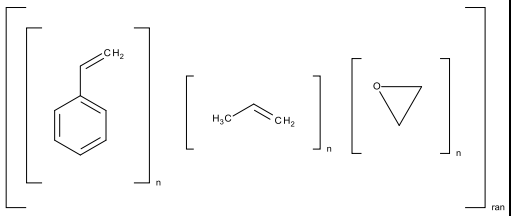
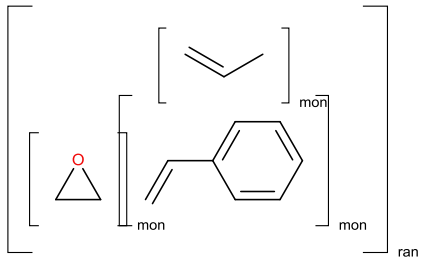
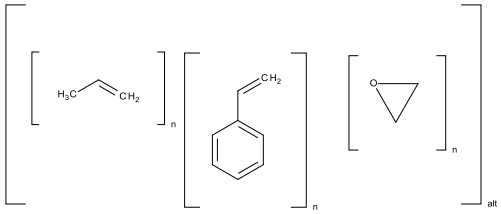
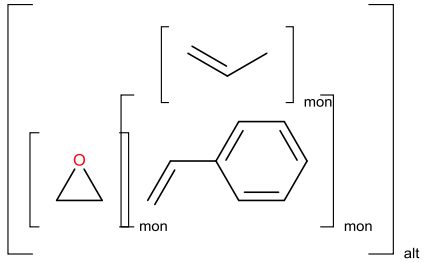
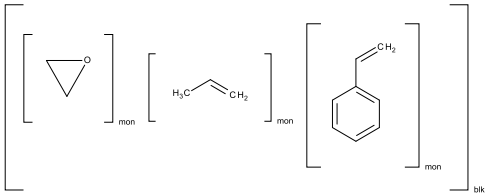
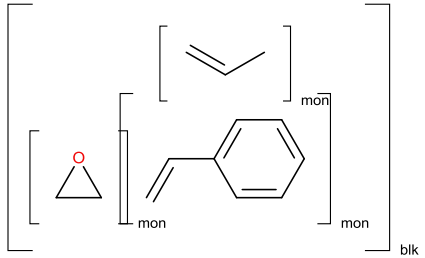
Note that:

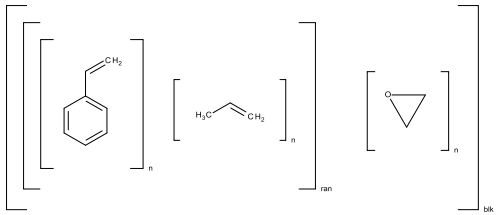
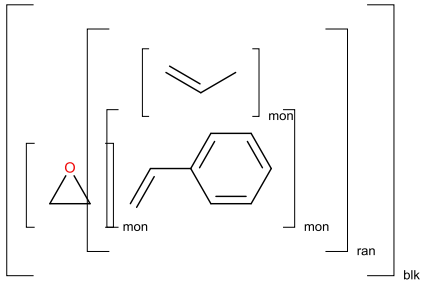
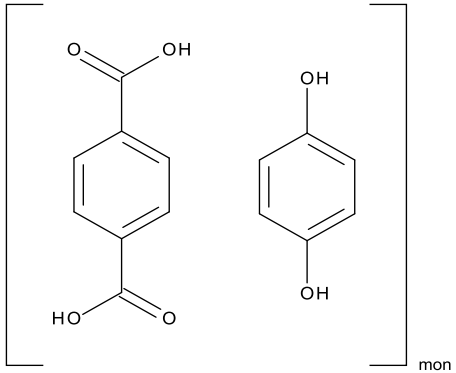
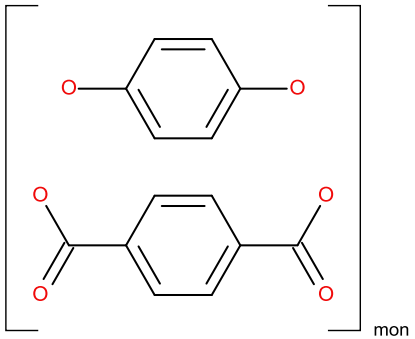
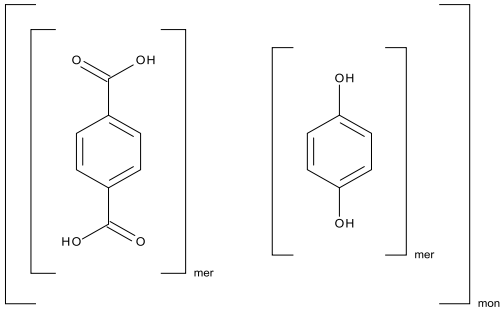
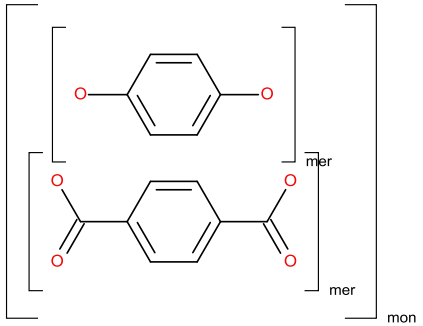
- If SRU connectivity (“HT”, “HH”, “EU”) is not specified, the default “EU” (‘either/unknown’) is inserted.
- For copolymers, if the type is not specified, random (“RAN”) is assumed.

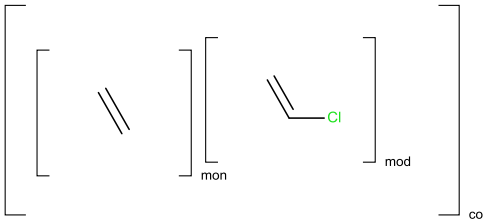
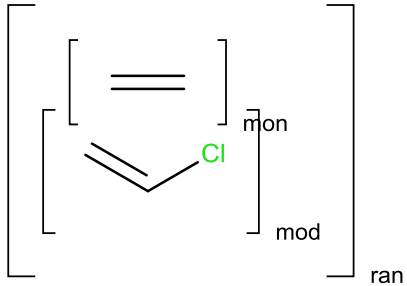
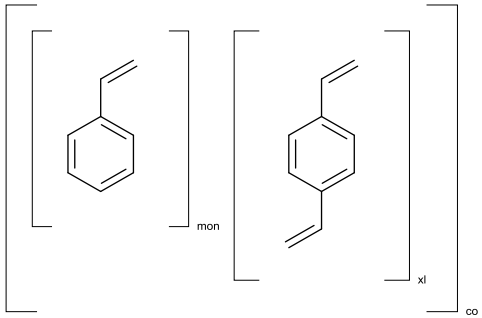
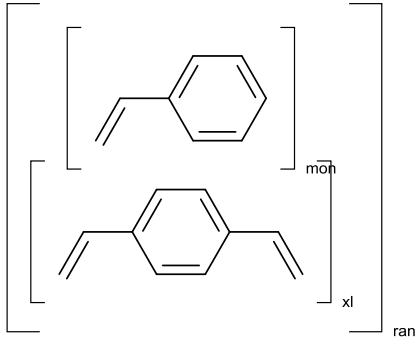
Table 2. Examples of InChI for polymers.

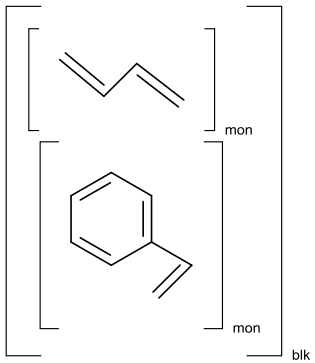
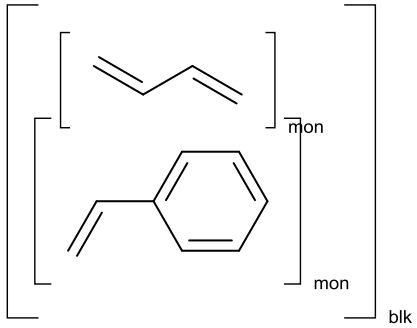
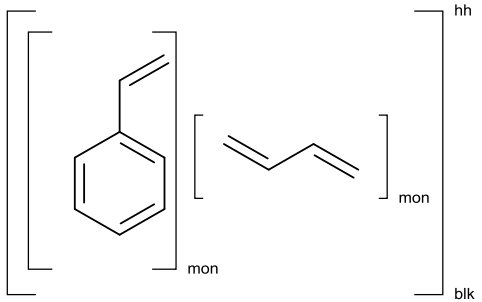
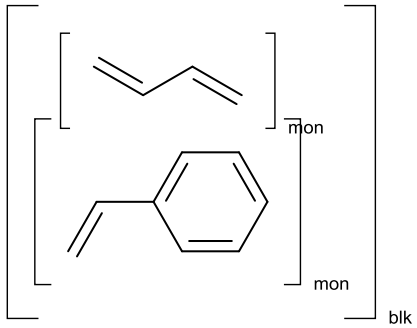
Numbering follows numbers of records in the file PEX.SDF included in this distribution. Structures back-converted from InChI as follows: restore from InChI with inchi-1 with options ‘InChI2Struct’ and ‘OutputSDF’, then import produced *xyz*- and stereo-less Molfile to Accelrys Draw 4.1 & make ‘Chemistry->Clean’.

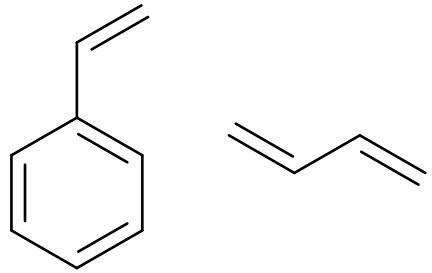
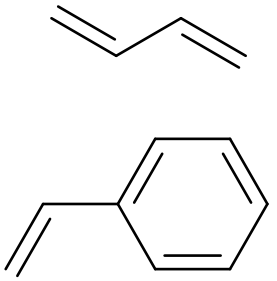
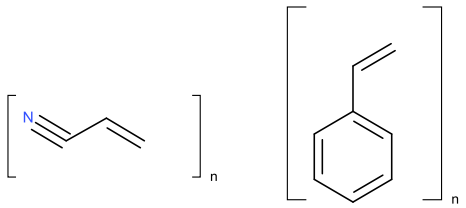
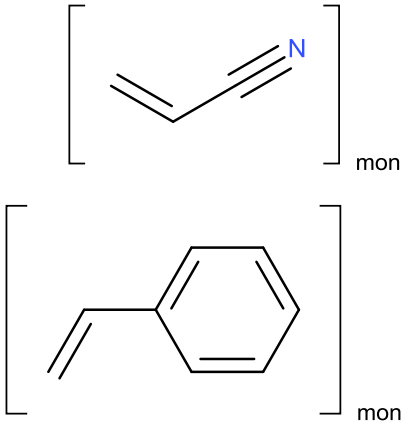
Name	Structure	InChI/Key	Comment and/or structure as back-converted from InChI*
pex.sdf/1		InChI=1B/C2H4/c1-2/h1-2H2/z200-1-2 InChIKey=VGGSQFUCUMXWEO-JEIJVWOYBA-N	
2		InChI=1B/C8H8/c1-2-8-6-4-3-5-7-8/h2-7H,1H2/z200-1-8 InChIKey=PPBRXRYQALVLMV-ZMQGHSLKBA-N	

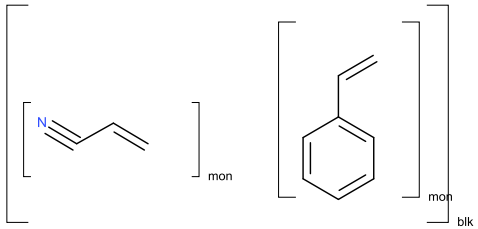
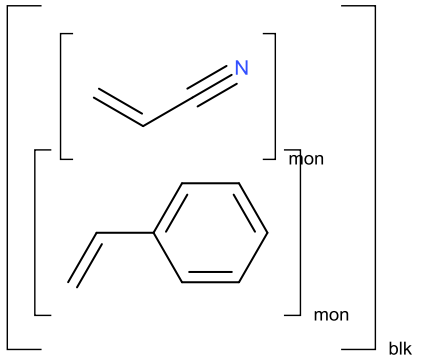
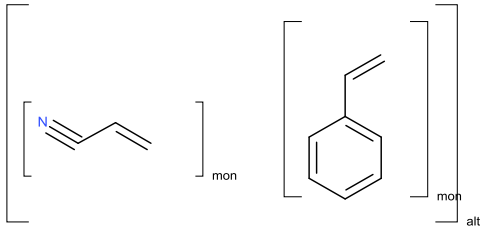
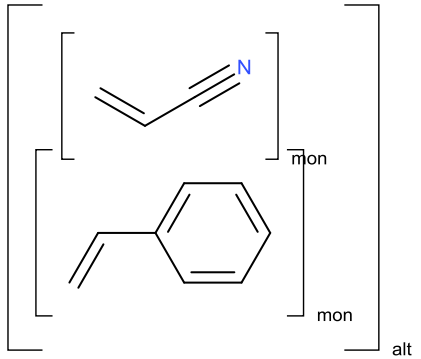
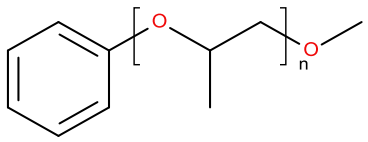
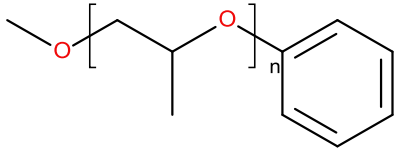
3		InChI=1B/C8H8.C3H6.C2H4O/c1-2-8-6-4-3-5-7-8;1-3-2;1-2-3-1/h2-7H,1H2;3H,1H2,2H3;1-2H2/z200-9-11;200-12-14;200-1-8;320-1-14 InChIKey=MYPPWGFYCAYBAJ-CALUOSFZBA-N	
4		InChI=1B/C8H8.C3H6.C2H4O/c1-2-8-6-4-3-5-7-8;1-3-2;1-2-3-1/h2-7H,1H2;3H,1H2,2H3;1-2H2/z200-9-11;200-12-14;200-1-8;310-1-14 InChIKey=MYPPWGFYCAYBAJ-WRCKSDIVBA-N	
5		InChI=1B/C8H8.C3H6.C2H4O/c1-2-8-6-4-3-5-7-8;1-3-2;1-2-3-1/h2-7H,1H2;3H,1H2,2H3;1-2H2/z200-9-11;200-12-14;200-1-8;330-1-14 InChIKey=MYPPWGFYCAYBAJ-CQPMJGTBBA-N	

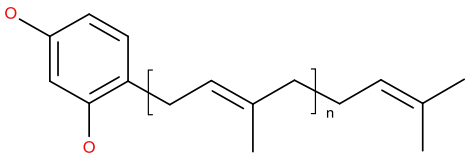
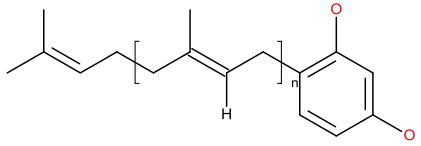
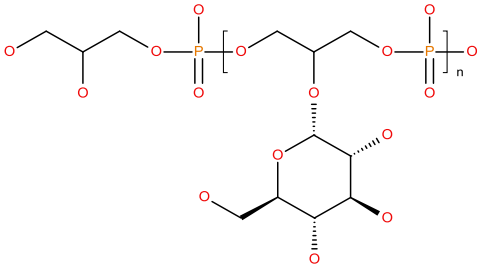
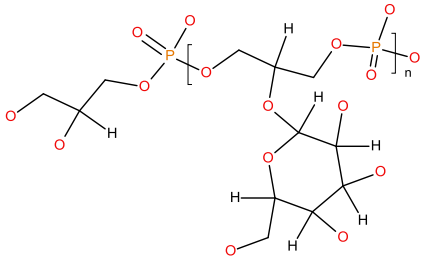
6		InChI=1B/C8H8.C3H6.C2H4O/c1-2-8-6-4-3-5-7-8;1-3-2;1-2-3-1/h2-7H,1H2;3H,1H2,2H3;1-2H2/z200-9-11;200-12-14;200-1-8;320-1-11;330-1-14 InChIKey=MYPPWGFYCAYBAJ-MCGIIRCZBA-N	
7		InChI=1B/C8H6O4.C6H6O2/c9-7(10)5-1-2-6(4-3-5)8(11)12;7-5-1-2-6(8)4-3-5/h1-4H,(H,9,10)(H,11,12);1-4,7-8H/z200-1-20 InChIKey=RYGHNMANMHVMMN-NRYBNUJIBA-N	
8		InChI=1B/C8H6O4.C6H6O2/c9-7(10)5-1-2-6(4-3-5)8(11)12;7-5-1-2-6(8)4-3-5/h1-4H,(H,9,10)(H,11,12);1-4,7-8H/z600-13-20;600-1-12;200-1-20 InChIKey=RYGHNMANMHVMMN-AMBOAQDBBA-N	

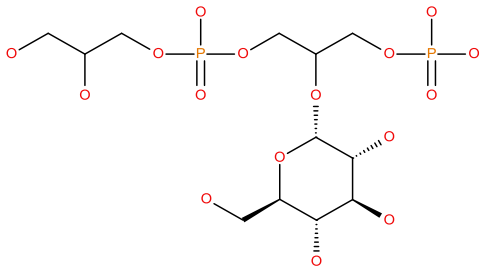
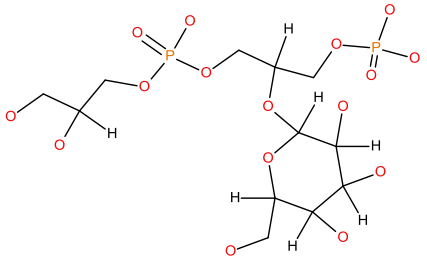
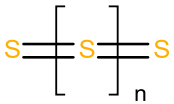
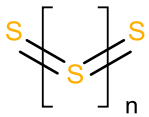
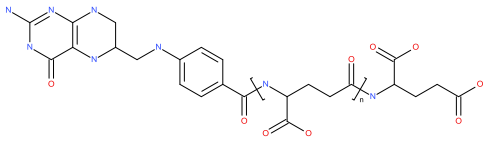
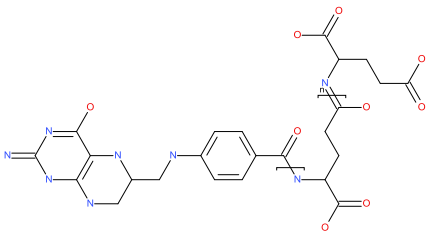
9		InChI=1B/C2H3Cl.C2H4/c1-2-3;1-2/h2H,1H2;1-2H2/z200-4-5;400-1-3;320-1-5 InChIKey=KRGNPJFAKZHQPS-PIORLKTDBA-N	
10		InChI=1B/C10H10.C8H8/c1-3-9-5-7-10(4-2)8-6-9;1-2-8-6-4-3-5-7-8/h3-8H,1-2H2;2-7H,1H2/z200-11-18;500-1-10;320-1-18 InChIKey=KOLSKAPZYPZSGC-AVCIZLQUBA-N	

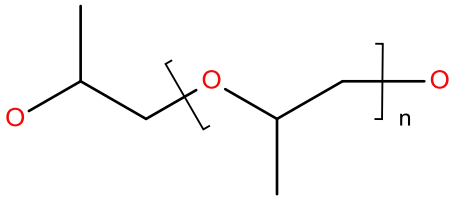
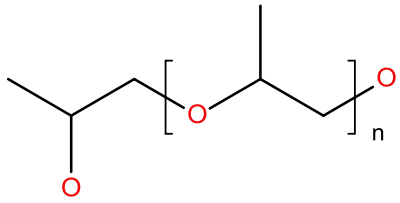
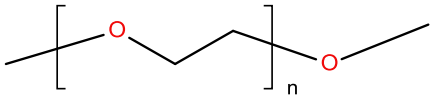
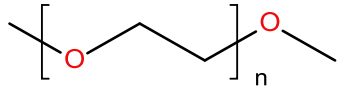
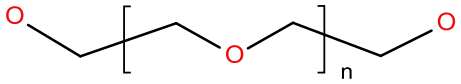
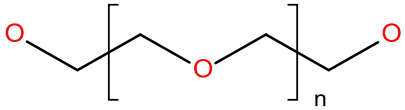
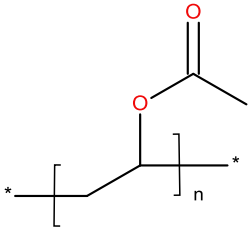
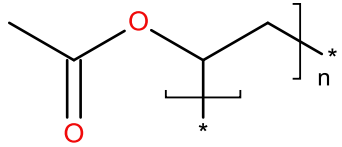
11		InChI=1B/C8H8.C4H6/c1-2-8-6-4-3-5-7-8;1-3-4-2/h2-7H,1H2;3-4H,1-2H2/z200-9-12;200-1-8;330-1-12 InChIKey=MTAZNLWOLGHBHU-ZNVYRHKRBA-N	
12		InChI=1B/C8H8.C4H6/c1-2-8-6-4-3-5-7-8;1-3-4-2/h2-7H,1H2;3-4H,1-2H2/z200-9-12;200-1-8;330-1-12 InChIKey=MTAZNLWOLGHBHU-ZNVYRHKRBA-N	 ‘hh’ is ignored

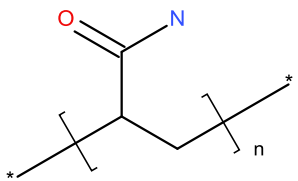
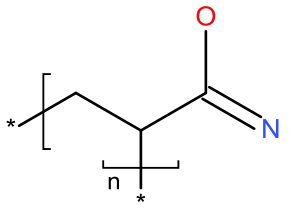
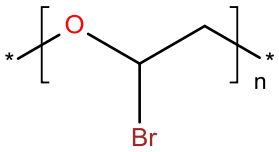
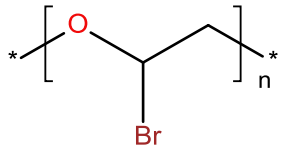
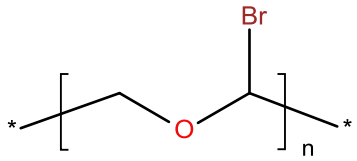
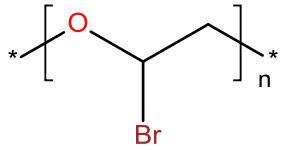
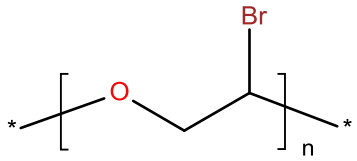
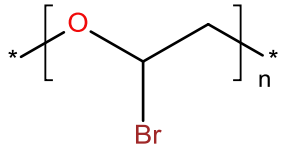
13		InChI=1S/C8H8.C4H6/c1-2-8-6-4-3-5-7-8;1-3-4-2/h2-7H,1H2;3-4H,1-2H2 InChIKey=MTAZNLWOLGHBHU-UHFFFAOYSA-N	 Not a polymer. Note the same 1st block as the previous entry
14		InChI=1B/C8H8.C3H3N/c1-2-8-6-4-3-5-7-8;1-2-3-4/h2-7H,1H2;2H,1H2/z200-9-12;200-1-8 InChIKey=SCUZVMOVTVSBLE-BEPCRINZBA-N	 Considered as mixture of polymers

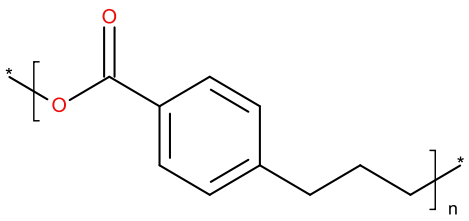
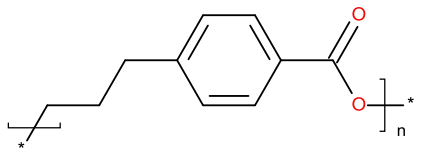
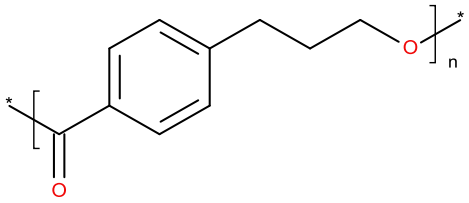
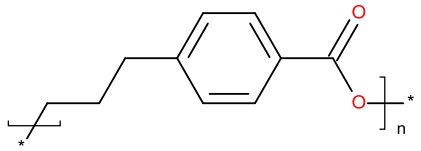
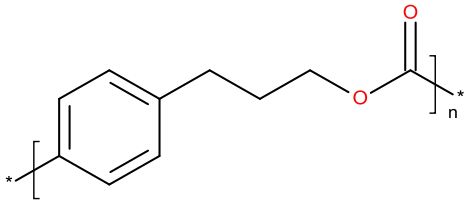
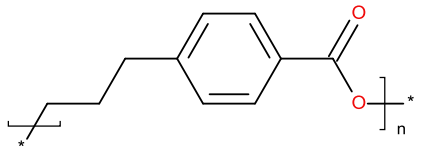
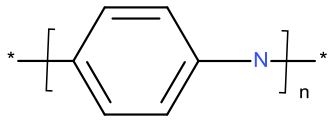
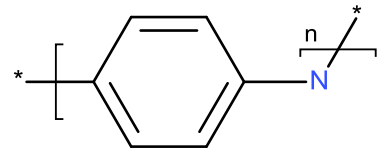
15		InChI=1B/C8H8.C3H3N/c1-2-8-6-4-3-5-7-8;1-2-3-4/h2-7H,1H2;2H,1H2/z200-9-12;200-1-8;330-1-12 InChIKey=SCUZVMOVTVSBLE-ZNVYRHKRBA-N	 Block copolymer
16		InChI=1B/C8H8.C3H3N/c1-2-8-6-4-3-5-7-8;1-2-3-4/h2-7H,1H2;2H,1H2/z200-9-12;200-1-8;310-1-12 InChIKey=SCUZVMOVTVSBLE-YMBOZTOTBA-N	 Alternating copolymer
17		InChI=1B/C10H14O2/c1-9(8-11-2)12-10-6-4-3-5-7-10/h3-7,9H,8H2,1-2H3/z101-1,8-9,12(10-12,11-8)	

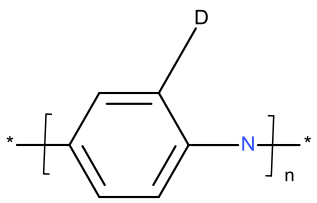
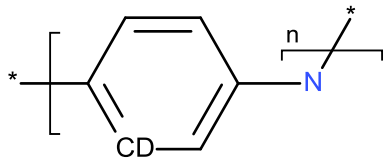
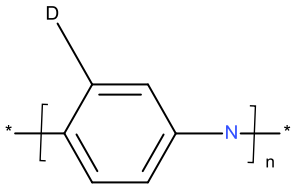
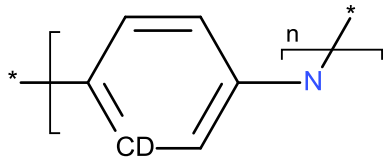
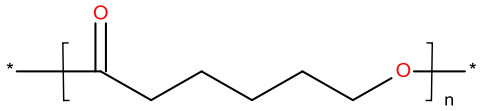
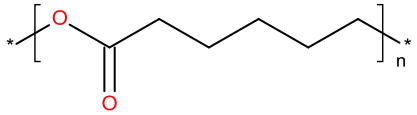
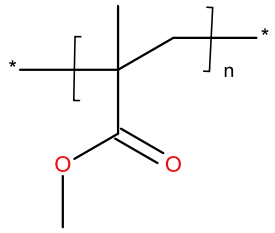
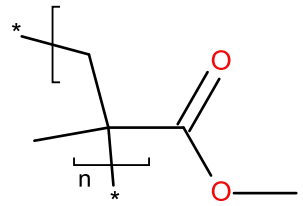
		InChIKey=WGAOQOCTPHKPGH-VUBNIWPLBA-N	
18		InChI=1B/C16H22O2/c1-12(2)5-4-6-13(3)7-8-14-9-10-15(17)11-16(14)18/h5,7,9-11,17-18H,4,6,8H2,1-3H3/z101-3,6-8,13(4-6,14-8)/b13-7+ InChIKey=RGUSQESFCCLNEG-HDMSAIMJBA-N	
19		InChI=1B/C12H26O16P2/c13-1-6(15)3-25-30(22,23)26-5-7(4-24-29(19,20)21)27-12-11(18)10(17)9(16)8(2-14)28-12/h6-18H,1-5H2,(H,22,23)(H2,19,20,21)/z101-2,4-5,7-12,14,16-19,21,24,26-29(20-29,30-26)/t6?,7?,8-,9-,10+,11-,12+/m1/s1 InChIKey=CVGGAOGCVJSXBB-OHILBRHSBA-N	

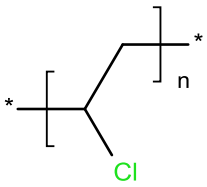
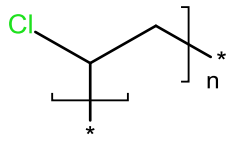
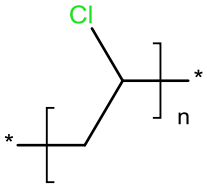
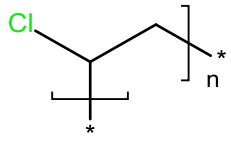
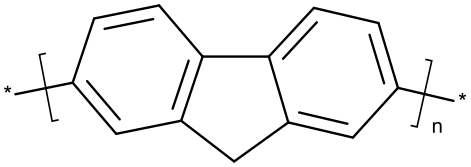
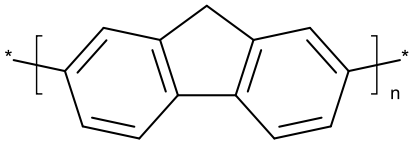
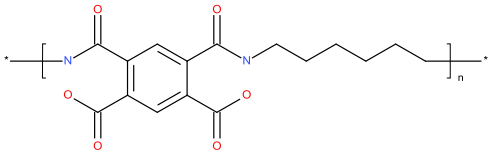
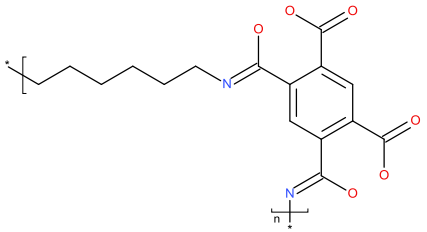
20		InChI=1S/C12H26O16P2/c13-1-6(15)3-25-30(22,23)26-5-7(4-24-29(19,20)21)27-12-11(18)10(17)9(16)8(2-14)28-12/h6-18H,1-5H2,(H,22,23)(H2,19,20,21)/t6?,7?,8-,9-,10+,11-,12+/m1/s1 InChIKey=CVGGAOGCVJSXBB-XMDCDNQBSA-N	 Not a polymer. Note the same 1st block tas the previous entry
21		InChI=1B/S3/c1-3-2/z101-3(1-3,2-3) InChIKey=NVSDADJBGGUCLP-IVYKBIFBBA-N	
22		InChI=1B/C24H30N8O9/c25-24-31-19-18(21(37)32-24)28-13(10-27-19)9-26-12-3-1-11(2-4-12)20(36)30-15(23(40)41)5-7-16(33)29-14(22(38)39)6-8-17(34)35/h1-4,13-15,26,28H,5-10H2,(H,29,33)(H,30,36)(H,34,35)(H,38,39)(H,40,41)(H4,25,27,31,32,37)/z103-5,7,15-16,23,30,33,40-41(20-30,29-16) InChIKey=ZAOGJXDWOQXFBW-LLEXUJLJBA-N	

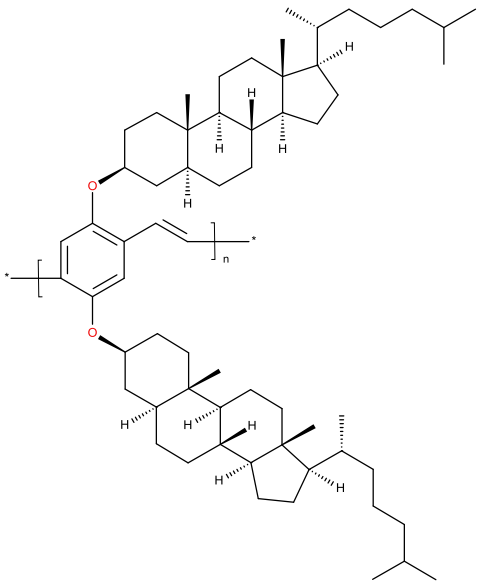
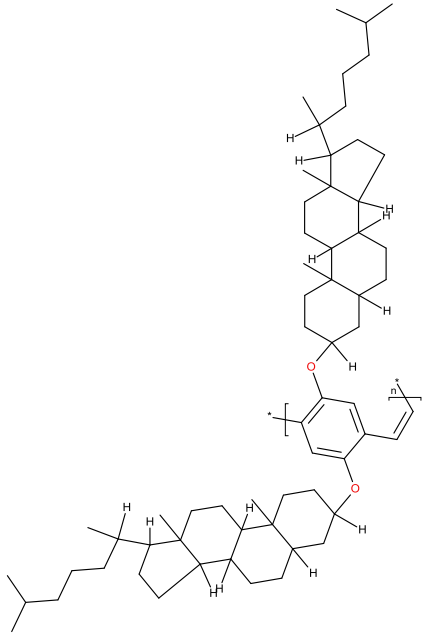
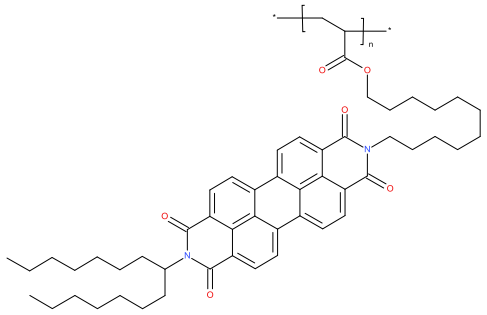
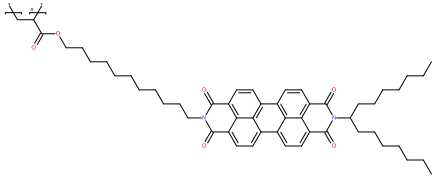
23		InChI=1B/C6H14O3/c1-5(8)4-9-6(2)3-7/h5-8H,3-4H2,1-2H3/z101-2-3,6,9(4-9,7-3) InChIKey=DUFKCOQISQKSAV-QZIZWDOBBA-N	
24		InChI=1B/C4H10O2/c1-5-3-4-6-2/h3-4H2,1-2H3/z101-3-5(1-5,6-4) InChIKey=XTHFKEDIFFGKHM-MYPDUYLMBA-N	
25		InChI=1B/C4H10O3/c5-1-3-7-4-2-6/h5-6H,1-4H2/z101-3-4,7(1-3,2-4) InChIKey=MTHSVFCYNBDYFN-ZQIFOVHGBA-N	
26		InChI=1B/C4H6O2/c1-3-6-4(2)5/h3H,1H2,2H3/z101-1-6(1.3) InChIKey=XTXRWKRVRITETP-IKXXVMLBBA-N	

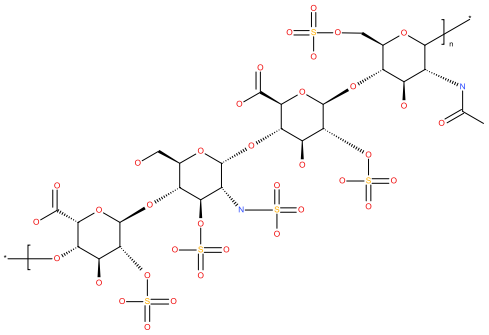
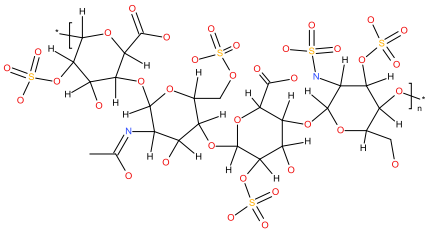
27		InChI=1B/C3H5NO/c1-2-3(4)5/h2H,1H2,(H2,4,5)/z101-1-5(1.2) InChIKey=HRPVXLWXLXDGHG-YZJLBCBGBA-N	
28		InChI=1B/C2H3BrO/c3-2-1-4-2/h2H,1H2/z101-1-4(1,2,1,4,2,4) InChIKey=XOOVDXMNWOFROX-UAKBPNCWBA-N	
29		InChI=1B/C2H3BrO/c3-2-1-4-2/h2H,1H2/z101-1-4(1,2,1,4,2,4) InChIKey=XOOVDXMNWOFROX-UAKBPNCWBA-N	
30		InChI=1B/C2H3BrO/c3-2-1-4-2/h2H,1H2/z101-1-4(1,2,1,4,2,4) InChIKey=XOOVDXMNWOFROX-UAKBPNCWBA-N	

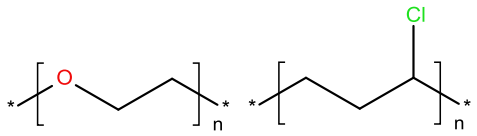
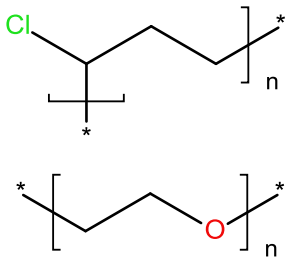
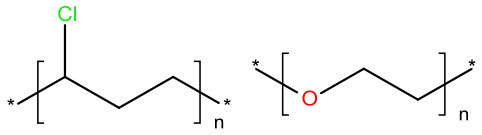
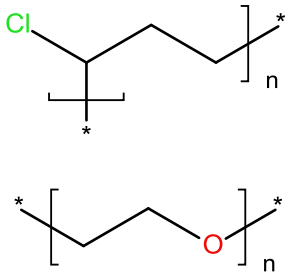
31		InChI=1B/C10H10O2/c11-10-9-5-3-8(4-6-9)2-1-7-12-10/h3-6H,1-2,7H2/z101-1-12(1,2,1,7,2,8,7,12,9,10,10,12) InChIKey=MZCUXAXPLLHRKB-WSHGCIBBBA-N	
32		InChI=1B/C10H10O2/c11-10-9-5-3-8(4-6-9)2-1-7-12-10/h3-6H,1-2,7H2/z101-1-12(1,2,1,7,2,8,7,12,9,10,10,12) InChIKey=MZCUXAXPLLHRKB-WSHGCIBBBA-N	
33		InChI=1B/C10H10O2/c11-10-9-5-3-8(4-6-9)2-1-7-12-10/h3-6H,1-2,7H2/z101-1-12(1,2,1,7,2,8,7,12,9,10,10,12) InChIKey=MZCUXAXPLLHRKB-WSHGCIBBBA-N	
34		InChI=1B/C6H5N/c1-2-6-4-3-5(1)7-6/h1-4,7H/z101-1-7(5,7,6,7) InChIKey=MRPDSIUDKSQKSH-DNFXZTSDBA-N	

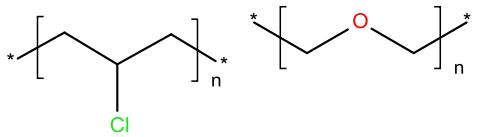
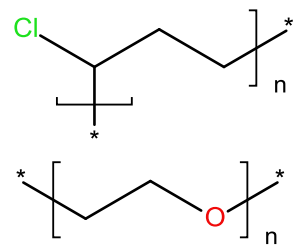
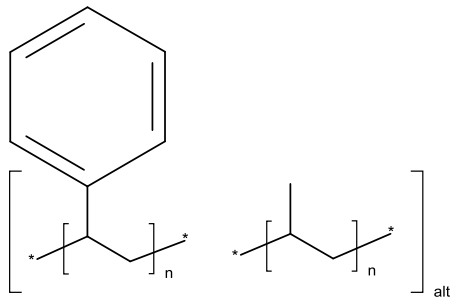
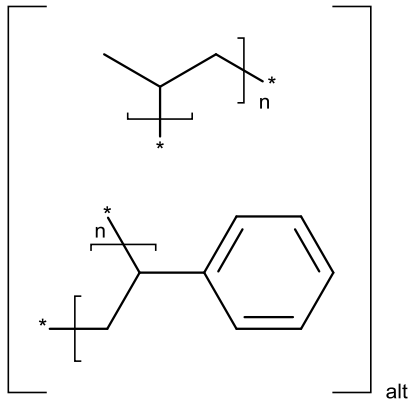
35		InChI=1B/C6H5N/c1-2-6-4-3-5(1)7-6/h1-4,7H/z101-1-7(5,7,6,7)/i1D InChIKey=MRPDSIUDKSQKSH-JTPWJMIPBA-N	
36		InChI=1B/C6H5N/c1-2-6-4-3-5(1)7-6/h1-4,7H/z101-1-7(5,7,6,7)/i1D InChIKey=MRPDSIUDKSQKSH-JTPWJMIPBA-N	
37		InChI=1B/C6H10O2/c7-6-4-2-1-3-5-8-6/h1-5H2/z101-1-8(1,2,1,3,2,4,3,5,4,6,5,8,6,8) InChIKey=PAPBSGBWRJIAAV-CMRMDLKMBA-N	
38		InChI=1B/C5H8O2/c1-4(2)5(6)7-3/h1H2,2-3H3/z101-1-7(1.4) InChIKey=VVQNEPGJFQJSBK-KCAOANCDBA-N	

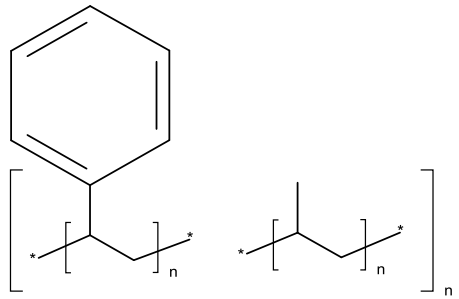
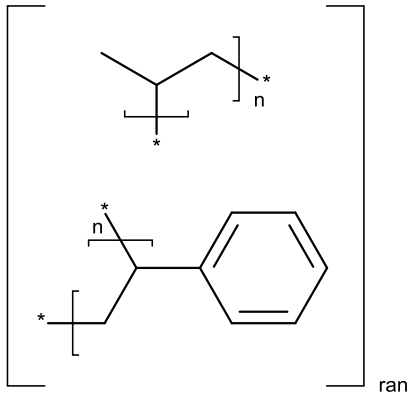
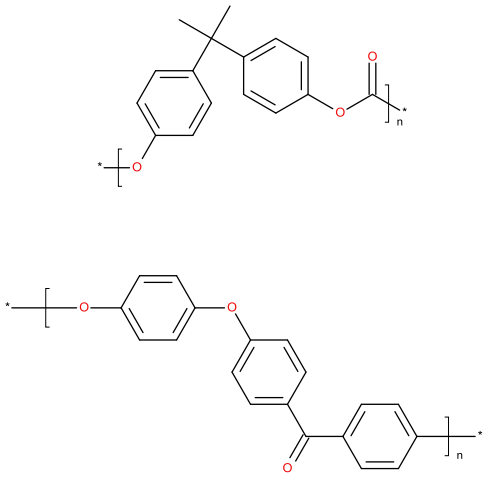
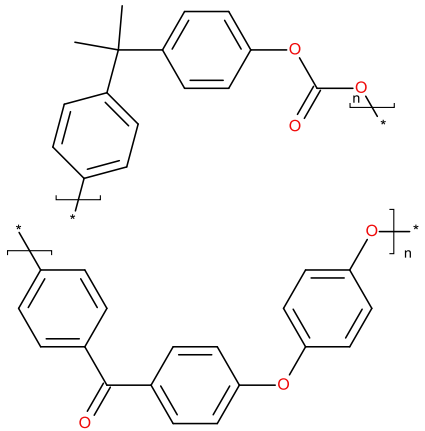
39		InChI=1B/C2H3Cl/c1-2-3/h2H,1H2/z101-1-3(1.2) InChIKey=BZHJMEDXRYGGRV-QLSTWJNGBA-N	
40		InChI=1B/C2H3Cl/c1-2-3/h2H,1H2/z101-1-3(1.2) InChIKey=BZHJMEDXRYGGRV-QLSTWJNGBA-N	
41		InChI=1B/C13H8/c1-3-12-10-5-8(1)9-2-4-13(12)11(6-9)7-10/h1-6H,7H2/z101-1-13(8,9) InChIKey=JFICTVGWKKNZIQ-XVERVCKYBA-N	
42		InChI=1B/C16H18N2O6/c19-13-9-7-10(12(16(23)24)8-11(9)15(21)22)14(20)18-6-4-2-1-3-5-17-13/h7-8H,1-6H2,(H,17,19)(H,18,20)(H,21,22)(H,23,24)/z101-1-24(1,2,1,3,2,4,3,5,4,6,5,17,6,18,9,13,10,14)	

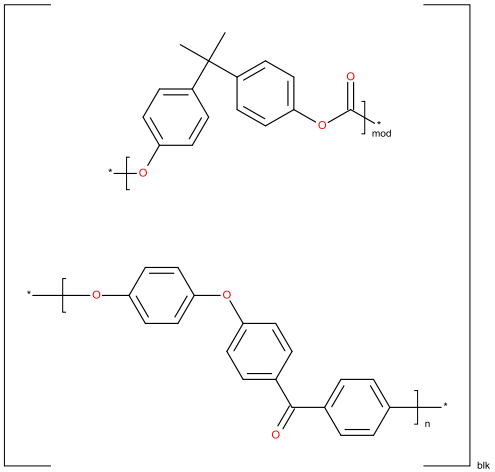
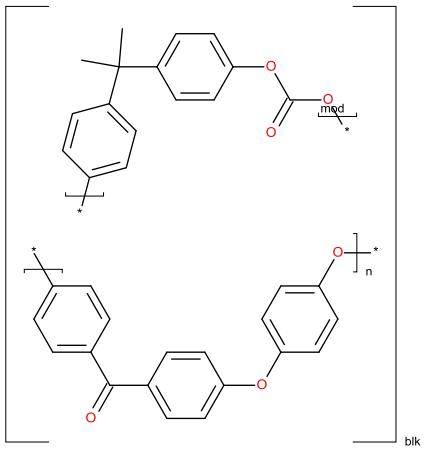
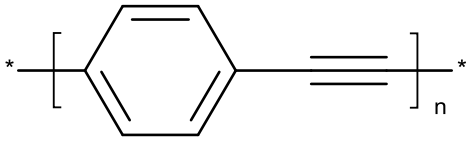
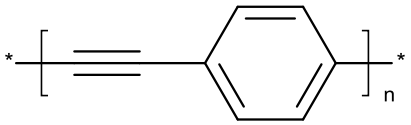
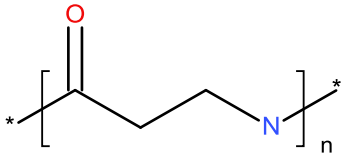
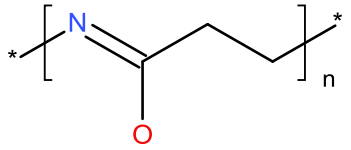
		InChIKey=NRXUBKMTMFYTLT-HYQSQRQEBA-N	
43		InChI=1B/C62H98O2/c1-39(2)13-11-15-41(5)51-23-25-53-49-21-19-45-37-47(27-31-59(45,7)55(49)29-33-61(51,53)9)63-57-35-44-18-17-43(57)36-58(44)64-48-28-32-60(8)46(38-48)20-22-50-54-26-24-52(42(6)16-12-14-40(3)4)62(54,10)34-30-56(50)60/h17-18,35-36,39-42,45-56H,11-16,19-34,37-38H2,1-10H3/z101-1-64(17,18,17,43,18,44)/t41-,42-,45+,46+,47+,48+,49+,50+,51-,52-,53+,54+,55+,56+,59+,60+,61-,62-/m1/s1 InChIKey=PYGFEFGUKGGHSU-DBPKQELEBA-N	
44		InChI=1B/C53H64N2O6/c1-4-7-9-16-20-24-36(25-21-17-10-8-5-2)55-52(59)43-32-28-39-37-26-30-41-48-42(31-27-38(46(37)48)40-29-33-44(53(55)60)49(43)47(39)40)51(58)54(50(41)57)34-22-18-14-12-11-13-15-19-23-	

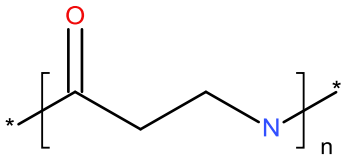
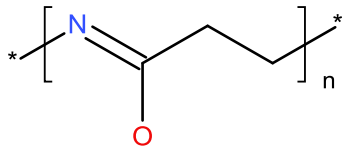
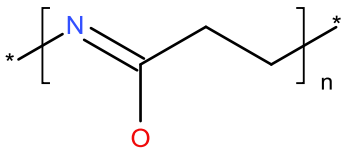
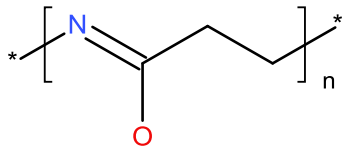
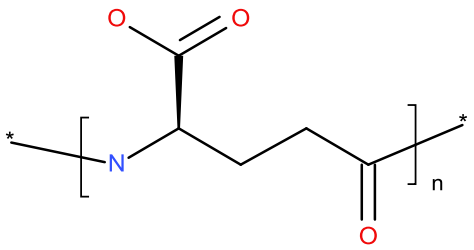
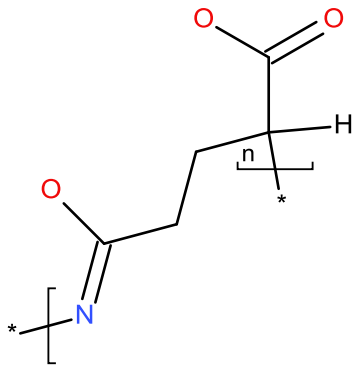
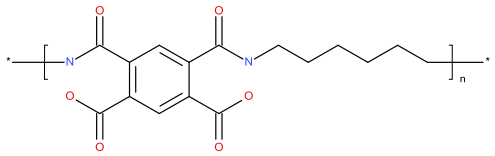
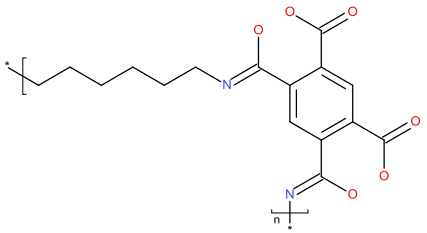
		35-61-45(56)6-3/h6,26-33,36H,3-5,7-25,34-35H2,1-2H3/z101-1-61(3.6) InChIKey=CXJULOVRHUZQEG-PHGMDXHZBA-N	
45		InChI=1B/C26H40N2O36S5/c1-4(30)27-7-9(31)12-6(3-53-66(41,42)43)55-23(7)58-15-10(32)18(64-69(50,51)52)26(61-19(15)21(34)35)57-13-5(2-29)54-24(8(28-65(38,39)40)14(13)62-67(44,45)46)59-16-11(33)17(63-68(47,48)49)25(56-12)60-20(16)22(36)37/h5-20,23-26,28-29,31-33H,2-3H2,1H3,(H,27,30)(H,34,35)(H,36,37)(H,38,39,40)(H,41,42,43)(H,44,45,46)(H,47,48,49)(H,50,51,52)/z101-1-69(12,56,13,57,15,58,16,59,23,58,24,59,25,56,26,57)/t5-,6-,7-,8-,9-,10+,11+,12-,13-,14-,15+,16+,17-,18-,19-,20+,23?,24-,25-,26-/m1/s1 InChIKey=RTKRRPJNBRRQDW-XTCKGPBSBA-N	

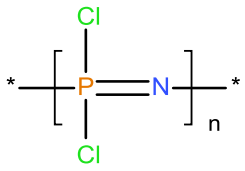
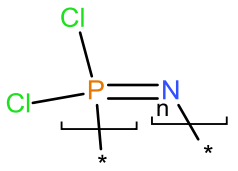
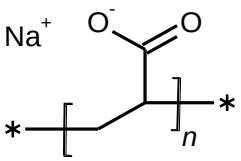
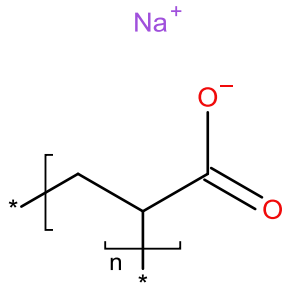
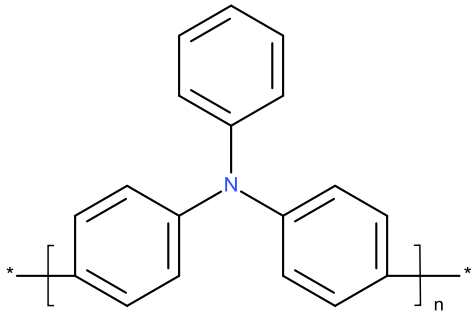
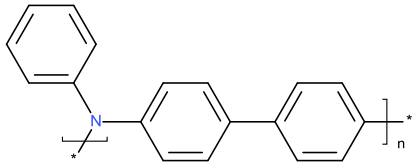
46		InChI=1B/C3H5Cl.C2H4O/c4-3-1-2-3;1-2-3-1/h3H,1-2H2;1-2H2/z101-5-7(5,6,5,7,6,7);101-1-4(1,2,1,3,2,3) InChIKey=KVIOBMVNVRVGPK-JKXHHPFUBA-N	 NB: mixture of polymers, not copolymer
47		InChI=1B/C3H5Cl.C2H4O/c4-3-1-2-3;1-2-3-1/h3H,1-2H2;1-2H2/z101-5-7(5,6,5,7,6,7);101-1-4(1,2,1,3,2,3) InChIKey=KVIOBMVNVRVGPK-JKXHHPFUBA-N	 NB: mixture of polymers, not copolymer

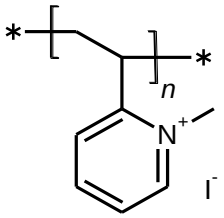
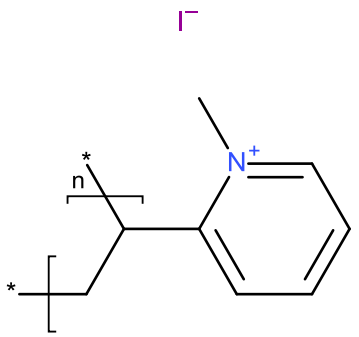
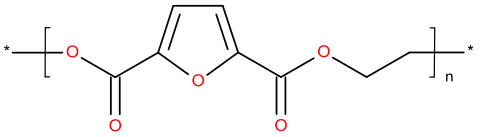
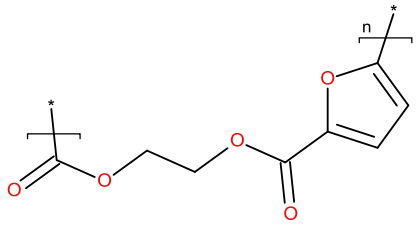
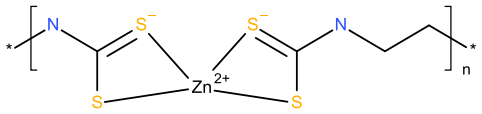
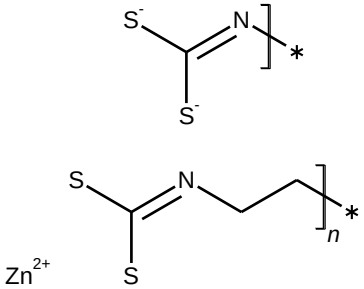
48		InChI=1B/C3H5Cl.C2H4O/c4-3-1-2-3;1-2-3-1/h3H,1-2H2;1-2H2/z101-5-7(5,6,5,7,6,7);101-1-4(1,2,1,3,2,3) InChIKey=KVIOBMVNVRVGPK-JKXHHPFUBA-N	 NB: mixture of polymers, not copolymer
49		InChI=1B/C8H8.C3H6/c1-2-8-6-4-3-5-7-8;1-3-2/h2-7H,1H2;3H,1H2,2H3/z101-9-11(9.11);101-1-8(1.2);311-1-15 InChIKey=YARNEMCKJLFQHG-NTLSNBPHBA-N	

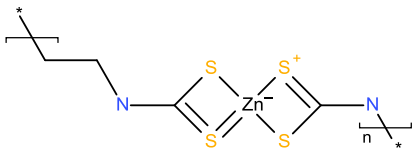
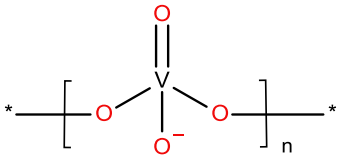
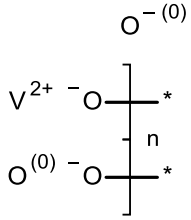
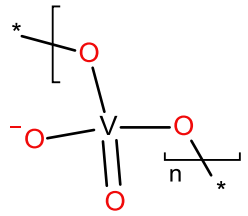
50		InChI=1B/C8H8.C3H6/c1-2-8-6-4-3-5-7-8;1-3-2/h2-7H,1H2;3H,1H2,2H3/z101-9-11(9.11);101-1-8(1.2);321-1-15 InChIKey=YARNEMCKJLFQHG-AESXVLFWBA-N	
51		InChI=1B/C19H12O3.C16H14O3/c20-19-13-1-5-15(6-2-13)21-17-9-11-18(12-10-17)22-16-7-3-14(19)4-8-16;1-16(2)11-3-7-13(8-4-11)18-15(17)19-12-5-9-14(10-6-12)20-16/h1-12H;3-10H,1-2H3/z101-23-41(33,38,34,38,35,40,36,41,37,40,37,41);101-1-22(13,19,14,19,15,21,16,22,17,21,18,22) InChIKey=HHFXSTKAFNJYEC-JELLZUPSBA-N	 NB: mixture of polymers, not copolymer

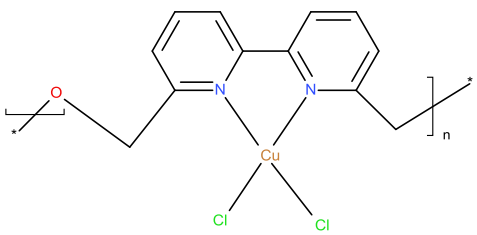
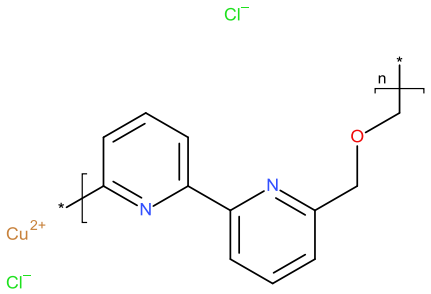
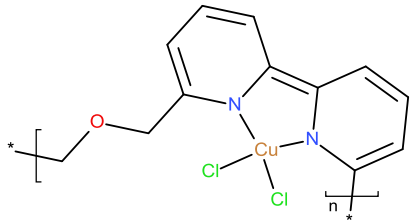
52		InChI=1B/C19H12O3.C16H14O3/c20-19-13-1-5-15(6-2-13)21-17-9-11-18(12-10-17)22-16-7-3-14(19)4-8-16;1-16(2)11-3-7-13(8-4-11)18-15(17)19-14-9-5-12(16)6-10-14/h1-12H;3-10H,1-2H3/z401-23-41(33,38,34,38,35,40,36,41,37,40,37,41);101-1-22(13,19,14,19,15,21,16,22,17,21,18,22);331-1-45 InChIKey=HHFXSTKAFNJYEC-FBXUJDOCBA-N	 Block-copolymer
53		InChI=1B/C8H4/c1-2-8-5-3-7(1)4-6-8/h1-3,5H/z101-1-8(4,6,4,7,6,8) InChIKey=GQEZFUOMAFZPSC-BVCYNDGCBA-N	
54		InChI=1B/C3H5NO/c5-3-1-2-4-3/h1-2H2,(H,4,5)/z101-1-5(1,2,1,3,2,4) InChIKey=MNFORVFSTILPAW-GKSSZUKRBA-N	

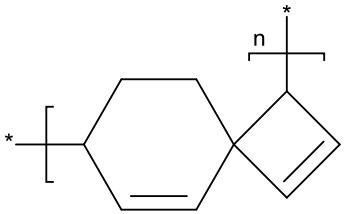
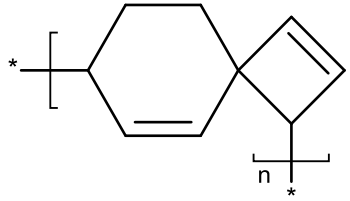
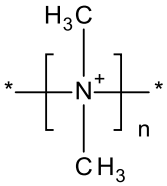
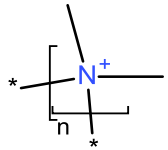
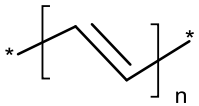
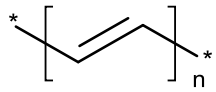
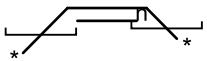
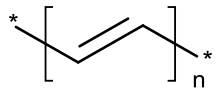
55		InChI=1B/C3H5NO/c5-3-1-2-4-3/h1-2H2,(H,4,5)/z101-1-5(1,2,1,3,2,4 InChIKey=MNFORVFSTILPAW-GKSSZUKRBA-N	
56		InChI=1B/C3H5NO/c5-3-1-2-4-3/h1-2H2,(H,4,5)/z101-1-5(1,2,1,3,2,4) InChIKey=MNFORVFSTILPAW-GKSSZUKRBA-N	
57		InChI=1B/C5H7NO3/c7-4-2-1-3(6-4)5(8)9/h3H,1-2H2,(H,6,7)(H,8,9)/z101-1-9(1,2,1,3,2,4,3,6)/t3-/m1/s1 InChIKey=ODHCTXKNWHHXJC-AQUWEPPIBA-N	
58		InChI=1B/C16H18N2O6/c19-13-9-7-10(12(16(23)24)8-11(9)15(21)22)14(20)18-6-4-2-1-3-5-17-13/h7-8H,1-6H2,(H,17,19)(H,18,20)(H,21,22)(H,23,24)/z101-1-	

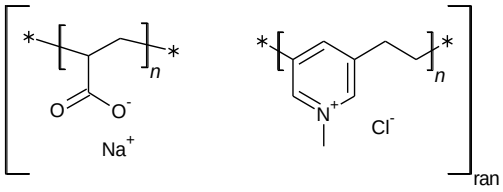
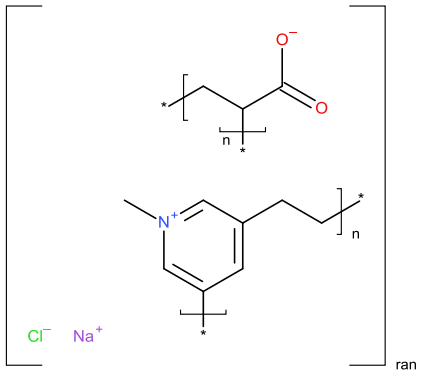
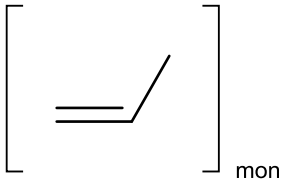
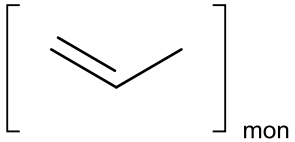
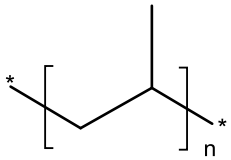
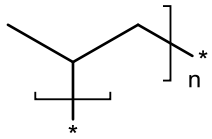
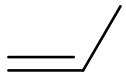
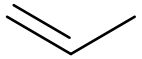
		24(1,2,1,3,2,4,3,5,4,6,5,17,6,18,9,13,10,14) InChIKey=NRXUBKMTMFYTLT-HYQSQRQEBA-N	
59		InChI=1B/Cl2NP/c1-4(2)3/z101-1-4(3.4) InChIKey=UXJHQBVRZUANLK-KBEGDARSBA-N	
60		InChI=1B/C3H4O2.Na/c1-2-3(4)5;/h2H,1H2,(H,4,5);/q;+1/p-1/z101-1-6(1.2) InChIKey=NNMHYFLPFNGQFZ-GJEZIXODBA-M	
61		InChI=1B/C18H13N/c1-2-4-16(5-3-1)19-17-10-6-14(7-11-17)15-8-12-18(19)13-9-15/h1-13H/z101-1-19(14,15,17,19,18,19) InChIKey=WUTRCYSYTJLPJT-DUOYJIBFBA-N	

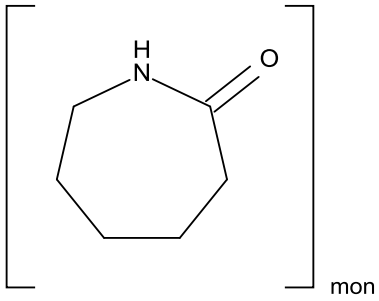
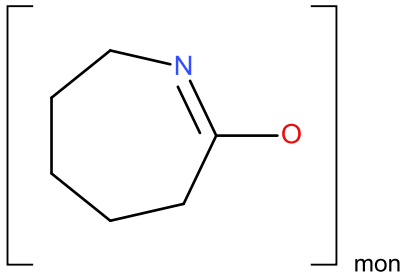
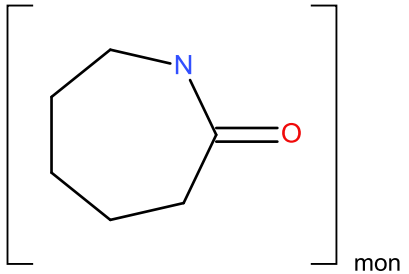
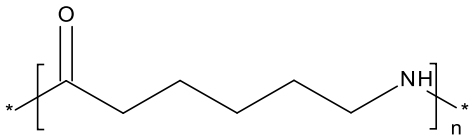
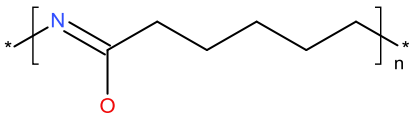
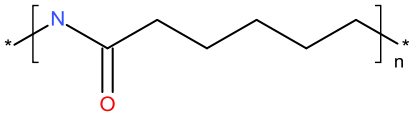
62		InChI=1B/C8H10N.HI/c1-3-8-6-4-5-7-9(8)2;/h3-7H,1H2,2H3;1H/q+1;/p-1/z101-1-10(1.3) InChIKey=DWMSYQDOPYSCJR-DDSQTIRRBA-M	
63		InChI=1B/C8H6O5/c9-7-5-1-2-6(13-5)8(10)12-4-3-11-7/h1-2H,3-4H2/z101-1-13(3,4,3,11,4,12,5,7,6,8,7,11,8,12) InChIKey=DVVGBNZLQNDSPA-XVRIZPPSBA-N	
64		InChI=1B/C4H8N2S4.Zn/c7-3(8)5-1-2-6-4(9)10;/h1-2H2,(H2,5,7,8)(H2,6,9,10);/q;+2/p-2/z101-1-11(1,5) InChIKey=AMHNZOICSMBGDH-SJNBWECRBA-L	

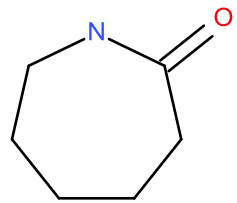
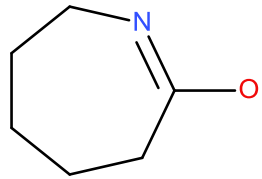
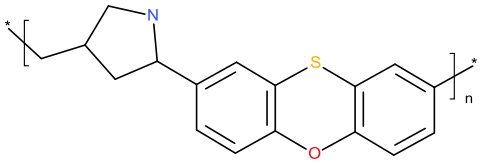
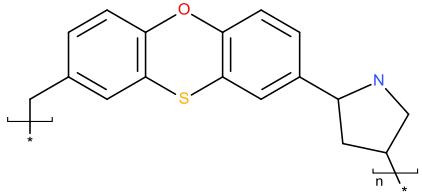
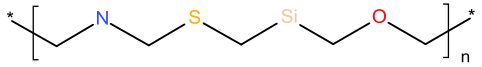
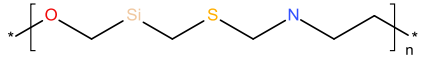
	Same with /RecMet	InChI=1B/C4H8N2S4.Zn/c7-3(8)5-1-2-6-4(9)10;/h1-2H2,(H2,5,7,8)(H2,6,9,10);/q;+2/p-2/z101-1-11(1,5)/rC4H6N2S4Zn/c1-2-6-4-9-11(10-4)7-3(5-1)8-11/h5-6H,1-2H2/z101-1-11(1,5) InChIKey=AMHNZOICSMBGDH-ZPQHTIIXBA-L	
65		InChI=1B/O2.2O.V/c1-2;;;/q-2;;-1;+2/z101-1-5(1,2) InChIKey=PIZXTTZLZREQOK-OXOBPHIVBA-N	
	Same with /RecMet	InChI=1B/O2.2O.V/c1-2;;;/q-2;;-1;+2/z101-1-5(1,2)/rO4V/c1-5(2)3-4-5/q-1/z101-1-5(3,4) InChIKey=PIZXTTZLZREQOK-LFGDFKGNBA-N	

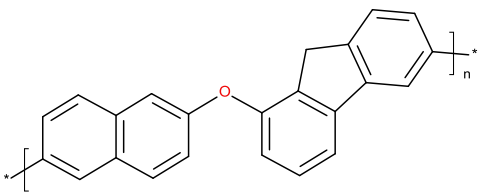
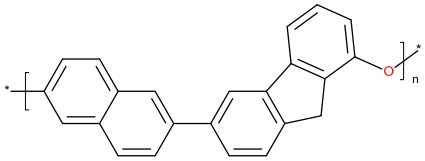
66		InChI=1B/C12H10N2O.2ClH.Cu/c1-3-9-7-15-8-10-4-2-6-12(14-10)11(5-1)13-9;;;/h1-6H,7-8H2;2*1H;/q;;;+2/p-2/z101-1-18(7,9,7,15,8,10,8,15,11,12) InChIKey=SJLWIGOSCVCHFV-VDYOQJQMBA-L	
	Same with /RecMet	InChI=1B/C12H10N2O.2ClH.Cu/c1-3-9-7-15-8-10-4-2-6-12(14-10)11(5-1)13-9;;;/h1-6H,7-8H2;2*1H;/q;;;+2/p-2/z101-1-18(7,9,7,15,8,10,8,15,11,12)/rC12H10Cl2CuN2O/c13-15(14)16-9-3-1-5-11(16)12-6-2-4-10(17(12)15)8-18-7-9/h1-6H,7-8H2/z101-1-18(7,9,7,18,8,10,8,18,11,12) InChIKey=SJLWIGOSCVCHFV-VIBYZETQBA-L	

67		InChI=1B/C9H10/c1-4-9-5-2-7(1)8(9)3-6-9/h1,3-4,6-8H,2,5H2/z101-1-9(7,8) InChIKey=RQMLRGLXQHWVQB-WCYUVGCXBA-N	
68		InChI=1B/C2H6N/c1-3-2/h1-2H3/q+1/z101-1-3(3) InChIKey=JLBDYIULKDRRIS-DJAGSJGEBA-N	
69		InChI=1B/C2H2/c1-2/h1-2H/z101-1-2(1.2) InChIKey=HSFWRNGVRCDJHI-SIBUVIRQBA-N	
70		InChI=1B/C2H2/c1-2/h1-2H/z101-1-2(1.2) InChIKey=HSFWRNGVRCDJHI-SIBUVIRQBA-N	

71		InChI=1B/C8H10N.C3H4O2.ClH.Na/c1-9-5-7-2-3-8(4-7)6-9;1-2-3(4)5;;/h4-6H,2-3H2,1H3;2H,1H2,(H,4,5);1H;/q+1;;;+1/p-2/z101-10-14,16(10.11);101-1-9,15(2,3,2,7,3,8);321-1-20 InChIKey=RHMLQKOGNDZGNR-QNNKLHLLBA-L	
72		InChI=1B/C3H6/c1-3-2/h3H,1H2,2H3/z200-1-3 InChIKey=QQONPFPTGQHPMA-DJNVOPQRBA-N	
73		InChI=1B/C3H6/c1-3-2/h3H,1H2,2H3/z101-1-3(1.3) InChIKey=QQONPFPTGQHPMA-KDIWSDSHBA-N	
74		InChI=1S/C3H6/c1-3-2/h3H,1H2,2H3 InChIKey=QQONPFPTGQHPMA-UHFFFAOYSA-N	

			Not a polymer
75		InChI=1B/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)/z200-1-8 InChIKey=JBKVHLHDHHXQEQ-ZMQGHSLKBA-N	
	Same with /FixedH	InChI=1B/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)/z200-1-8/f/h7H InChIKey=JBKVHLHDHHXQEQ-ZWBUTNKYBA-N	
76		InChI=1B/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)/z101-1-8(1,2,1,3,2,4,3,5,4,6,5,7) InChIKey=JBKVHLHDHHXQEQ-DZWZRWJOBA-N	
	Same with /FixedH	Structure: 76	

		InChI=1B/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8)/z101-1-8(1,2,1,3,2,4,3,5,4,6,5,7)/f/h7H InChIKey=JBKVHLHDHHXQEQ-VWJAHPQWBA-N	
77		InChI=1S/C6H11NO/c8-6-4-2-1-3-5-7-6/h1-5H2,(H,7,8) InChIKey=JBKVHLHDHHXQEQ-UHFFFAOYSA-N	 Not a polymer
78		InChI=1B/C17H15NOS/c1-3-14-16-7-10(1)5-11-6-13(18-9-11)12-2-4-15(19-14)17(8-12)20-16/h1-4,7-8,11,13,18H,5-6,9H2/z101-1-20(5,10,5,11,12,13) InChIKey=NVAVLZKTSMLUDJ-ALPRJYBXBA-N	
79		InChI=1B/C5H13NOSSi/c1-2-7-4-9-5-8-3-6-1/h6H,1-5,9H2/z101-1-9(1,2,1,6,2,7,3,6,3,8,4,7,4,9,5,8,5,9) InChIKey=ZCYFYWGBNPBAHW-RUMMOZOHBA-N	

80		InChI=1B/C23H14O/c1-2-20-21-12-17-6-7-18(21)13-22(20)23(3-1)24-19-9-8-15-10-14(17)4-5-16(15)11-19/h1-12H,13H2/z101-1-24(14,17,19,24,23,24) InChIKey=IKWWVSMTFFXWKZ-GSYXJMKABA-N	

Support of "large" molecules

General

Large molecules (up to 32767 atoms) are now supported, in experimental mode. (Besides the number of atoms, some other internal limits were relaxed)

Both inchi-1 executable and InChI Software Library (API) support this mode if option 'LargeMolecules' is supplied; otherwise, the software would still be limited to 1024 atoms.

Note that InChIs produced have a prefix 'InChI=1B' indicating beta status of these identifiers. Analogously, the flag character 'B' is used in InChIKey instead of 'S' (Standard) or 'N' (Non-standard).

Testing

The ability of InChI Software to handle large molecules was extensively tested using Protein Data Bank, PDB data, ~10^5 molecules.

For this purpose, PDB files were first converted using OpenBabel 2.3.2 to Molfile[4] format (chiefly V3000 with some V2000 format files). Then a maximal-contiguous fragment was extracted for each 'whole-molecule' and corresponding Molfiles were used in further experiments (otherwise we would get a false impression of how the algorithm scales up: InChI'fying an *n*-mer protein chain would require ~ *n*-fold the time for a monomer, too good to be true).

The resulting files were fed to inchi-1 or mol2inchi executables; in the latter case, additional experiments on re-numbering atoms followed by checks on whether the same InChI is produced were performed.

---

[4] Some molecules were lost in conversion. Some PDB .ent files caused OpenBabel hang (hence a timeout of 120 sec was specified); some produced zero-length Molfiles; etc.

The graph obtained of processing time vs. number of atoms is shown on Figure 1 below.

As observed, 99% of the longest fragments of the examined dataset of ~10^5 molecules from PDB (up to ~16,000 atoms) were converted to InChI in not more than 180 sec per entry. Average processing time was 3.8 sec (average size was 2400 atoms).

A cautionary note: InChI was not designed for molecules much larger than 1000 atoms. Though in principle algorithms should work in the last case, it is not completely clear how robust InChI canonicalization and normalization would be to size increase. Here is an issue on which a feedback from community is most welcome.

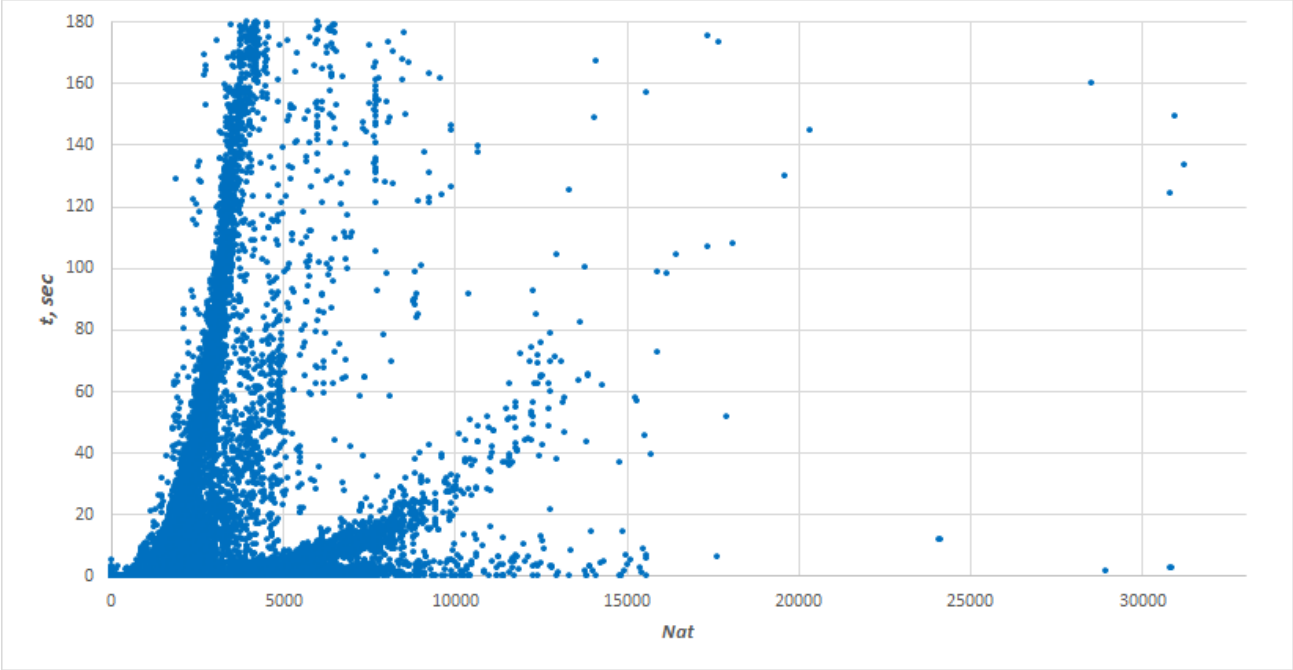


Figure 1. InChI generation time, sec, vs. number of atoms in molecules from PDB. Test machine: Xeon E3-1245V2 3.4 GHz, 7200 rpm HDD; single-threaded execution, 64-bit Linux OS (Ubuntu 14.04 LTS), mol2inchi executable; timeout for treating individual molecules has been set to 180 sec (stopped by timeout: 2624). Total number of points on this graph is 115,983.

Known issues

Renumbering tests performed to date involved generation of InChI for 100 random re-numberings of atomic systems for ~6x10^4 PDB molecules.

Two entries were found to give different InChIs on renumbering:

PDB codes `pdb2pfo` and `pdb4ged`.

New in InChI API

InChI Software v. 1.05 includes a significantly modified and extended Library, INCH API.

Generation of InChI for polymers requires use of the newly added Ex (extended functionality) API function GetINCHIEx(). Also added is native API support for direct Molfile to InChI conversion through a new function MakeINCHIFromMolfileText(). This function uses the same Molfile parser as inchi-1 executable thus ensuring that any correct caller of the InChI Library procedure will produce the same result as inchi-1.

A whole new set of API calls, IXA functions, is included. IXA stands for Extended InChI API. In particular, it contains new API procedures including low-level functions to deal with atoms, bonds, etc.

Accordingly, the two new demo programs mol2inchi and test_ixa are added in this release.

For the details, see the document “InChI_API_Reference.pdf”, included in this distribution.

Support of V3000 Molfile format

The ability to read and parse large (up to 32767 atoms) input files in Molfile V3000 format was added to the inchi-1 executable and the API procedure MakeINCHIFromMolfileText(). This is necessary for treating large molecules (previous versions supported only V2000 format limited to not more than 1000 atoms).

In addition, provisional support for extended features of Molfile V3000 was also added, both to inchi-1 and the InChI Software Library, API. This means that extended data (on haptic coordination bonds and stereo collections) are read and parsed; however, they are not used currently (as this requires significant modification of the Identifier itself, not just the Software).

Thread safety

As global and static objects may cause problems in a multi-threaded environment due to a possible race condition arising when more than one thread tries to access, they had to be removed from the InChI code. Corresponding changes were implemented.

In particular, global variables associated with sorting and timing were replaced by objects which are passed to all functions that need access to the related data. Also, several functions which uses static variables that were initialized upon first entry into the function have been modified in order to avoid a race condition.

Tests were performed for various applications using InChI Software Library and running up to 32 threads concurrently, and for datasets up to the whole Pubchem Compound set. No problems were encountered.

A final note: the facility implemented is *support of safe threading*, not the organization of multi-threaded execution itself – that is up to library users, i.e. application developers/software vendors.

Other

Changes to inchi-1 executable

Several convenience changes were made to inchi-1 executable.

LargeMolecules

A new option 'LargeMolecules' instructs the program to accept molecules containing more than 1024 (but less than 32767) atoms.

Polymers

A new option 'Polymers' instructs the program to accept polymer data in input V2000 Molfiles.

Output at Error an empty InChI

The new option 'Output at Error an empty InChI", `OutErrInChI` (`/OutErrInChI` under Windows, `-OutErrInChI` under Linux) instructs the program to output empty InChI and corresponding InChIKey if error occurs (default behaviour is output nothing).

The Standard InChI and InChIKey for empty entity are:

`InChI=1S//`

`InChIKey=MOSFIJXAXDLOML-UHFFFAOYSA-N`

The Non-standard InChI and InChIKey are:

`InChI=1//`

`InChIKey=MOSFIJXAXDLOML-UHFFFAOYNA-N`

Selection of record in SDF input file

Option `Record:N` (`/Record:N` under Windows, `-Record:N` under Linux) instructs the program to process only the N-th record of the input file in SDF format.

This convenience option is equivalent to combined options `Start:N` (start processing from N-th record) and `End:N` (end processing at N-th record).

Other

Implementation of wildcard expansion in Allow Multiple Input, AMI mode (like in `inchi-1 /AMI *.mol`) under Windows is improved, in order to tolerate possible large expansion volumes.

Testing

Regression testing

There were several minor fixes/changes made after software release v. 1.04. To ensure compatibility with previous release, the new software has been extensively regression-tested against standard and non-standard InChIs generated with v. 1.04 Software, in both Windows and Linux environments, with the various option combinations for standard and non-standard InChI (totalling 33 option sets; options new in v. 1.05 were intentionally omitted).

The test sets included:

1. "InChI_TestSet" (public). This is a test set of 2,186 structures which has been created previously and included in InChI Software distribution as "InChI validation suite". The structures include some very tricky and "chemically strange" ones, to verify InChI behaviour in exotic cases.

2. "MDB" (proprietary) 100,000 structures.

3. "MSL-NIST" (proprietary). 191,436 structures.

4. "NCI" (public). 249,081 structures from "NCI Open Database Compounds", retrieved from: http://cactus.nci.nih.gov/ncidb2/download.html

5. "ChEMBL20" (public). 1,456,020 structures from http://www.ebi.ac.uk/chembl - the version of ChEMBL is chembl_20. Retrieved on 2015-02-02.

6. "PubChem Compound" (public). 60,915,175 structures. Retrieved from PubChem on 2015-10-28.

No problems have been detected.

InChI round-trip test

The success rate for conversion of InChI to structures was measured using Rtrip option of `test_ixa` program or two-pass (structure to InChI followed by InChI to structure) execution of inchi-1. In this round trip, InChIs generated from Molfile-format records were then used to restore structures and re-create InChIs from those structures, via corresponding API calls; original and final InChIs were then compared.

The results for ChEMBL v. 20 (1,456,020) and PubChem Compound (60,915,175 molecules; checked in 64 bit Linux environmemt) are as follow:

Dataset	Success rate, %
Standard InChI	
ChEMBL	99.98
PubChem Compound	99.93
Non-standard (FixedH)	
ChEMBL	99.90
PubChem Compound	99.93
Non-standard (RecMet)	
ChEMBL	99.98
PubChem Compound	99.92
Non-standard (FixedH RecMet)	
ChEMBL	99.90
PubChem Compound	99.92

Distribution package

Binaries

This package includes 'command line' InChI executable and InChI API Library binaries (32 and 64 bit versions are supplied for both Windows and Linux).

Also included is winchi-1.exe, a graphical Windows application (a 32 bit version which will also run under 64 bit Windows).

File/directory INCHI-1-BIN

windows/	
winchi-1.exe	InChI graphical Windows application
windows/32bit/	
inchi-1.exe	InChI stand-alone command line executable, 32 bit
windows/32bit/dll/	
libinchi.dll	InChI dynamic-link library, 32 bit
windows/64bit/	
inchi-1.exe	InChI stand-alone command line executable, 64 bit
windows/64bit/dll	
libinchi.dll	InChI dynamic-link library, 64 bit
linux/	
linux/32bit/	
inchi-1.gz	InChI stand-alone command line executable, 32 bit; gzipped
linux/32bit/so/	
libinchi.so.1.05.00.gz	shared library for InChI API, 32 bit; gzipped
linux/64bit/	
inchi-1.gz	InChI stand-alone command line executable, 64 bit; gzipped
linux/64bit/so	
libinchi.so.1.05.00.gz	shared library for InChI API, 64 bit; gzipped

Note that InChI stand-alone executable inchi-1[.exe] does not require dll/so libraries.

Source codes and demo programs

InChI Software binaries are placed in the file/directory INCHI-1-BIN. Example data files are placed in the file/directory INCHI-1-TEST. Documentation is placed in the file/directory INCHI-1-DOC.

InChI Software source codes are placed in the file/directory INCHI-1-SRC. This file/directory also contains examples of InChI API usage, for C ('inchi_main', ‘mol2inchi’, ‘test_ixa’, see projects for MS Visual Studio 2008 in 'vc9' and for gcc:Linux in ‘gcc’ subdirs) and Python 3 ('python_sample'). Also supplied are InChI API Library source codes and related projects/makefiles.

The projects/makefiles necessary to build inchi-1 executable and demo programs are located within corresponding directories, as well as the source codes specific for these apps (see below). Note that a part of code which forms a common codebase is placed in special directory, INCHI-1-SRC/INCHI_BASE/src. To ensure proper build of InChI applications/library, the directory structure below (a tree under INCHI-1-SRC) should be preserved. Note also that, though InChI library ‘libinchi’ may be build using its own projects/makefiles (under INCHI-1-SRC/INCHI_API/libinchi/), it is automatically co-created upon building of any demo program.

INCHI-1-SRC/INCHI_BASE	
src	C source files - common codebase used to build both InChI Library and inchi-1 executable
INCHI-1-SRC/INCHI_EXE	
bin	directory where the binaries of inchi-1 executable are created/stored
inchi-1	a home directory for inchi-1 command-line executable
inchi-1/src	C source files specific for inchi-1 executable
inchi-1/gcc	gcc makefiles for inchi-1 executable (Linux, 64- and 32-bit)

inchi-1/vc9	MS VS2008 project for inchi-1 executable (Windows)

INCHI-1-SRC/INCHI_API

bin	directory where the binaries (of both library and all demo programs) are created/stored
libinchi	a home for InChI Software Library
libinchi/src	C source files specific for InChI Software Library
libinchi/gcc	gcc makefiles for libinchi library (Linux so)
libinchi/vc9	MS VS2008 project for libinchi library (Windows dll)
demos	a home for demo programs calling InChI library (API)
demos/inchi_main/src	C source files specific for inchi_main demo
demos/inchi_main/gcc	gcc makefiles for inchi_main demo program (Linux)
demos/inchi_main/vc9	MS VS2008 project for inchi_main demo program (Windows)
demos/mol2inchi/src	C source files specific for mol2inchi demo program
demos/mol2inchi/gcc	gcc makefiles for mol2inchi demo program (Linux)
demos/mol2inchi/vc9	MS VS2008 project for mol2inchi demo program (Windows)
demos/test_ixa/src	C source files specific for test_ixa demo program
demos/test_ixa /gcc	gcc makefiles for test_ixa demo program (Linux)
demos/test_ixa /vc9	MS VS2008 project for test_ixa demo program (Windows)
demos/python_sample	Python 3 source files specific for Python demo program

For details, please refer also to ‘readme.txt’ files in the directories.

Other

The documentation (Release Notes; InChI Technical Manual; InChI User Guide; InChI API Reference) in PDF format is supplied in the INCHI-1-DOC section of this distribution package.

Test data are supplied in the INCHI-1-TEST file/directory. Note the file pex.sdf there which contains Molfile presentation of the above-described example polymer data.

Acknowledgements

We are grateful to many people and organizations who are continuously supporting InChI development, in various forms. Special thanks are due to those who specifically contributed to the current release/testing, as listed in the accompanying file ‘External-contributors.pdf’.