

SUBSTANCES

VO.1 2012/07/22

A Chemical Database

Clemens NIEDERBERGER

<https://bitbucket.org/cgnieder/substances/>
contact@mychemistry.eu

English documentation

The **SUBSTANCES** package allows you to create a database like file that contains data of various chemicals. These data can be retrieved in the document. An index creation of the chemicals used in the document is directly supported.

Contents			
1 Licence and Requirements	1	5 Retrieving the Data	6
2 About	2	6 Additional Commands	8
3 Options	2	7 Create an Index	10
4 The Database	2	7.1 Formatting Commands	10
4.1 Declaring the Chemicals	2	7.2 Using makeidx	10
4.2 Available Fields	3	7.3 Using splitidx	11
4.2.1 Always Defined Fields	3	7.4 Using imakeidx	11
4.2.2 Style-dependend Fields	4	8 The Example Database	12
4.3 Define Custom Styles	5	Index of Chemicals	19
4.3.1 Background	5	Package Index	20
4.3.2 Declare New Fields or Change Existing Fields	5		

1 Licence and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the \LaTeX Project Public License, version 1.3 or later (<http://www.latex-project.org/lppl.txt>). The package has the status “maintained.”

`SUBSTANCES` loads and needs the following packages: `expl3`,¹ `xparse`,² `xtemplate`³ and `l3keys2e`.⁴ It also needs the chemistry package `chemmacros`.⁵

2 About

The `SUBSTANCES` package allows you to create a database like file that contains data of various chemicals. These data can be retrieved in the document. An index creation of the chemicals used in the document is directly supported.

3 Options

`SUBSTANCES` has only a few options:

- ▶ `draft` = `true|false` Default: false
If set to true all warnings will be errors.
- ▶ `final` = `true|false` Default: true
The opposite of `draft`.
- ▶ `index` = `true|false` Default: false
Add index entries when `\chem` is called, see section 7.
- ▶ `style` = `<style>` Default: default
Load specific style, see section 4.3.
- ▶ `strict` = `true|false` Default: false
If set to true all warnings will be errors. This option overwrites any `draft` or `final` option that is passed on by the document class.

4 The Database

4.1 Declaring the Chemicals

The data about substances are stored via the command

- ▶ `\DeclareSubstance{<id>}{<list of properties>}`

An entry could look like this:

```
1 \DeclareSubstance{NaCl}{  
2   name   = Sodiumchloride ,  
3   sum    = NaCl ,  
4   CAS    = 7647-14-5 ,  
5   mass   = 58.44 ,  
6   mp     = 801 ,  
7   bp     = 1465 ,
```

¹ CTAN: expl3 ² CTAN: xparse ³ CTAN: xtemplate ⁴ CTAN: l3keys2e ⁵ CTAN: chemmacros

```

8   phase    = solid ,
9   density  = 2.17
10  }

```

Such entries can either be declared in the document preamble or probably more useful in a file with the ending `.sub`. Such a file can be input in the document via

- ▶ `\LoadSubstances{<filename>}` → input the file *without* specifying the file ending.

Suppose you have the file `mysubstances.sub` then you input it in the document preamble via `\LoadSubstances{mysubstances}`.

4.2 Available Fields

4.2.1 Always Defined Fields

Below all fields defined by `SUBSTANCES` are listed.⁶

- ▶ `name` required
The IUPAC name of the substance. This is the only field that *has* to be used. The field's input is parsed with chemmacros' command `\iupac`.
- ▶ `sort` optional
If you plan to use the `index` option you should specify this field to get the sorting of the index right. This then creates index entries `\index{<sort field><name field>}`.
- ▶ `alt` optional
An alternative name. The field's input is parsed with chemmacros' command `\iupac`.
- ▶ `altsort` optional
This is the same as the `sort` field but for the alternative name.
- ▶ `CAS` optional
The Chemical Abstract Service (CAS) number. The input needs to be input in the form `<num>-<num>-<num>`.
- ▶ `PubChem` optional
The PubChem number.

The `CAS` field processes the number using the macro `\CAS{<number>}` which is defined like this:

```

1  \def\CAS#1-#2-#3{\iupac{#1\-#2\-#3}}
2  \NewDocumentCommand\CAS{m}{\@CAS#1}

```

You're free to redefine it to your needs.

⁶ Look in the file `substances-examples.sub` which is part of this package and should be in the same place as this documentation for example uses.

4.2.2 Style-dependent Fields

SUBSTANCES defines the style ‘default’ which is loaded if no other style has been specified. It defines the following additional fields and loads the packages chemfig⁷ and siunitx.⁸

- ▶ **formula** optional
The molecular formula of the substance. The field’s input is parsed with chemmacros’ command `\ch`.
- ▶ **structure** optional
The structural formula of the substance. The field’s input is parsed with chemfig’s command `\chemfig`.
- ▶ **mp** optional
The boiling point. The field’s entry is input into the siunitx command `\SI` in the following way: `\SI{<field>}{\celsius}`.
- ▶ **bp** optional
The melting point. The field’s entry is input into the siunitx command `\SI` in the following way: `\SI{<field>}{\celsius}`.
- ▶ **density** optional
The density. The field’s entry is input into the siunitx command `\SI` in the following way: `\SI{<field>}{\gram\per\cmc}`.
- ▶ **phase** optional
The state of aggregation.
- ▶ **pKa** optional
The pK_A value. The field’s entry is input into the siunitx command `\num`.
- ▶ **pKa1** optional
The first of several pK_A values. The field’s entry is input into the siunitx command `\num`.
- ▶ **pKa2** optional
The second of several pK_A values. The field’s entry is input into the siunitx command `\num`.
- ▶ **pKb** optional
The pK_B values. The field’s entry is input into the siunitx command `\num`.
- ▶ **pKb1** optional
The first of several pK_B values. The field’s entry is input into the siunitx command `\num`.
- ▶ **pKb2** optional
The second of several pK_B values. The field’s entry is input into the siunitx command `\num`.
- ▶ **pictograms** optional
The GHS pictograms. This field takes a list of pictogram names as they’re input into chemmacros’ command `\ghspic`.

⁷ CTAN: chemfig ⁸ CTAN: siunitx

- ▶ **H** optional
The H statements. This field takes a list of numbers as they're input into chemmacros' command `\ghs{h}{<number>}`.
- ▶ **P** optional
The P statements. This field takes a list of pictogram names as they're input into chemmacros' command `\ghs{p}{<number>}`.
- ▶ **EUH** optional
The EUH statements. This field takes a list of pictogram names as they're input into chemmacros' command `\ghs{h}{<number>}`.
- ▶ **LD50** optional
The Median Lethal Dose (LD₅₀) in mg kg⁻¹. The field's entry is input into the siunitx command `\SI` in the following way: `\SI{<field>}{\milli\gram\per\kilo\gram}`.

4.3 Define Custom Styles

4.3.1 Background

You might have other needs for fields than the ones defined by `SUBSTANCES` and the 'default' style. You can easily define your own style which means that you save a file with the name `substances-<style>.def`. In it you save the commands you need and use the command `\DeclareSubstanceProperty`, which is explained in the next section, to declare your own fields.

Please beware that this style file is loaded by the package when the `expl3` namespace is active. This means that all spaces are ignored and you need to use `~` if you need a space. It also means that `_` and `:` are letters (as is `@`). Should you ever decide to call `\ExplSyntaxOff` to turn off the `expl3` namespace *don't* forget to use `\ExplSyntaxOn` at the end of the file!

4.3.2 Declare New Fields or Change Existing Fields

You might want other fields or change the definition of the predefined ones. For this there's

- ▶ `\DeclareSubstanceProperty*{<field name>}[<pre code>][<post code>]`

This command declares a new property field for a substance. The star makes the property a required one which means an error will be issued if a substance is declared without it. The optional arguments `<pre code>` and `<post code>` specify any code that should be input directly before or after the field entry, respectively. The `<pre code>` may end with a command that takes one mandatory argument. In this case the field entry will be its argument.

The following example would define a field `EC` which uses a custom command to parse the field entry. The European Commission Number (EC) is assigned to chemical substances for regulatory purposes within the European Union by the regulatory authorities.

```

1 \makeatletter
2 \def\@EC#1-#2-#3{#1-#2-#3}
3 \newcommand*\EC[1]{\@EC#1}
4 \makeatother
5 \DeclareSubstanceProperty{EC}[\EC]

```

For further examples of the usage of pre and post code look at the definition of the `name` and the `mp` field:

```

1 \DeclareSubstanceProperty*{name}[\iupac]
2 \DeclareSubstanceProperty{mp}[\SI][{\celsius}]

```

5 Retrieving the Data

There are two commands defined by `SUBSTANCES` that allow the retrieving of the data. The command `\chem` is intended as user command, the command `\GetSubstanceProperty` can be used to define your own user command (perhaps in your own style file, see section 4.3).

- ▶ `\chem*{[<pre>] [<post>]{<id>}[<property>]}`
- ▶ `\GetSubstanceProperty{<id>}{<property>}`

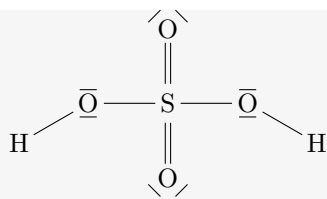
If the command `\chem` is called without the `[<property>]` argument the name entry will be called. The starred version calls the `alt` entry if it is defined and the name entry otherwise. The arguments `[<pre>]` and `[<post>]` add arbitrary input before or after the output, respectively.

All of the next examples use the data defined in the file `substances-examples.sub` that is part of this package, see section 8.

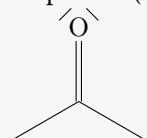
```

1 \chem{H2SO4}[structure] \newline
2 \chem{H2SO4} has the boiling point \(\chem[T_b =]{H2SO4}[bp]\) and a
3 density of \(\chem[\rho =]{H2SO4}[density]\).
4
5 Compare the melting points of methane and ethane,
6 \(\chem[T_m =]{methane}[mp]\) and \(\chem[T_m =]{ethane}[mp]\),
7 with the boiling points \(\chem[T_b =]{methane}[bp]\) and
8 \(\chem[T_b =]{ethane}[bp]\).
9
10 \chem{NaCl} has the \ac{CAS} number \chem{NaCl}[CAS].
11
12 \chem{acetone} (\chem*{acetone}) is the most simple ketone:
13
14 \chem{acetone}[structure]

```



Sulfuric Acid has the boiling point $T_b = 279.6\text{ }^\circ\text{C}$ and a density of $\rho = 1.8356\text{ g cm}^{-3}$.
 Compare the melting points of methane and ethane, $T_m = -182\text{ }^\circ\text{C}$ and $T_m = -183\text{ }^\circ\text{C}$, with
 the boiling points $T_b = -162\text{ }^\circ\text{C}$ and $T_b = -89\text{ }^\circ\text{C}$.
 Sodiumchloride has the CAS number 7647-14-5.
 Propanone (Acetone) is the most simple ketone:



The following code creates table 1.

```

1 \begin{table}[htp]
2 \centering\chemsetup[ghsystem]{hide}
3 \sisetup{scientific-notation=fixed,fixed-exponent=0,per-mode=
  symbol}
4 \begin{tabular}{l>{\raggedright\arraybackslash}p{.6\linewidth}}
5 \toprule
6 name & & \chem{methane} \\
7 formula & & \chem{methane}[formula] \\
8 & & \chem{methane}[structure] \\
9 \midrule
10 CAS & & \chem{methane}[CAS] \\
11 PubChem & & \chem{methane}[PubChem] \\
12 \midrule
13 boiling point & & \chem{methane}[bp] \\
14 melting point & & \chem{methane}[mp] \\
15 density & & \chem{methane}[density] \\
16 molar mass & & \chem{methane}[mass] \\
17 \midrule
18 & & \chem{methane}[pictograms] \\
19 H statements & & \chem{methane}[H] \\
20 P statements & & \chem{methane}[P] \\
21 \bottomrule
22 \end{tabular}
23 \caption{\label{tab:methane}All properties of \chem{methane} that
  have been
24 saved in the example database.}
25 \end{table}

```


name	Methane
formula	CH ₄ $\begin{array}{c} \text{H} \\ \\ \text{H} - \text{C} - \text{H} \\ \\ \text{H} \end{array}$
CAS	74-82-8
PubChem	297
boiling point	-162 °C
melting point	-182 °C
density	0.000 72 g/cm ³
molar mass	16.04 g/mol
	
H statements	H220
P statements	P210, P377, P381, P410 + P403

Table 1: All properties of Methane that have been saved in the example database.

6 Additional Commands

SUBSTANCES provides a few commands that maybe are useful in building custom macros for styles. A field exists if it has been defined with `\DeclareSubstanceProperty` regardless if it has been used or not. A substance exists if it has been defined with `\DeclareSubstance`.

- ▶ `\GetSubstanceProperty{<id>}{<field>}` → You know that already: retrieve the property specified in `<field>` for a given substance. This command is *not* expandable.
- ▶ `\RetrieveSubstanceProperty{<id>}{<field>}` → Same as `\GetSubstanceProperty` but expandable.
- ▶ `\ForAllSubstancesDo{<code>}` → Loops through all existing substances. Inside `<code>` #1 may be used to refer to the `<id>` of the current substance. This command is expandable.
- ▶ `\AllSubstancesSequence` → A sequence of all substances. This is a sequence of balanced groups each containing the `<id>` of a substance. This command is expandable.
- ▶ `\AllSubstancesClist` → A comma separated list of all substances. Every `<id>` is separated from the next with a comma. This command is expandable.
- ▶ `\IfSubstancePropertyTF{<id>}{<field>}{<>true code>}{<>false code>}` → Tests if the property `<field>` is defined for the substance `<id>` and returns either `<>true code>` or `<>false code>`.

code>. This command is expandable.

- ▶ `\IfSubstancePropertyT{<id>}{<field>}{<true code>}` → Tests if the property <field> is defined for the substance <id> and returns <true code> if it is. This command is expandable.
- ▶ `\IfSubstancePropertyF{<id>}{<field>}{<false code>}` → Tests if the property <field> is defined for the substance <id> and returns <false code> if it isn't. This command is expandable.
- ▶ `\IfSubstanceFieldTF{<field>}{<true code>}{<false code>}` → Tests if the property <field> exists and returns either <true code> or <false code>. This command is expandable.
- ▶ `\IfSubstanceFieldT{<field>}{<true code>}` → Tests if the property <field> exists and returns <true code> if it does. This command is expandable.
- ▶ `\IfSubstanceFieldF{<field>}{<false code>}` → Tests if the property <field> exists and returns <false code> if it doesn't. This command is expandable.
- ▶ `\IfSubstanceExistTF{<id>}{<true code>}{<false code>}` → Tests if the property <field> exists and returns either <true code> or <false code>. This command is expandable.
- ▶ `\IfSubstanceExistT{<id>}{<true code>}` → Tests if the substance <id> exists and returns <true code> if it does. This command is expandable.
- ▶ `\IfSubstanceExistF{<id>}{<false code>}` → Tests if the substance <id> exists and returns <false code> if it doesn't. This command is expandable.

```
1 Just to demonstrate how these commands can be used. And to get
2 our demonstration index filled.\par
3 \newcounter{substances}
4 \ForAllSubstancesDo{%
5   \ifnum0=\value{substances}\relax
6   \else, \fi
7   \stepcounter{substances}%
8   \chem{#1}%
9   \IfSubstancePropertyT{#1}{alt}{
10    (\chem*{#1})}}
```

Just to demonstrate how these commands can be used. And to get our demonstration index filled.

Sodiumchloride, Hydrochloric Acid, Nitric Acid, Sulfuric Acid, Methane, Ethane, Propane, Butane (*n*-Butane), Pentane (*n*-Pentane), Hexane (*n*-Hexane), Heptane (*n*-Heptane), Octane (*n*-Octane), Nonane (*n*-Nonane), Decane (*n*-Decane), Propanone (Acetone)

7 Create an Index

When `SUBSTANCES` is called with `index = true` the command `\chem` will add index entries each time it is used. In this case the entries of the fields `name`, `sort`, `alt` and `altsort` will be expanded during the process. You should keep that in mind if some error arises. It might be due to a `\textbf` or similar in your database. In this case you either need to replace it with some robust command or put a `\noexpand` in front of it.

Alternative names as specified in the `alt` also get an index entry with a reference to the one of the corresponding `name` field. The entry of the `name` field in this case gets the `alt` name appended in braces.

This behaviour is not customizable for the time being. It is planned for future versions of this package, though.

As a demonstration an index for all chemicals used in this documentation is created with the help of the package `imakeidx`.⁹

7.1 Formatting Commands

The index entries are formatted with the following commands. You can redefine them to your needs. If you do make sure they have the same number of required arguments and are expandable!

- ▶ `\SubstanceIndexNameEntry#1#2{#1@#2}` → Formats the name if no `alt` field is given. #1 refers to the `sort` field entry and #2 refers to the `name` field entry.
- ▶ `\SubstanceIndexNameAltEntry#1#2#3{#1@#2 (#3)}` → Formats the name if also the `alt` field is given. #1 refers to the `sort` field entry, #2 refers to the `name` field entry, and #3 to the `alt` field entry.
- ▶ `\SubstanceIndexAltEntry#1#2#3{#1@#3|see#2}` → Formats the entry for the `alt` field. #1 refers to the `altsort` field entry, #2 refers to the `name` field entry, and #3 to the `alt` field entry.

7.2 Using `makeidx`

Using the option `index = true` with the standard way to create an index will add the entries `\index{<name>}` to the index. This means you would mix them with other entries if you have any. Below a sample document is shown that needs to be compiled with `pdflatex`, `makeindex` and again with `pdflatex`.

```
1 \documentclass{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[index]{substances}
4 \LoadSubstances{substances-examples}
5
6 \usepackage{makeidx}
7 \makeindex
8 \begin{document}
```

⁹ CTAN: `imakeidx`

```

9
10 \newcounter{substances}
11 \ForAllSubstancesDo{%
12   \ifnum0=\value{substances}\relax
13   \else, \fi
14   \stepcounter{substances}\chem{#1}}
15
16 \printindex
17 \end{document}

```

7.3 Using splitidx

Maybe a separate index for the chemicals will make more sense. In this case you could use the package `splitidx`.¹⁰ `SUBSTANCES` will recognize this and create `\sindex[\jobname-chem]{<name>}` entries each time `\chem` is used. The sample document below needs to be compiled with `pdflatex`, `splitindex` and again with `pdflatex`.

```

1 \documentclass{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[index]{substances}
4 \LoadSubstances{substances-examples}
5
6 \usepackage{splitidx}
7 \makeindex
8 \newindex[Chemicals]{\jobname-chem}
9 \begin{document}
10
11 \newcounter{substances}
12 \ForAllSubstancesDo{%
13   \ifnum0=\value{substances}\relax
14   \else, \fi
15   \stepcounter{substances}\chem{#1}}
16
17 \printindex[\jobname-chem]
18 \end{document}

```

7.4 Using imakeidx

Another way to create multiple indexes is the package `imakeidx`. `SUBSTANCES` recognizes its usage and creates index entries `\index[\jobname-chem]{<name>}`. The sample document below needs to be compiled with `pdflatex --shell-escape`.

```

1 \documentclass{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[index]{substances}

```

¹⁰ CTAN: `splitidx`

```

4 \LoadSubstances{substances-examples}
5
6 \usepackage{imakeidx}
7 \makeindex[name=\jobname-chem,title=Chemicals]
8 \begin{document}
9
10 \newcounter{substances}
11 \ForAllSubstancesDo{%
12   \ifnum0=\value{substances}\relax
13   \else, \fi
14   \stepcounter{substances}\chem{#1}}
15
16 \printindex[\jobname-chem]
17 \end{document}

```

8 The Example Database

The following code shows the example database `substances-examples.sub` that is part of this package.

```

1 % -----
2 % the SUBSTANCES package
3 %
4 %   A Chemical Database
5 %
6 % -----
7 % Clemens Niederberger
8 % Web:   https://bitbucket.org/cgnieder/substances/
9 % E-Mail: contact@mychemistry.eu
10 % -----
11 % Copyright 2012 Clemens Niederberger
12 %
13 % This work may be distributed and/or modified under the
14 % conditions of the LaTeX Project Public License, either version 1.3
15 % of this license or (at your option) any later version.
16 % The latest version of this license is in
17 %   http://www.latex-project.org/lppl.txt
18 % and version 1.3 or later is part of all distributions of LaTeX
19 % version 2005/12/01 or later.
20 %
21 % This work has the LPPL maintenance status ‘maintained’.
22 %
23 % The Current Maintainer of this work is Clemens Niederberger.
24 % -----
25 % The substances package consists of the files
26 % - substances.sty, substances-default.def, substances-examples.sub,
27 %   substances_en.tex, substances_en.pdf, README
28 % -----
29 % If you have any ideas, questions, suggestions or bugs to report, please
30 % feel free to contact me.
31 % -----
32 %

```

```

33 % example database to the package 'substances'
34 %
35 \DeclareChemIUPAC\normal{\textit{n}}
36 \DeclareSubstance{NaCl}{
37   name      = Sodium\|chloride ,
38   sort      = Sodiumchloride ,
39   formula   = NaCl ,
40   CAS       = 7647-14-5,
41   mass      = 58.44 ,
42   mp        = 801 ,
43   bp        = 1465 ,
44   phase     = solid ,
45   density   = 2.17
46 }
47
48 \DeclareSubstance{HCl}{
49   name      = Hydro\|chloric Acid ,
50   formula   = HCl ,
51   CAS       = 7647-01-0 ,
52   pictograms = {acid,exclam} ,
53   H         = {314,335} ,
54   P         = {260,301+330+331,303+361+353,305+351+338,405,501} ,
55   mass      = 36.46 ,
56   density   = 1.19 ,
57   mp        = -30
58 }
59
60 \DeclareSubstance{HNO3}{
61   name      = Nitric Acid ,
62   sort      = Nitric Acid ,
63   formula   = HNO3 ,
64   CAS       = 7697-37-2 ,
65   PubChem   = 944 ,
66   mass      = 63.01 ,
67   density   = 1.51 ,
68   mp        = -42 ,
69   bp        = 86 ,
70   pKa       = -1.37 ,
71   pictograms = {flame-0,acid} ,
72   H         = {272,314} ,
73   P         = {220,280,305+351+338,310}
74 }
75
76 \DeclareSubstance{H2SO4}{
77   name      = Sulfuric Acid ,
78   sort      = Sulfuric Acid ,
79   formula   = H2SO4 ,
80   structure = {H-[:30]\Lewis{26,0}-S(=[2]\Lewis{13,0})(=[6]\Lewis{57,0})-\Lewis{26,0}-[:30]H} ,
81   CAS       = 7664-93-9 ,
82   PubChem   = 1118 ,
83   mass      = 98.08 ,
84   density   = 1.8356 ,
85   mp        = 10.38 ,

```

```

86   bp           = 279.6 ,
87   phase        = liquid ,
88   pKa          = -3.0 ,
89   pKa1         = -3.0 ,
90   pKa2         = 1.9 ,
91   pictograms   = acid ,
92   H            = 314 ,
93   P            = {280,301+330+331,309,310,305+351+338} ,
94   LD50         = 510
95 }
96
97 \DeclareSubstance{methane}{
98   name         = Methane ,
99   sort         = Methane ,
100  formula      = CH4 ,
101  structure     = H-C(-[2]H)(-[6]H)-H ,
102  CAS          = 74-82-8 ,
103  PubChem      = 297 ,
104  pictograms   = {flame,bottle} ,
105  H            = 220 ,
106  P            = {210,377,381,410+403} ,
107  mass         = 16.04 ,
108  density      = 0.72e-3 ,
109  mp           = -182 ,
110  bp           = -162 ,
111  phase        = gaseous
112 }
113
114 \DeclareSubstance{ethane}{
115   name         = Ethane ,
116   sort         = Ethane ,
117   formula      = C2H6 ,
118   structure     = H-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-H ,
119   CAS          = 74-84-0 ,
120   PubChem      = 6324 ,
121   pictograms   = {flame,bottle} ,
122   H            = 220 ,
123   P            = {210,377,381,403} ,
124   mass         = 30.07 ,
125   density      = 0.72e-3 ,
126   mp           = -183 ,
127   bp           = -89 ,
128   phase        = gaseous
129 }
130
131 \DeclareSubstance{propane}{
132   name         = Propane ,
133   sort         = Propane ,
134   formula      = C3H8 ,
135   structure     = H-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-H ,
136   CAS          = 74-98-6 ,
137   pictograms   = {flame,bottle} ,
138   H            = 220 ,
139   P            = {201,377,381,403} ,

```

```

140 mass      = 44.10 ,
141 density   = 2.01e-3 ,
142 mp        = -188 ,
143 bp        = -42 ,
144 phase     = gaseous
145 }
146
147 \DeclareSubstance{butane}{
148 name      = Butane ,
149 sort      = Butane ,
150 alt       = \normal\ -Butane ,
151 altsort   = n-Butane ,
152 formula   = C4H10 ,
153 structure = H-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-H ,
154 CAS       = 106-97-8 ,
155 PubChem   = 7843 ,
156 pictograms = {flame,bottle} ,
157 H         = {220,280} ,
158 P         = {201,377,381,403} ,
159 mass      = 58.12 ,
160 density   = 2.71e-3 ,
161 mp        = -138.3 ,
162 bp        = -0.5 ,
163 phase     = gaseous
164 }
165
166 \DeclareSubstance{pentane}{
167 name      = Pentane ,
168 sort      = Pentane ,
169 alt       = \normal\ -Pentane ,
170 altsort   = n-Pentane ,
171 formula   = C5H12 ,
172 structure = H-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-C(-[2]H)(-[6]H)-H ,
173 CAS       = 109-66-0 ,
174 PubChem   = 8003 ,
175 pictograms = {flame,health,exclam,aqpol} ,
176 H         = {225,304,336,411} ,
177 EUH       = 066 ,
178 P         = {273,301+310,331,403+235} ,
179 mass      = 72.15 ,
180 density   = 0.63 ,
181 mp        = -130 ,
182 bp        = 36 ,
183 phase     = liquid
184 }
185
186 \DeclareSubstance{hexane}{
187 name      = Hexane ,
188 sort      = Hexane ,
189 alt       = \normal\ -Hexane ,
190 altsort   = n-Hexane ,
191 formula   = C6H14 ,
192 structure = -[:30]-[:30]-[:30]-[:30]-[:30] ,

```

```

193 CAS      = 110-54-3 ,
194 PubChem  = 8058 ,
195 pictograms = {flame,health,exclam,aqpol} ,
196 H        = {225,361f,304,373,315,336,411} ,
197 P        = {210,240,273,301+310,331,302+352,403+235} ,
198 mass     = 86.18 ,
199 density  = 0.66 ,
200 mp       = -95 ,
201 bp       = 69 ,
202 phase    = liquid
203 }
204
205 \DeclareSubstance{heptane}{
206   name     = Heptane ,
207   sort     = Heptane ,
208   alt      = \normal\ -Heptane ,
209   altsort  = n-Heptane ,
210   formula  = C7H16 ,
211   structure = -[:30]-[:30]-[:30]-[:30]-[:30]-[:30] ,
212   CAS      = 142-82-5 ,
213   PubChem  = 8900 ,
214   pictograms = {flame,health,exclam,aqpol} ,
215   H        = {225,304,315,336,410} ,
216   P        = {210,273,301+310,331,302+352,403+235} ,
217   mass     = 100.21 ,
218   density  = 0.68 ,
219   mp       = -91 ,
220   bp       = 98 ,
221   phase    = liquid
222 }
223
224 \DeclareSubstance{octane}{
225   name     = Octane ,
226   sort     = Octane ,
227   alt      = \normal\ -Octane ,
228   altsort  = n-Octane ,
229   formula  = C8H18 ,
230   structure = -[:30]-[:30]-[:30]-[:30]-[:30]-[:30]-[:30] ,
231   CAS      = 111-65-9 ,
232   PubChem  = 356 ,
233   pictograms = {flame,health,exclam,aqpol} ,
234   H        = {225,304,315,336,410} ,
235   P        = {210,273,301+330+331,302+352} ,
236   mass     = 114.23 ,
237   density  = 0.70 ,
238   mp       = -56.8 ,
239   bp       = 126 ,
240   phase    = liquid
241 }
242
243 \DeclareSubstance{nonane}{
244   name     = Nonane ,
245   sort     = Nonane ,
246   alt      = \normal\ -Nonane ,

```



```

247   altsort      = n-Nonane ,
248   formula      = C9H20 ,
249   structure    = -[:30]-[:30]-[:30]-[:30]-[:30]-[:30]-[:30]-[:30] ,
250   CAS          = 111-84-2 ,
251   PubChem      = 8141 ,
252   pictograms   = {flame,exclam,health} ,
253   H            = {226,304,315,319,332,336,413} ,
254   P            = {261,301+310,305+351+338,331} ,
255   mass         = 128.26 ,
256   density      = 0.72 ,
257   mp           = -54 ,
258   bp           = 151 ,
259   phase        = liquid
260 }
261
262 \DeclareSubstance{decane}{
263   name          = Decane ,
264   sort          = Decane ,
265   alt           = \normal-Decane ,
266   altsort       = n-Decane ,
267   formula       = C10H22 ,
268   structure     = -[:30]-[:30]-[:30]-[:30]-[:30]-[:30]-[:30]-[:30]-[:30] ,
269   CAS           = 124-18-5 ,
270   PubChem       = 15600 ,
271   pictograms   = {flame,health} ,
272   H             = {226,304} ,
273   P             = {210,260,262,301+310,331} ,
274   mass          = 142.29 ,
275   density       = 0.73 ,
276   mp           = -29.7 ,
277   bp           = 174 ,
278   phase        = liquid
279 }
280
281 \DeclareSubstance{acetone}{
282   name          = Propanone ,
283   sort          = Propanone ,
284   alt           = Acetone ,
285   altsort       = Acetone ,
286   formula       = C3H6O ,
287   structure     = {-[:30](=[2]\Lewis{13,0})-[:30]} ,
288   CAS           = 67-64-1 ,
289   PubChem       = 180 ,
290   mass          = 58.08 ,
291   density       = 0.79 ,
292   mp           = -95 ,
293   bp           = 56 ,
294   pictograms   = {flame,exclam} ,
295   H             = {225,319,336} ,
296   EUH           = {066} ,
297   P             = {210,233,305+351+338} ,
298   LD50         = 5800
299 }
300

```

301 \endinput

Index of Chemicals

Acetone, *see* Propanone

Butane (*n*-Butane), 9

Decane (*n*-Decane), 9

Ethane, 7, 9

Heptane (*n*-Heptane), 9

Hexane (*n*-Hexane), 9

Methane, 7–9

n-Butane, *see* Butane

n-Decane, *see* Decane

n-Heptane, *see* Heptane

n-Hexane, *see* Hexane

n-Nonane, *see* Nonane

n-Octane, *see* Octane

n-Pentane, *see* Pentane

Nitric Acid, 9

Nonane (*n*-Nonane), 9

Octane (*n*-Octane), 9

Pentane (*n*-Pentane), 9

Propane, 9

Propanone (Acetone), 7, 9

Sodiumchloride, 7, 9

Sulfuric Acid, 7, 9

Package Index

A	
<code>\AllSubstancesClist</code>	8
<code>\AllSubstancesSequence</code>	8
<code>alt</code>	3, 10
<code>altsort</code>	3, 10
B	
<code>bp</code>	4
C	
<code>CAS</code>	3
<code>\CAS</code>	3
<code>\celsius</code>	4
<code>\ch</code>	4
<code>\chem</code>	2, 6, 10 f.
<code>chemfig</code>	4
<code>\chemfig</code>	4
<code>chemmacros</code>	2–5
<code>\cmc</code>	4
D	
<code>\DeclareSubstance</code>	2, 8
<code>\DeclareSubstanceProperty</code>	5, 8
<code>density</code>	4
<code>draft</code>	2
E	
<code>EC</code>	5
<code>EUH</code>	5
<code>expl3</code>	2
<code>\ExplSyntaxOff</code>	5
<code>\ExplSyntaxOn</code>	5
F	
<code>final</code>	2
<code>\ForAllSubstancesDo</code>	8
<code>formula</code>	4
G	
<code>\GetSubstanceProperty</code>	6, 8
<code>\ghs</code>	5
<code>\ghspic</code>	4
<code>\gram</code>	4 f.
H	
<code>H</code>	5
I	
<code>\IfSubstanceExistF</code>	9
<code>\IfSubstanceExistT</code>	9
<code>\IfSubstanceExistTF</code>	9
<code>\IfSubstanceFieldF</code>	9
<code>\IfSubstanceFieldT</code>	9
<code>\IfSubstanceFieldTF</code>	9
<code>\IfSubstancePropertyF</code>	9
<code>\IfSubstancePropertyT</code>	9
<code>\IfSubstancePropertyTF</code>	8
<code>imakeidx</code>	10 f.
<code>index</code>	2 f., 10
<code>\index</code>	3, 10 f.
<code>\iupac</code>	3
J	
<code>\jobname</code>	11
K	
<code>\kilo</code>	5
L	
<code>l3keysze</code>	2
<code>LD50</code>	5
<code>\LoadSubstances</code>	3
M	
<code>\milli</code>	5
<code>mp</code>	4, 6
N	
<code>name</code>	3, 6, 10
<code>\noexpand</code>	10
<code>\num</code>	4
P	
<code>P</code>	5
<code>\per</code>	4 f.
<code>phase</code>	4
<code>pictograms</code>	4
<code>pKa</code>	4

pKa1	4	splitidx	11
pKa2	4	strict	2
pKb	4	structure	4
pKb1	4	style	2
pKb2	4	\SubstanceIndexAltEntry	10
PubChem	3	\SubstanceIndexNameAltEntry	10
		\SubstanceIndexNameEntry	10
R		T	
\RetrieveSubstanceProperty	8	\textbf	10
S		X	
\SI	4 f.	xparse	2
\sindex	11	xtemplate	2
siunitx	4 f.		
sort	3, 10		