The **cooking-units** package*

Ben Vitecek
b.vitecek@gmx.at

2020/01/13

Abstract

This package enables user to globally format units, to switch between them and change your recipes to a given number of persons.
For not implemented units or differences between Imperial and U.S. unit you may have a look at appendix B.
It should be used for light-hearted things like cookery books (and not e.g. scientific texts; use e.g. siunitx for those).
Please read through the section “Important Changes”

Contents

1 Introduction 2
1.1 Supported languages . . . . . . . . . . . . . . . . . . . . . . . . . 2

2 The Commands 3

3 Label & refs: Changing the amount of the recipe 4
3.1 Rounding temperatures . . . . . . . . . . . . . . . . . . . . . . . . . 5

4 Predefined units & some notes 6

5 Defining units 6

6 Defining options to change units 8

7 Language support 12
7.1 Phrases . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

*This document corresponds to Benedikt Vitecek v1.46, dated 2020/01/13.
# Options

## Load time options

## Normal options

### Unit Specific options

### Command behavior

### Hooks

### Input and Outputs

### Rounding options

### Fractions

### Spaces

### label & refs

## Weird options

# Bugs & Feedback

# Bens Einheitensammelsurium (Bens unit Almanac)

## Translations

### English

### american

### German

### French

## US, Imperial and Other units

# Change History

# Index

## Introduction

While writing on a cookery book I used – for some reasons whatsoever – three different units for weight: kilogram (kg), gram (g) and decagram (dag, or older: dkg). Later my mother told me that she doesn’t like it if a cookery book uses more than two different units (for weight in this case). Happily I hardly used Decagram and therefore didn’t have many problems changing the units. But, well ... I am using \LaTeX{} and changing those units by hand seemed not very \LaTeX{}-like, so I started writing some code to convert units. I expanded the code, rewrote it in \LaTeX{}3 (which is much more pleasant than \LaTeX{}2ε) and here it is.
1.1 Supported languages

- German
- English
- French (currently suboptimal\(^1\))

Want to contribute a new language or make a correction to an existing one? See section 9 for more details. Wanna just check the existing translations? See appendix A.

2 The Commands

This package offers the following commands for unit printing (and converting):

- \texttt{\textbackslash cunum}<\langle label\rangle>[\langle options\rangle]{\langle amount\rangle}[\langle space\rangle]{\langle unit-key\rangle}
- \texttt{\textbackslash cutext}<\langle label\rangle>[\langle options\rangle]{\langle amount\rangle}[\langle unit-key\rangle]
- \texttt{\textbackslash Cutext}<\langle label\rangle>[\langle options\rangle]{\langle amount\rangle}[\langle unit-key\rangle]
- \texttt{\textbackslash cuam}<\langle label\rangle>[\langle options\rangle]{\langle amount\rangle}
- \texttt{\textbackslash cusetup}\{\langle options\rangle\}

Numbers and units are printed using \texttt{\textbackslash cunum}. The numerical part can interpret \_ and / as (mixed) fractions and -- as a separator for ranges; to convert units use the option \langle old-unit\rangle=\langle new-unit\rangle\(^2\). It furthermore allows the sign ? to be used as a placeholder for not known amounts and raises a warning to remind that this amount needs a check-up\(^3\). \langle space\rangle adds a space between the number and the unit using \texttt{\phantom}.

For a list of predefined units have a look at table 1. \langle label\rangle is explained in section 3.

\begin{verbatim}
1 kg \texttt{\cunum{1}}{kg}\\
2.3 kg \texttt{\cunum{2.3}}{kg}\\
2.3 kg \texttt{\cunum{2,3}}{kg}\\
2–3 kg \texttt{\cunum{2--3}}{kg}\\
2.5–3.5 kg \texttt{\cunum{2.5--3.5}}{kg}\\
2500–3500 g \texttt{\cunum[kg=g]{2.5--3,5}}{kg}\\
392 °F \texttt{\cunum[C=F]{180--200}}{C}\\
356–392 °F \texttt{\cunum[C=F]{180--200}}{C}\\
\frac{1}{2} m \texttt{\cunum{1/2}}{m}\\
1\frac{1}{2} m \texttt{\cunum{1_1/2}}{m}\\
1\frac{1}{2} m \texttt{\cunum[m=cm]{1_1/2}}{m}\\
? ℓ \texttt{\cunum{?}}{m}\\
50 dag \texttt{\cunum{50}}{dag}\\
5 dag \texttt{\cunum{5}}{dag}\\
1.12 m \texttt{\cunum{1.1234}}{m}\\
\end{verbatim}

\(^1\)You can only get limited information from the internet.
\(^2\)New keys can be added and defined, see section 4 and section 5 for further information.
\(^3\)You can customize this behavior, see section 8.
Decimal numbers are automatically rounded to 2 digits after the colon, temperatures (C, F, K and Re) are automatically rounded to integers.\(^4\)

`\text` and `\Text` print the number and the written name of the unit. Since v1.10 it works similar\(^5\) to `\numeral`: it allows the conversion between units and interprets the numerical part (again _ and / are used for (mixed) fractions and -- for ranges). Furthermore, `\text` and `\Text` allow the usages of numerals (see section 8.1 for more information).

<table>
<thead>
<tr>
<th>Number</th>
<th><code>\text</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 litre</td>
<td><code>\text{1}{l}\</code></td>
</tr>
<tr>
<td>1 litre</td>
<td><code>\Text{1}{l}\</code></td>
</tr>
<tr>
<td>1 to 2 litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>12 litres</td>
<td><code>\text{12}{l}\</code></td>
</tr>
<tr>
<td>13 litres</td>
<td><code>\Text{13}{l}\</code></td>
</tr>
</tbody>
</table>

You can customize the numeral functions used with `\numeral-function` and `\numeral-function`. Furthermore, since v1.10 `\text` and `\Text` also allow their units to be changed (this behavior can be altered using `\text-change-unit`):

<table>
<thead>
<tr>
<th>Number</th>
<th><code>\text</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 litre</td>
<td><code>\text{1}{l}\</code></td>
</tr>
<tr>
<td>1 litre</td>
<td><code>\Text{1}{l}\</code></td>
</tr>
<tr>
<td>1 to 2 litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>12 litres</td>
<td><code>\text{12}{l}\</code></td>
</tr>
<tr>
<td>13 litres</td>
<td><code>\Text{13}{l}\</code></td>
</tr>
</tbody>
</table>

You can use (e.g.) package option `use-fmtcount-numerals=true`

<table>
<thead>
<tr>
<th>Number</th>
<th><code>\text</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>one litre</td>
<td><code>\text{1}{l}\</code></td>
</tr>
<tr>
<td>One litre</td>
<td><code>\Text{1}{l}\</code></td>
</tr>
<tr>
<td>one to two litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>One to two litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>twelve litres</td>
<td><code>\text{12}{l}\</code></td>
</tr>
<tr>
<td>13 litres</td>
<td><code>\Text{13}{l}\</code></td>
</tr>
</tbody>
</table>

You can customize the numeral functions used with `\numeral-function` and `\numeral-function`. Furthermore, since v1.10 `\text` and `\Text` also allow their units to be changed (this behavior can be altered using `\text-change-unit`):

<table>
<thead>
<tr>
<th>Number</th>
<th><code>\text</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 litre</td>
<td><code>\text{1}{l}\</code></td>
</tr>
<tr>
<td>1 litre</td>
<td><code>\Text{1}{l}\</code></td>
</tr>
<tr>
<td>1 to 2 litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>12 litres</td>
<td><code>\text{12}{l}\</code></td>
</tr>
<tr>
<td>13 litres</td>
<td><code>\Text{13}{l}\</code></td>
</tr>
</tbody>
</table>

You can use (e.g.) package option `use-fmtcount-numerals=true`

<table>
<thead>
<tr>
<th>Number</th>
<th><code>\text</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>one litre</td>
<td><code>\text{1}{l}\</code></td>
</tr>
<tr>
<td>One litre</td>
<td><code>\Text{1}{l}\</code></td>
</tr>
<tr>
<td>one to two litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>One to two litres</td>
<td><code>\Text{1--2}{l}\</code></td>
</tr>
<tr>
<td>twelve litres</td>
<td><code>\text{12}{l}\</code></td>
</tr>
<tr>
<td>13 litres</td>
<td><code>\Text{13}{l}\</code></td>
</tr>
</tbody>
</table>

\`\cuam\` works like \`\numeral\`, but without a unit, so changing units doesn’t affect it. Like `\numeral` _ and / are used to imply a (mixed) fraction and -- is used for ranges.

<table>
<thead>
<tr>
<th>Number</th>
<th><code>\text</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td><code>\cuam{3}\</code></td>
</tr>
<tr>
<td>2–3</td>
<td><code>\cuam{2--3}\</code></td>
</tr>
<tr>
<td>2/3</td>
<td><code>\cuam{2/3}\</code></td>
</tr>
<tr>
<td>1 1/3</td>
<td><code>\cuam{1_1/3}\</code></td>
</tr>
</tbody>
</table>

Furthermore it allows the concept of “phrases” (replacing a positive integer by a word; such as “12” becoming “dozen”) which can be activated by the option `use-phrases` (as I don’t know any English phrases, I switched the language to German for the following examples)

\(^4\) You can – of course – change this behavior, see section 8.
\(^5\) One could also say “exactly like”.
\(^6\) At least I think
3 Label & refs: Changing the amount of the recipe

What if you don’t want to change units, but the amounts of the recipe because you cook not for 4 persons, but for 2 and don’t like to do the math? Simple, use the following commands:

- \culabel{⟨label⟩}{⟨number of persons⟩}
- \curef{⟨label⟩}

The first one is the important one: It defines a ⟨label⟩ for a recipe which is initially for ⟨number of persons⟩. Afterwards ⟨label⟩ can be used to tell the commands from section 2 that the given amounts are for ⟨number of persons⟩. Each ⟨label⟩ must be unique and an error is raised if a ⟨label⟩ is already defined.

If you would like to print the number of persons this recipe is for, use \curef, which is fully expandable.

The following example uses \culabel to specify that the recipe is initially intended for 2 persons:

```
\culabel{recipe}{2}
```

recipe for 2 persons:
10–20 dag flour,
\(\frac{1}{2}\) ℓ water,
10 gramme nuts,
2–3 eggs,
180°C (356°F) open fire

In combination with the option set-number-of-persons and recalculate-amount you can have this recipe changed to four persons:

```
\culabel{recipe}{2}
\% adding options:
\cusetup{set-number-of-persons=4,recalculate-amount=true}
```

recipe for 4 persons:
20–40 dag flour,
1 ℓ water,
20 gramme nuts,
4–6 eggs,
180°C (356°F) open fire
Note that fractions are automatically evaluated and that only values with a ⟨label⟩ are changed (\cunum{180}{C} for example stays the same which also makes sense as the heat should be the same).

3.1 Rounding temperatures

By default temperatures are rounded to integers (using round-precision=0). Since 1.30 it is possible to round amounts to a negative precision. If you want to round temperatures to the tens see the following example (set-option-for-(unit) is described in section 8.2.1).

\begin{verbatim}
182°C \cunum{182}{C}
356°F \cunum[C=F]{180}{C}
144°Ré \cunum[C=Re]{180}{C}
453K \cunum[C=K]{180}{C}
\cusetoptionfor{C,F,K,Re}{round-precision=-1}
180°C \cunum{182}{C}
360°F \cunum[C=F]{180}{C}
140°Ré \cunum[C=Re]{180}{C}
450K \cunum[C=K]{180}{C}
\end{verbatim}

4 Predefined units & some notes

In table 1 and you can find all predefined units which can be transformed into each other (sorted by group). Other predefined units (which cannot be used for transformation) are shown in table 2. Table 3 pretty much exists just for fun.

Table 1: This table shows all units which can be transformed into each other, sorted by group. The columns “default” show the abbreviations used if for given language no translation is defined. The translations used for \cutext and \Cutext are shown in appendix A. Note that “electron volt” exists just for fun.

<table>
<thead>
<tr>
<th>description</th>
<th>key</th>
<th>default</th>
<th>description</th>
<th>key</th>
<th>default</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilogramme</td>
<td>kg</td>
<td>kg</td>
<td>metre</td>
<td>m</td>
<td>m</td>
</tr>
<tr>
<td>decagramme</td>
<td>dag</td>
<td>dag</td>
<td>decimetre</td>
<td>dm</td>
<td>dm</td>
</tr>
<tr>
<td>gramme</td>
<td>g</td>
<td>g</td>
<td>centimetre</td>
<td>cm</td>
<td>cm</td>
</tr>
<tr>
<td>ounce</td>
<td>oz</td>
<td>oz</td>
<td>millilitre</td>
<td>mm</td>
<td>mm</td>
</tr>
<tr>
<td>pound (of butter)</td>
<td>lb</td>
<td>lb</td>
<td>inch</td>
<td>in</td>
<td>in</td>
</tr>
<tr>
<td>day</td>
<td>d</td>
<td>d</td>
<td>litre</td>
<td>l</td>
<td>l</td>
</tr>
<tr>
<td>hour</td>
<td>h</td>
<td>h</td>
<td>decilitre</td>
<td>dl</td>
<td>dl</td>
</tr>
<tr>
<td>minute</td>
<td>min</td>
<td>min</td>
<td>centilitre</td>
<td>cl</td>
<td>cl</td>
</tr>
<tr>
<td>second</td>
<td>s</td>
<td>s</td>
<td>millilitre</td>
<td>ml</td>
<td>ml</td>
</tr>
<tr>
<td>calorie</td>
<td>cal</td>
<td>cal</td>
<td>degree Celsius</td>
<td>°C</td>
<td>°C</td>
</tr>
<tr>
<td>kilocalorie</td>
<td>kcal</td>
<td>kcal</td>
<td>degree Fahrenheit</td>
<td>°F</td>
<td>°F</td>
</tr>
<tr>
<td>joule</td>
<td>J</td>
<td>J</td>
<td>degree Réaumur</td>
<td>°Re</td>
<td>°Re</td>
</tr>
<tr>
<td>kilojoule</td>
<td>kJ</td>
<td>kJ</td>
<td>kelvin</td>
<td>K</td>
<td>K</td>
</tr>
<tr>
<td>electron volt</td>
<td>eV</td>
<td>eV</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6
Table 2: A (not only) spoonful of (more or less) country and language dependent units. Please note that sometimes a translation is nearly impossible as a unit (e.g. “saltspoonful”) may not exist in another language (like german; at least I never heard of it). So please only use units known to you.

<table>
<thead>
<tr>
<th>description</th>
<th>key</th>
<th>symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>pinch</td>
<td>pn</td>
<td>pinch</td>
</tr>
<tr>
<td>tablespoon</td>
<td>EL</td>
<td>EL</td>
</tr>
<tr>
<td>teaspoon</td>
<td>TL</td>
<td>TL</td>
</tr>
<tr>
<td>dessertspoonful</td>
<td>dsp</td>
<td>dsp.</td>
</tr>
<tr>
<td>coffeespoonful</td>
<td>csp</td>
<td>csp.</td>
</tr>
<tr>
<td>saltspoonful</td>
<td>ssp</td>
<td>ssp.</td>
</tr>
<tr>
<td>Messerspitze (point of a knife)</td>
<td>Msp</td>
<td>Msp.</td>
</tr>
</tbody>
</table>

Table 3: List of (not really) nonsense units (exist just for fun, there will be no support for those units; unless – of course – you really want it).

<table>
<thead>
<tr>
<th>unit-key</th>
<th>symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>eVc-2</td>
<td>$eV/c^2$</td>
</tr>
<tr>
<td>hbareV-1</td>
<td>$h/eV$</td>
</tr>
<tr>
<td>chbareV-1</td>
<td>$\hbar/eV$</td>
</tr>
<tr>
<td>(chbareV-1)$^3$</td>
<td>$\hbar^3/eV^3$</td>
</tr>
</tbody>
</table>

5 Defining units

New units can be defined using

- \texttt{\textbackslash declarecookingunit\{}\langle symbol\rangle\{\langle unit-key\rangle\}}
- \texttt{\textbackslash newcookingunit\{}\langle symbol\rangle\{\langle new-unit-key\rangle\}}
- \texttt{\textbackslash providecookingunit\{}\langle symbol\rangle\{\langle new-unit-key\rangle\}}
These commands define the unit \textlangle unit-key \rangle. If the key is not the same as the printed symbol use \textlangle symbol \rangle. Note that \textlangle unit-key \rangle can neither contain / nor \. Please note due to the current implementation catcodes may cause trouble. For example using : inside \textlangle unit-key \rangle may cause the key to “not be defined” in the document. If that happens try removing or changing the \textlangle unit-key \rangle.

\textbackslash newcookingunit raises an error if the unit is already defined, \textbackslash declarecookingunit creates or (if given) overwrites \textlangle symbol \rangle and \textbackslash providecookingunit does nothing if the unit is already defined. All units have male gender \textbackslash m by default.

Some examples:

\begin{verbatim}
\declarecookingunit{kg}
\declarecookingunit{g}
\declarecookingunit[Msp.]{Msp}
\declarecookingunit[\ensuremath{{}^\circ}\kern-\scriptspace C]{\degree C}
\end{verbatim}

\textbf{Note:} The definition of the printed degree Celsius is copied and pasted from (a maybe older version of) \texttt{siunitx}.

\textbackslash declarecookingderivatives \textlangle (unit-list) \rangle \textlangle (unit-key) \rangle \textlangle (mathematical-relation) \rangle \textlangle (unit-symbol) \rangle

This function is experimental. Defines new units which are a combination of the units given in \textlangle unit-list \rangle and their linked-units. \textlangle unit-key \rangle, \textlangle mathematical-relation \rangle and \textlangle unit-symbol \rangle accept \#1 to \#n as arguments with \( n \) being the number of units given in \textlangle unit-list \rangle. \( n \) cannot be greater than 8 (and it will probably compile for quite a while). Also note that this command doesn’t work/isn’t tested for single keys. Also note that it is quite possible that an “overflow-error” will occur if there are too many units.

\textbf{Note:} Due to catcodes (and my inability to deal with them properly) there can be problems when using : (and probably other signs) inside \textlangle unit-key \rangle.

\textbf{Example:} Your homework is to change the unit of energy \( \text{kg m}^2 \text{s}^{-2} \) into \( \text{oz in}^2 \text{ min}^{-2} \). To check if you are correct you use \textbackslash declarecookingderivatives:

\begin{verbatim}
\declarecookingderivatives{kg,m,s}{#1*#2-#3}
{ (#1)*(#2)^2/(#3)^2 } {\sfrac{#1\,#2^2}{#3^2}}
\end{verbatim}

Using \texttt{\cunum[kg*m-s=oz*in-min]{1}{kg*m-s}} shows that \( 1 \text{ kg m}^2 \text{s}^{-2} \) is equal to 196829101.34 \text{ oz in}^2 \text{ min}^{-2}. (Note: This is really experimental and it can easily happen that an “overflow-error” occurs) (like for me which is why I hard-coded the results after testing).
6 Defining options to change units

Options (to change units) can be newly defined or added to already existing keys (units) using

- \definekeys
- \definesinglekey
- \addkeys
- \addsinglekeys
- \addtokeys

I apologize for the (name) inconsistency between \definekeys and \definesinglekey (although they are named similarly, they work differently).

\definekeys{⟨\textit{unit-key-1}⟩}{
\{⟨\textit{unit-key-2}⟩\} {\{1 \textit{unit-key-1} are ... \textit{unit-key-2}\}}
\{⟨\textit{unit-key-3}⟩\} {\{1 \textit{unit-key-1} are ... \textit{unit-key-3}\}}
\{⟨\textit{unit-key-4}⟩\} {\{1 \textit{unit-key-1} are ... \textit{unit-key-4}\}}
...}
\definesinglekey{⟨\textit{unit-key-1}⟩}{
\{⟨\textit{unit-key-2}⟩\} {\{1 \textit{unit-key-2 are ... \textit{unit-key-1}\}}
\{⟨\textit{unit-key-3}⟩\} {\{1 \textit{unit-key-3 are ... \textit{unit-key-1}\}}
...}

If you define new units (see section 5) and cannot add them to already existing keys you can use \definekeys or \definesinglekey respectively to define new keys.

\definekeys takes \{⟨\textit{unit-key-1}⟩\} as a “basis”, defines a key with the name \{\textit{unit-key-1}\} and adds the values \{\textit{unit-key-1}\}, \{\textit{unit-key-2}\}, \{\textit{unit-key-3}\}, etc. Furthermore this command also defines the keys \{\textit{unit-key-2}\}, \{\textit{unit-key-3}\}, etc. with the same values as \{\textit{unit-key-1}\}. Please note that \{\ldots\} has to be a number.

Sometimes it is not that easy and the conversion of one unit into another needs are more complicated formula (see for example temperatures). If that is the case use \definesinglekey. As the name says it defines only the key \{\textit{unit-key-1}\} with the values \{\textit{unit-key-1}\}, \{\textit{unit-key-2}\}, etc. The advantage of this command is that now \{\ldots\} can be a formula and the numerical input can be placed explicitly using #1.

Example: This example defines following keys with their respective value:

- the key \textit{kg} with the values \textit{kg}, \textit{dag}, \textit{g} and \textit{oz}
- the key \textit{dag} with the values \textit{kg}, \textit{dag}, \textit{g} and \textit{oz}
- the key \textit{g} with the values \textit{kg}, \textit{dag}, \textit{g} and \textit{oz}
- the key \textit{oz} with the values \textit{kg}, \textit{dag}, \textit{g} and \textit{oz}
- the key \textit{d} with the values \textit{d}, \textit{h}, \textit{min} and \textit{s}
1 kg = 1 kg  \quad 1 kg = 100 dag  \quad 1 kg = 1000 g  \\
1 kg = 35.27399 oz  \quad 1 kg = 2.2046226 lb

\texttt{\textbackslash cudefinekeys} \{kg\}
\begin{verbatim}
{\{dag\} 100 } \%\% 1 kg are 100 dag  
{\{g\} 1000 } \%\% 1 kg are 1000 g  
{\{oz\} 35.27399 } \%\% 1 kg are 35.27399 oz  
{\{lb\} 2.2046226 } \%\% 1 kg are 2.2046226 lb
\end{verbatim}

\texttt{\textbackslash cudefinekeys} \{d\}
\begin{verbatim}
{\{h\} 24 } \%\% 1 day are 24 hours  
{\{min\} 1440 } \%\% 1 day are 1440 minutes  
{\{s\} 86400 } \%\% 1 day are 86400 seconds
\end{verbatim}

To convert degree Fahrenheit to degree Celsius, kelvin and degree Réamur one needs the formulas\footnote{See Wikipedia.}

\begin{align*}
T_C &= (T_F - 32) \cdot \frac{5}{9} \\
T_K &= (T_F - 459.67) \cdot \frac{5}{9} \\
T_{Re} &= (T_F - 32) \cdot \frac{4}{9}
\end{align*}

with $T_F$ being the input temperature in degree Fahrenheit and $T_C$ being the same temperature in degree Celsius, etc. Using \texttt{\textbackslash cudefinesinglekey} the key F with values C, K and Re is defined:

\texttt{\textbackslash cudefinesinglekey} \{F\}
\begin{verbatim}
{\{C\} \{ \ ( \#1 - 32 \) \* 5/9 \} \%\% see formulas above  
{\{K\} \{ \ ( \#1 + 459.67 \) \* 5/9 \}} 
{\{Re\} \{ \ ( \#1 - 32 \) \* 4/9 \}}
\end{verbatim}

This defines the key F with the values F, C, K and Re.

\footnote{See Wikipedia.}
These commands add \langle unit-key-2 \rangle, \ldots to the already defined key \langle unit-key-1 \rangle. \texttt{\cuaddkeys} takes the already defined key \{\langle unit-key-1 \rangle\} as a “basis”, and adds \langle unit-key-2 \rangle, \langle unit-key-3 \rangle, \ldots to its values. Furthermore it adds those new values to other keys linked to \langle unit-key-1 \rangle and defines the new keys \langle unit-key-2 \rangle, \langle unit-key-3 \rangle, \ldots with the same values as \langle unit-key-1 \rangle.

If the conversion is more complicated use \texttt{\cuaddsinglekeys}. It adds \langle unit-key-2 \rangle, \ldots as values to \langle unit-key-1 \rangle. The numerical input can be placed using \#1 (see \texttt{\cudefinesinglekey}). This command neither defines new keys nor does it add values to other keys than \langle unit-key-1 \rangle.

Example: Suppose you are British (I am sorry, I can’t think of another reason to use those units) and you want to implement ‘stone’ (yes, I was surprised myself that such a unit exists, but it even appears in a Sherlock Holmes story). You exactly know that 1 st equals 14 lb, well ... now you have two choices. \texttt{\cuaddkeys} or \texttt{\cuaddtokeys} (use the one best fitting). This example uses the first, the next the latter one.

\begin{verbatim}
\newcookingunit{st} \% defining new unit 'stone'
\cuaddkeys{lb} \% adding st to lb (could also add to kg, dag and oz)
{ {st} { 1/14 } \% 1 lb are 1/14 st as 14 lb are 1 st }
\end{verbatim}

\begin{verbatim}
0.07 st \cunum[lb=st]{0.07}{lb}\% \cunum[lb=st]{1}{lb}\%
14 lb \cunum[st=lb]{14}{lb}\% \cunum[st=lb]{1}{lb}\%
6350.29 g \cunum[st=kg]{6350.29}{kg}\% \cunum[st=kg]{1}{kg}\%
0.16 st \cunum[kg=st]{0.16}{kg}\% \cunum[kg=st]{1}{kg}\%
101.6 kg \cunum[st=kg]{101.6}{kg}\% \cunum[st=kg]{16}{kg}\%
\end{verbatim}

Example: Now you want to add degree Rømer and convert Celsius to degree Rømer:

\begin{equation}
T_{Ro} = T_C \times \frac{21}{40} + 7.5
\end{equation}

\texttt{\cunum[lb=st]{0.07}{lb}} \% \cunum[lb=st]{1}{lb} \%
\texttt{\cunum[st=lb]{14}{lb}} \% \cunum[st=lb]{1}{lb} \%
\texttt{\cunum[st=kg]{6350.29}{kg}} \% \cunum[st=kg]{1}{kg} \%
\texttt{\cunum[kg=st]{0.16}{kg}} \% \cunum[kg=st]{1}{kg} \%
\texttt{\cunum[st=kg]{101.6}{kg}} \% \cunum[st=kg]{16}{kg} \%

\texttt{\cunum[lb=st]{14}{lb}} \% \cunum[lb=st]{1}{lb} \%
\texttt{\cunum[st=lb]{6350.29}{kg}} \% \cunum[st=lb]{1}{kg} \%
\texttt{\cunum[st=kg]{0.16}{kg}} \% \cunum[st=kg]{1}{kg} \%
\texttt{\cunum[st=kg]{101.6}{kg}} \% \cunum[st=kg]{16}{kg} \%

\{\text{Ro}\} \{ \#1 \times 21/40 + 7.5 \}

\cusetup \% round to integer automatically
\{
  \setoptionfor\text{Ro} = \{ \text{round-precision} = 0 \}
\}

\begin{align*}
10^\circ \text{C} & \quad \\text{\cunum(10)\{C}\}} \\
13^\circ \text{Ro} & \quad \\text{\cunum[C=Ro\{10\}\{C}\}}
\end{align*}

\text{Works similar to} \text{\cuaddkeys} \text{regarding the definition of keys.}

\textbf{Example:} \text{Continuing the example from before, this time with} \text{\cuaddtokeys}:
\begin{align*}
\text{\newcookingunit\{st\} \% defining (again) new unit 'stone'} \\
\text{\cuaddtokeys\{lb\} \{st\} \{ 14 \} \% 1 \text{ st are 14 lb}}
\end{align*}

\begin{align*}
0.07 \text{st} & \quad \\text{\cunum[lb=st\{1\}\{lb\}} \\
14 \text{lb} & \quad \\text{\cunum[st=lb\{1\}\{st\}} \\
6350.29 \text{g} & \quad \\text{\cunum[st=g\{1\}\{st\}} \\
6.35 \text{kg} & \quad \\text{\cunum[st=kg\{1\}\{st\}} \\
0.16 \text{st} & \quad \\text{\cunum[kg=st\{1\}\{kg\}} \\
101.6 \text{kg} & \quad \\text{\cunum[kg=st\{16\}\{st\}}
\end{align*}

\section{Language support}

Unit names and symbols depend on the language. To change the name and symbol for given language you can use \texttt{\cudedefinename}; to only change symbols use \texttt{\cudedefinesymbol}.

\begin{itemize}
  \item \texttt{decimal-mark}
  \item \texttt{cutext-range-sign}
  \item \texttt{one(m)}
  \item \texttt{one(f)}
  \item \texttt{one(n)}
\end{itemize}

Those are special keys (as they cannot be used as units). Not only are printed units language depending, but as is the decimal mark (,. or ,) and the text which substitutes the range-sign. To set the decimal mark use \texttt{decimal-mark} (see examples below), to set the range-sign for \texttt{cutext} and \texttt{Cutext} use \texttt{cutext-range-sign}.

Note that \texttt{cutext-range-sign} is “overwritten” by the option \texttt{cutext-range-sign}. If the \texttt{option} is set, then the language symbol will be ignored.

Furthermore if you are using numerals you may also use the keys \texttt{one(m)}, \texttt{one(f)} and \texttt{one(n)}. Integers below a certain value (see option \texttt{use-numerals-below}) are written-out. The only problem is the written-out “1” mostly depends on the gender of the word following (e.g. “ein Baum” (m), “eine Pflanze” (f) and “ein Auto” (n)). To set the written-out “1” to be correct with the gender of the used unit, use these keys (see also examples below).
\cudefinename{\textit{Language}}
{
  \{\textit{unit-key-1}\} \{\textit{symbol-1}\} \{\textit{singular-1}\} \{\textit{plural-1}\} <\textit{gender}>
  \{\textit{unit-key-2}\} \{\textit{symbol-2}\} \{\textit{singular-2}\} \{\textit{plural-2}\} <\textit{gender}>
  ...
}

This command defines the names (and optionally the symbol) of the units printed in \cutext and \Cutext (and \cunum regarding the symbol) for the specific \textit{Language}. For details regarding \textit{language} see the translations documentation.

If the plural form of the name differs from the singular form use \{\textit{plural}\} to specify the plural form, else it will be equal to its singular form. The singular form is only used if the number in \cutext and \Cutext is equal to 1.

\textit{gender} can be m (maskulin), f (femminin) or n (neutrum). If not given m is used as default.

\cudefinename{English}
{
  \{kg\} \{kilogramme\}
  \{oz\} \{ounce\}
  \{h\} \{hour\} \{hours\}
  \{C\} \{degree\space Celsius\} \{degrees\space Celsius\}
  \{decimal-marker\} \{.\}
  \{cutext-range-sign\} \{-to-\}
  \{one(m)\} \{one\}
  \{one(f)\} \{one\}
  \{one(n)\} \{one\}
}

\cudefinename{German}
{
  \{kg\} \{Kilogramm\} \langle n\rangle
  \{oz\} \{Unze\} \langle f\rangle
  \{d\} \{Tag\} \{Tage\}
  \{h\} \{Stunde\} \{Stunden\} \langle f\rangle
  \{C\} \{Grad\space Celsius\}
  \{decimal-marker\} \{,\}
  \{cutext-range-sign\} \{-bis-\}
  \{one(m)\} \{ein\}
  \{one(f)\} \{eine\}
  \{one(n)\} \{ein\}
}

\cudefinesymbol{\textit{Language}}
{
  \{\textit{unit-key-1}\} \{\textit{symbol-1}\}
  \{\textit{unit-key-2}\} \{\textit{symbol-2}\}
  ...
}

This command defines the symbols of the units printed in \cunum for the specific \textit{Language}. It works similar as \cudefinename, but only the symbols (and no names) can be set. For details regarding \textit{language} see the translations documentation.
\cudefinesymbol {English}
{
{decimal-mark} {.}
{cutext-range-sign} {\text{-to-}}
{one(m)} {one}
{one(f)} {one}
{one(n)} {one}
}
\cudefinesymbol {German}
{
{decimal-mark} {,}
{cutext-range-sign} {\text{-bis-}}
{one(m)} {ein}
{one(f)} {eine}
{one(n)} {ein}
}
\cudefinesymbol {French}
{
{l} {L}
{dl} {dL}
{cl} {cL}
{ml} {mL}
{decimal-mark} {.}
{one(m)} {un}
{one(f)} {une}
{one(n)} {un}
}

Example: Imagine that instead of the abbreviation “dag” for “decagramme” you want to use “ducks” (because ... I don’t know). You can easily do this via

\cudefinesymbol {English}
{
{dag} {ducks}
}

As you can see it may be a bit suboptimal as there is no plural version allowed. You do it anyway and end up with:

\cunum{12}{dag} weed\
3 ducks nuts \cunum{3}[0]{dag} nuts\
10ducks duckmeat \cunum{10}{dag} duckmeat

7.1 Phrases

Each language has synonyms for certain (integer) numbers. This package supports those phrases and they can be implemented with the following command and used by \cuam:
This command pairs for a given \{Language\} (see package translations) the number \{integer-1\} with \{phrase-1\} (\& plural and gender). The package then checks if the amount given in \cuam is either this number or a multiple of it.

If the behavior of checking for a multiple is not wanted, you can use the optional star * for a given \{integer\}:

\{gender\} can be m, f or n. It is m by default.

Afterwards the numbers are ordered from highest to lowest so that the phrase with the highest number is used (if used at all).

Furthermore, it chooses star (*) phrases over non-star phrases.

Note: Numbers with the optional star * are stored as negative numbers.

Example: The following example creates some phrases for the language “German”:

\cudefinephrase {German} {
    \{12\} \{Dutzend\} \<n\> \%% implemented by default \\
    \{60\} \{Schock\} \<n\> \\
    \{6\}* \{halbes Dutzend\} \<n\>
}

Let’s just use them (german language activated!):

\cusetup{use-phrases=true}

\cuam{12}\\
\cuam{24}\\
\cuam{60}\\
\cuam{120}\\
\cuam{6}\\
\cuam{18}

As you can see, “Schock” (60) is preferred over “Dutzend” (12) as it linked to the higher number. Furthermore, for 6 the phrase “halbes Dutzend” (half a dozen) is used, but because it is a star version it is not used for 18.

8 Options

Options in cooking-units can mostly be set globally using \cusetup or locally using the optional argument of the respective command (but not as a package option). The only exception is the option given in section 8.1 which needs to be used as a package option.
cooking-units allows you to attach options to units. Those options are activated if (and only if) the specific unit is used or if another unit is converted into it. Those options allow you to e.g. round temperatures to integers automatically. Furthermore, those added options are overwritten by local options.

\cusetoptionfor sets \langle options \rangle to each unit in \langle unit-list \rangle overwriting the old ones. \cuaddoptionfor adds \langle options \rangle to each unit in \langle unit-list \rangle. \cuclearoptionfor clears all options given to each unit in \langle unit-list \rangle.

Example: Temperatures C, F, K and Re are by default rounded to integers.

```
75 °C\cunum{75.23}{C}\n
75 °F\cunum{75.23}{F}\n
75 K\cunum{75.23}{K}\n
75 °Ré\cunum{75.23}{Re}\n
\cusetoptionfor{C,F,K,Re}{round-precision=-1}\n
80 °C\cunum{75.23}{C}\n
80 °F\cunum{75.23}{F}\n
80 K\cunum{75.23}{K}\n
80 °Ré\cunum{75.23}{Re}\n
\cuclearoptionfor{C,F,K,Re}\n
75.23 °C\cunum{75.23}{C}\n
75.23 °F\cunum{75.23}{F}\n
75.23 K\cunum{75.23}{K}\n
75.23 °Ré\cunum{75.23}{Re}\n```

### 8.1 Load time options

\usepackage{use-fmtcount-numerals=(true/false)}{cooking-units}

If set to true loads package fmtcount and uses \numberstringnum for \cutext and \Numberstringnum for \Cutext to write-out numbers below use-numerals-below (13 by default), integers above are printed as numbers. You can decide to not print any numerals by setting print-numerals to false.

Note: You don’t need to use this function to use numerals. Using print-numerals and setting numeral-function and Numeral-function also works.

```
one kilogramme \cutext{1}{kg}\n
One kilogramme \Cutext{1}{kg}\n
two kilogramme \cutext{2}{kg}\n
Two kilogramme \Cutext{2}{kg}\n
twelve kilogramme \cutext{12}{kg}\n
Twelve kilogramme \Cutext{12}{kg}\n
13 kilogramme \cutext{13}{kg}\n
13 kilogramme \Cutext{13}{kg}\n
14 kilogramme \Cutext{14}{kg}\n```

Note: use-fmtcount-numerals is a package option as it needs to load fmtcount which is not loaded by default.
Note: Please note the keys one(m), one(f) and one(n) to change the printed “one” (as “one” is in many languages dependent on the gender of the following word. E.g in German: Masculine: ein Baum, Feminin: eine Pflanze, Neutrum: ein Auto).

Note: You can always change the functions used to print numerals with numeral-function and Numeral-function.

8.2 Normal options
Options in this subsection can only be set as local options or using \cusetup, but not as load time options.

8.2.1 Unit Specific options

\begin{itemize}
\item \texttt{<unit>} \hspace{1cm} \langle unit-key-1 \rangle = \langle unit-key-2 \rangle
\end{itemize}
Change \langle unit-key-1 \rangle to \langle unit-key-2 \rangle (see section 6 to define new options).

\begin{itemize}
\item \texttt{<group>} \hspace{1cm} \langle group \rangle = \langle unit-key \rangle
\end{itemize}
Changes each unit contained in \langle group \rangle to \langle unit-key \rangle (\langle unit-key \rangle must be part of \langle group \rangle).

<table>
<thead>
<tr>
<th>\langle group \rangle</th>
<th>default \langle unit-key \rangle s</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight</td>
<td>kg, dag, g, oz, lb, stick</td>
</tr>
<tr>
<td>length</td>
<td>m, dm, cm, mm, in</td>
</tr>
<tr>
<td>volume</td>
<td>l, dl, cl, ml</td>
</tr>
<tr>
<td>temperature</td>
<td>C, F, K, Re</td>
</tr>
<tr>
<td>energy</td>
<td>cal, kcal, J, kJ, eV</td>
</tr>
<tr>
<td>time</td>
<td>d, h, min, s</td>
</tr>
</tbody>
</table>

\texttt{\cusetup{weight=g}}
\texttt{\cunum{1}{kg}\\}
\texttt{\cunum{1}{dag}\\}
\texttt{\cunum{1}{g}\\}
\texttt{\cunum{1}{oz}\\}
\texttt{\cunum{1}{lb}\\}
\texttt{\cunum{1}{stick}\\}

You can define new groups using \texttt{\cudeclareunitgroup}: 

\texttt{\cudeclareunitgroup {\langle group-name \rangle} {\langle unit-list \rangle}}

Defines the group \langle group-name \rangle containing the list \langle unit-list \rangle. This allows the usage of \texttt{\meta{group-name}=\meta{unit-key}} to change all units in the group \langle group-name \rangle to \langle unit-key \rangle (which has to be part of \langle unit-list \rangle).
Example: Define the group “weight”:
\cudeclareunitgroup{weight} { kg, dag, g, oz, lb, stick }
Now \cusetup{weight=dag} can be used to change all units contained in weight to dag.

\cuaddtounitgroup\cuaddtounitgroup{(group)}{(unit-list)}
Adds \textlt{unit-list} to an already existing \textlt{group} (both need to exist).

Example: Adding st to the group weight
\cuaddtounitgroup{weight}{st}
\cusetup{weight=g}
\cunum{1}{kg}\n\cunum{1}{dag}\n\cunum{1}{g}\n\cunum{1}{oz}\n\cunum{1}{lb}\n\cunum{1}{stick}\n\cunum{1}{st}

\add-unit-to-group =
\{  
\textlt{group1} = \{\textlt{unit-key-list}\},  
\textlt{group2} = \{\textlt{unit-key-list}\},  
...  
\}

Add each \textlt{unit-key} in \textlt{unit-keys-list} to \textlt{group}. The key-val equivalent of \cuaddtounitgroup.

Example: The same as above: This example adds the unit st to the group weight and Ro to temperature.
\cusetup{  
\textlt{add-unit-to-group} = \{ weight = \{st\} , temperature = \{Ro\} \}  
}

1000 g \n\cunum{1}{kg}\n10 g \n\cunum{1}{dag}\n1 g \n\cunum{1}{g}\n28.35 g \n\cunum{1}{oz}\n453.59 g \n\cunum{1}{lb}\n113.4 g \n\cunum{1}{stick}\n6350.29 g \n\cunum{1}{st}

\set-option-for-\textlt{unit-key}  
\textlt{set-option-for-\textlt{unit-key}} = \{\textlt{key1 = value1}, ... \}\n
\add-option-for-\textlt{unit-key}  
\textlt{add-option-for-\textlt{unit-key}} = \{\textlt{key1 = value1}, ... \}\n
Sets and adds \textlt{key1=value1,...} to a specific \textlt{unit-key}, erase-all-options (see below) is used to erase all options for all \textlt{unit-key}s.

The less flexible key-value version of \cusetoptionfor and \cuaddoptionfor.
Example: The following rounds the values to integers for \( F, C, K \) and \( \text{Re} \):

\[
cusetup
\{ 
    \text{set-option-for-}F = \{ \text{round-precision} = 0 \} ,
    \text{set-option-for-}C = \{ \text{round-precision} = 0 \} ,
    \text{set-option-for-}K = \{ \text{round-precision} = 0 \} ,
    \text{set-option-for-}\text{Re} = \{ \text{round-precision} = 0 \} ,
\}
\]

although note that it would be easier to simply write

\[
cusetoptionfor \{F,C,K,\text{Re}\} \{ \text{round-precision} = 0 \}
\]

\begin{verbatim}
set-option-for set-option-for = 
    \{ \langle unit-key1 \rangle = \{ \langle keys=vals \rangle \}, \langle unit-key2 \rangle = \{ \langle keys=vals \rangle \}, ...
    \}
add-option-for = 
    \{ \langle unit-key1 \rangle = \{ \langle keys=vals \rangle \}, \langle unit-key2 \rangle = \{ \langle keys=vals \rangle \}, ...
    \}
\end{verbatim}

Sets/adds each \( \langle keys=vals \rangle \) to the specific \( \langle unit-key \rangle \). Works pretty much the same way their set-option-for-\( \langle unit-key \rangle \) and add-option-for-\( \langle unit-key \rangle \) counterparts.

The less flexible versions of the commands \cusetoptionfor and \cuaddoptionfor.

Example: The following example does the same as the example above:

\[
cusetup
\{ 
    \text{set-option-for} = 
    \{ 
        F = \{ \text{round-precision} = 0 \} ,
        C = \{ \text{round-precision} = 0 \} ,
        K = \{ \text{round-precision} = 0 \} ,
        \text{Re} = \{ \text{round-precision} = 0 \} ,
    \}
\}
\]

\begin{verbatim}
erase-all-options erase-all-options = \{ \langle unit-key1, unit-key2, ... \rangle \}
erase-all-options-for erase-all-options-for = \{ \langle unit-key, unit-key2, ... \rangle \}
\end{verbatim}

Erase options added to units. erase-all-options erases all options for all \( \langle unit-key \rangle \)s, erase-all-options-for is used to remove added options from the specified \( \langle unit-key \rangle \)s (key-value version of \culearoptionfor).
Example: The following code erases all attached options from C, F, K and Re:

\cusetup{ erase-all-options-for = {C, F, K, Re} }

It’s the same as

\cuclearoptionfor {C, F, K, Re}

8.2.2 Command behavior

<table>
<thead>
<tr>
<th>cutext-to-cunum</th>
<th>cutext-to-cunum = (true/false)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Want to get rid of all \cutext and \Cutext? Set this option to true and all \cutext and \Cutext are changed into \cunum.</td>
<td></td>
</tr>
</tbody>
</table>

| 1 kilogramme | \cutext{1}{kg}\ |
| 2 kilogramme | \Cutext{2}{kg}\ |
| ½ kilogramme | \cutext{1/2}{kg}\ |
| ? kilogramme | \cutext{?}{kg}\ |
| 1000 to 2000 gramme | \cutext[kg=g]{1--2}{kg}\ |
| \cusetup{cutext-to-cunum=true} |
| 1 kg | \cutext{1}{kg}\ |
| 2 kg | \Cutext{2}{kg}\ |
| ½ kg | \cutext{1/2}{kg}\ |
| ? kg | \cutext{?}{kg}\ |
| 1000–2000 g | \cutext[kg=g]{1--2}{kg}\ |

<table>
<thead>
<tr>
<th>cutext-change-unit</th>
<th>cutext-change-unit = (true/false)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set this option to false if you do not want the units of \cutext and \Cutext to be changed. Set to true by default.</td>
<td></td>
</tr>
</tbody>
</table>

| 1000 gramme | \cutext[kg=g]{1}{kg}\ |
| ½ kilogramme | \cutext[kg=g]{1/2}{kg}\ |
| 1000 to 2000 gramme | \cutext[kg=g]{1--2}{kg}\ |
| \cusetup{cutext-change-unit=false} |
| 1 kilogramme | \cutext[kg=g]{1}{kg}\ |
| ½ kilogramme | \cutext[kg=g]{1/2}{kg}\ |
| 1 to 2 kilogramme | \cutext[kg=g]{1--2}{kg}\ |

<table>
<thead>
<tr>
<th>cuam-version</th>
<th>cuam-version = (old/new)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cutext-version</td>
<td>cutext-version = (old/new)</td>
</tr>
<tr>
<td>Since v1.10 this package also parses and checks the input of \cutext and \Cutext and \cuam. If you want to restore the old behavior, set this option to old, but note that then you can neither change the amounts for a given number of persons nor change the unit of \cutext and \Cutext. Both of them are set to new by default.</td>
<td></td>
</tr>
</tbody>
</table>
8.2.3 Hooks

<table>
<thead>
<tr>
<th>commands-add-hook</th>
<th>cunum-add-hook</th>
<th>cutext-add-hook</th>
<th>Cutext-add-hook</th>
<th>cuam-add-hook</th>
</tr>
</thead>
<tbody>
<tr>
<td>commands-add-hook = {\text{\texttt{code}}}</td>
<td>cunum-add-hook = {\text{\texttt{code}}}</td>
<td>cutext-add-hook = {\text{\texttt{code}}}</td>
<td>Cutext-add-hook = {\text{\texttt{code}}}</td>
<td>cuam-add-hook = {\text{\texttt{code}}}</td>
</tr>
</tbody>
</table>

Adds \{\text{\texttt{code}}\} to the respective command (or in case of the first key: to all commands). The hook is executed after setting the keys, but before parsing and processing the input. Please be careful with spaces, they will be printed.

Example: You would like to count how often all commands of this package are used. Simply add:

\begin{verbatim}
\newcounter{CookingUnitsCounter} % or however you like it
\cusetup{commands-add-hook={\setcounter{CookingUnitsCounter}}}
\end{verbatim}

\texttt{\% beware of spaces inside the add-hook keys.}

to your preamble. The following table lists how often each command is used in this documentation (with help of \texttt{totalcount}):

<table>
<thead>
<tr>
<th>command</th>
<th>times</th>
</tr>
</thead>
<tbody>
<tr>
<td>cunum</td>
<td>217</td>
</tr>
<tr>
<td>cutext</td>
<td>66</td>
</tr>
<tr>
<td>Cutext</td>
<td>27</td>
</tr>
<tr>
<td>cuam</td>
<td>59</td>
</tr>
<tr>
<td>total</td>
<td>369</td>
</tr>
</tbody>
</table>

8.2.4 Input and Outputs

<table>
<thead>
<tr>
<th>expand-both</th>
<th>expand-both = {\text{n/o/f/x}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>expand-amount</td>
<td>expand-amount = {\text{n/o/f/x}}</td>
</tr>
<tr>
<td>expand-unit</td>
<td>expand-unit = {\text{n/o/f/x}}</td>
</tr>
</tbody>
</table>

By default the commands \texttt{cunum}, \texttt{cutext} and \texttt{Cutext} and \texttt{cunum} do not expand their input. You can change the expansion behavior of \{\text{amount}\} and/or \{\text{unit-key}\} using the options specified above. The meaning of the available values are the same as specified in the \LaTeX\ document “interface3”.

It is set to \texttt{n} (no expansion) by default.

<table>
<thead>
<tr>
<th>set-special-sign</th>
<th>set-special-sign = {(\text{character(s)})}</th>
</tr>
</thead>
<tbody>
<tr>
<td>add-special-sign</td>
<td>add-special-sign = {(\text{character(s)})}</td>
</tr>
</tbody>
</table>

Allows \{(\text{character(s)})\} to be used in the first mandatory argument of \texttt{cunum}, \texttt{cuam}, \texttt{cutext} and \texttt{Cutext} without raising an error (you can customize this behavior, see \texttt{set-unknown-message}). By default it is set to ?. Please note that the sign < is not allowed as a special sign.
\texttt{?} kg \hspace{1cm} \texttt{\textbackslash cunum}\{?\}\{kg\}\textbackslash \textbackslash
10?–20? kg \hspace{1cm} \texttt{\textbackslash cunum}\{g=kg\}\{10?--20?\}\{kg\}\textbackslash \textbackslash
\texttt{\textbackslash cusetup}\{add-special-sign={xX}\}\textbackslash 
\texttt{x} kg \hspace{1cm} \texttt{\textbackslash cunum}\{x\}\{kg\}\textbackslash \textbackslash
\texttt{X}--? kg \hspace{1cm} \texttt{\textbackslash cunum}\{X--?\}\{kg\}\textbackslash \textbackslash
\texttt{\textbackslash cusetup}\{set-special-sign={}\}\textbackslash 
1 kg \hspace{1cm} \texttt{\textbackslash cunum}\{1\}\{kg\}\textbackslash \textbackslash
1--2 kg \hspace{1cm} \texttt{\textbackslash cunum}\{1--2\}\{kg\}\textbackslash 

\texttt{set-unknown-message} \hspace{1cm} \texttt{set-unknown-message} = (error/warning/none)

Using a special sign (? by default) causes a warning to be raised. Set this option to \texttt{error} if you want an error (as an extra emphasis), \texttt{warning} if you want a warning (default) and \texttt{none} if you don’t want to know anything about it.

\texttt{set-cutext-translation-message} \hspace{1cm} \texttt{set-cutext-translation-message} = (error/warning/none)

If a translation for \texttt{\cutext} and \texttt{\Cutext} is not available the commands are replaced by \texttt{\cunum}. Currently – if this is happening – a warning is shown, you may change the behavior of the message (error, warning or not showing at all) using this option.

\texttt{print-numerals} \hspace{1cm} \texttt{print-numerals} = (true/false)

Prints numerals for integers smaller then \texttt{use-numerals-below} if set to \texttt{true}. If set to \texttt{false} no numerals are printed.

If you use the package option \texttt{use-fmtcount-numerals} this option is automatically set to \texttt{true}.

If you want to use another package, just set this option to \texttt{true} and use \texttt{numeral-function} and \texttt{Numeral-function}.

\textbf{Example:} (Using the package option \texttt{use-fmtcount-numerals}:

\texttt{one kilogramme} \hspace{1cm} \texttt{\cutext\{1\}\{kg\}\textbackslash \textbackslash
two kilogramme} \hspace{1cm} \texttt{\cutext\{2\}\{kg\}\textbackslash \textbackslash
twelve kilogramme} \hspace{1cm} \texttt{\cutext\{12\}\{kg\}\textbackslash \textbackslash
13 kilogramme} \hspace{1cm} \texttt{\cutext\{13\}\{kg\}\textbackslash \textbackslash
\texttt{\textbackslash cusetup}\{print-numerals=false\}\textbackslash 
1 kilogramme} \hspace{1cm} \texttt{\cutext\{1\}\{kg\}\textbackslash \textbackslash
2 kilogramme} \hspace{1cm} \texttt{\cutext\{2\}\{kg\}\textbackslash \textbackslash
12 kilogramme} \hspace{1cm} \texttt{\cutext\{12\}\{kg\}\textbackslash \textbackslash
13 kilogramme} \hspace{1cm} \texttt{\cutext\{13\}\{kg\}\textbackslash 

\texttt{use-numerals-below} \hspace{1cm} \texttt{use-numerals-below} = (integer)

If \texttt{print-numerals} is \texttt{true}, prints the numerals in \texttt{\cutext} and \texttt{\Cutext} for integers smaller than \texttt{(integer)}. \texttt{(integer)} is by default 13. You can deactivate the printing of numerals by \texttt{print-numerals=false}.

22
one kilogramme \cutext{1}\{kg\}\\
two kilogramme \cutext{2}\{kg\}\\
twelve kilogramme \cutext{12}\{kg\}\\
13 kilogramme \cutext{13}\{kg\}\\
  \cusetup{use-numerals-below=10}\n
one kilogramme \cutext{1}\{kg\}\\
two kilogramme \cutext{2}\{kg\}\\
12 kilogramme \cutext{12}\{kg\}\\
13 kilogramme \cutext{13}\{kg\}\\
  \cusetup{use-numerals-below=0}\n
1 kilogramme \cutext{1}\{kg\}\\
2 kilogramme \cutext{2}\{kg\}\\
12 kilogramme \cutext{12}\{kg\}\\
13 kilogramme \cutext{13}\{kg\}\\
  \cusetup{use-numerals-below=12001}\n
one thousand gramme \cutext[kg=g]{1}\{kg\}\\
two thousand gramme \cutext[kg=g]{2}\{kg\}\\
twelve thousand gramme \cutext[kg=g]{12}\{kg\}\\
13000 gramme \cutext[kg=g]{13}\{kg\}\\

\begin{tabular}{ll}
\textbf{numeral-function} & \textbf{numeral-function = (function)} \\
\textbf{Numeral-function} & \textbf{Numeral-function = (function)} \\
\end{tabular}

Sets the functions used for printing numerals. \texttt{numeral-function} is used for lowercase, \texttt{Numeral-function} for capitalized cases.

**Example:** Using the commands from \texttt{fmtcount} you can set the numeral function equal to

\begin{verbatim}
\cusetup{
  numeral-function = \numberstringnum ,
  Numeral-function = \Numberstringnum
}
\end{verbatim}

(this happens if you use the package option \texttt{use-fmtcount-numerals})

\begin{tabular}{ll}
\textbf{parse-number} & \textbf{parse-number = (true/false)} \\
\end{tabular}

If set to \texttt{false} prints the number of \texttt{\cunum, \cutext, \Cutext} and \texttt{\cuam} as they are (after some ... well ... parsing due to “_”). Is set to \texttt{true} by default.
range-sign = ⟨(string)⟩
cunum-range-sign = ⟨(string)⟩
cutext-range-sign = ⟨(string)⟩

\cunum-range-sign sets the printed range sign used in \cunum (and \cuam to \(⟨(string)⟩\)),
cutext-range-sign sets the printed range sign used in \cutext and \Cutext to \(⟨(string)⟩\).

Using range-sign sets the range signs for both \cunum (and \cuam) and \cutext/\Cutext to \(⟨(string)⟩\).

The default for \(⟨(string)⟩\) is –– (for both).

Since version 1.45 there also exits the language symbol cutext-range-sign (see section 7). If the option cutext-range-sign is set the language symbol will be ignored.
use-phrases

use-phrases = \langle true/false \rangle

Setting this option to true replaces certain integers (see section 7.1 for more information) with their phrase counterpart. This option is set to false by default.

Example: For the German language:

\begin{itemize}
  \item 12 \texttt{cuam{12}}
  \item 12–24 \texttt{cuam{12--24}}
  \item 36 \texttt{cuam{36}}
  \item 1 Dutzend \texttt{cuam{12}}
  \item 1–2 Dutzend \texttt{cuam{12--24}}
  \item 3 Dutzend \texttt{cuam{36}}
  \item ein Dutzend \texttt{cuam{12}}
  \item ein–zwei Dutzend \texttt{cuam{12--24}}
  \item drei Dutzend \texttt{cuam{36}}
\end{itemize}

8.2.5 Rounding options

round-precision

round-precision = \langle integer \rangle

Rounds the amount automatically to \langle integer \rangle digits after the colon. Note that units like C, F, K and Re are still rounded to integers due to \texttt{set-option-for-\langle unit-key \rangle}.

\begin{itemize}
  \item 1.23457 kg \texttt{cunum{1.23456789}{kg}}
  \item 0.01259 kg \texttt{cunum[g=kg]{12.587}{g}}
  \item 194 kg \texttt{cunum{194}{kg}}
  \item 392–410 °F \texttt{cunum[C=F]{200--210}{C}}
  \item -273 °C \texttt{cunum[K=C]{0.0012}{K}}
  \item 1.2 kg \texttt{cunum{1.23456789}{kg}}
  \item 12.6 kg \texttt{cunum{12.58}{kg}}
  \item 0.2 kg \texttt{cunum[g=kg]{194}{g}}
  \item 392–410 °F \texttt{cunum[C=F]{200--210}{C}}
  \item -273 °C \texttt{cunum[K=C]{0.0012}{K}}
\end{itemize}

Note: Also negative numbers are allowed.

\begin{itemize}
  \item -270 °C \texttt{cunum{-271,2}{C}}
  \item -270 °C \texttt{cunum[K=C]{0.0012}{K}}
  \item 180 °C \texttt{cunum{185}{C}}
  \item 360–390 °F \texttt{cunum[C=F]{180--200}{C}}
\end{itemize}

round-to-int

round-to-int = \langle true/false \rangle

This option is deprecated. Rounds the amount to an integer if set true. Use \texttt{round-precision=0} instead.
round-half \[\text{round-half} = \langle\text{default/commercial}\rangle\]

This option is only important for half-way numbers (e.g. 0.005). By setting it to \text{default} the value will be rounded to the nearest even number. Setting it to \text{commercial} rounds the value away from zero.

It is set to \text{default} by \ldots \text{default}.

\textbf{Note:} \text{default} actually refers to the fact that it is the default rounding algorithm used by \texttt{\textbackslash fp\_eval:n \{ round( ) \}} without a third argument.

\begin{verbatim}
\cusetup{round-half=default}
\cunum{0.005}{kg}\n\cunum{-0.005}{kg}\n\cunum{1.245}{kg}\n\cusetup{round-half=commercial}
\cunum{0.005}{kg}\n\cunum{-0.005}{kg}\n\cunum{1.245}{kg}
\end{verbatim}

8.2.6 Fractions

\textbf{eval-fraction} \[\text{eval-fraction} = \langle\text{true/false}\rangle\]

This option takes \text{true} or \text{false} as values. If set to \text{true} all fractions are evaluated. Please note that divisions through zero are not allowed.

\begin{verbatim}
\cusetup{eval-fraction=true}
\cunum{1/3}{kg}\n\cunum{1/2}{kg}\n\cunum{1/2}{kg}\n\cunum{-1/2}{kg}\n\cunum{1/2}{kg}\n\cunum{-1/2}{kg}\n\end{verbatim}

\textbf{convert-fraction} \[\text{convert-fraction} = \langle\text{true/false}\rangle\]

By default units of fractions are not converted into another unit. Setting this option to \text{true} allows fractions to be evaluated when a change of units is requested (and \textit{only} if a change of unit is requested).

\begin{verbatim}
\cusetup{convert-fraction=true}
\cunum{1/3}{kg}\n\cunum{1/3}{kg}\n\cunum{1/2}{kg}\n\cunum{1/2}{kg}\n\cunum{1/2}{kg}\n\cunum{1/2}{kg}\n\end{verbatim}

26
fraction-command \command

Sets the command used for printing fractions equal to \command. \command has to take two arguments. By default it is equal to \sfrac from xfrac.

Please note that the amount is *not* printed inside a math environment by default.

\newcommand\myfrac[2]{#1/#2}
\cusetup{fraction-command=\myfrac}
1/8
\cuam{1/8}\
1/2 kg
\cunum{1/2}{kg}\
4/5 °C
\cunum{4/5}{C}\
12/3 kg
\cunum{1_2/3}{kg}\
\cusetup{fraction-command=\nicefrac}
8/1
\cuam{1/8}\
1/2 kg
\cunum{1/2}{kg}\
4/5 °C
\cunum{4/5}{C}\
1 2/3 kg
\cunum{1_2/3}{kg}

fraction-inline \{input containing #1 and #2\}

Similar to fraction-command only that you don’t have to define a command to alter the output of the fraction.

\cusetup{fraction-inline={#1/#2}}
1/8
\cuam{1/8}\
1/2 kg
\cunum{1/2}{kg}\
4/5 °C
\cunum{4/5}{C}\
12/3 kg
\cunum{1_2/3}{kg}\
\cusetup{fraction-inline={\nicefrac{#2}{#1}}}8/1
\cuam{1/8}\
1/2 kg
\cunum{1/2}{kg}\
4/5 °C
\cunum{4/5}{C}\
1 2/3 kg
\cunum{1_2/3}{kg}

8.2.7 Spaces

mixed-fraction-space \{length\}

Sets the length between the fraction and the number in a mixed-fraction, default is 0.1em (because I said so; if someone has some literature or sources to look up the space, please let me know).

1 2/3
\cuam{1_2/3}\
1 3/4 kg
\cunum{1_3/4}{kg}\
10 3/4 kg
\cunum{10_3/4}{kg}\
\cusetup{mixed-fraction-space=1em}
1 2/3
\cuam{1_2/3}\
1 3/4 kg
\cunum{1_3/4}{kg}\
10 3/4 kg
\cunum{10_3/4}{kg}\
\cusetup{mixed-fraction-space=0em}
1 2/3
\cuam{1_2/3}\
1 3/4 kg
\cunum{1_3/4}{kg}\
10 3/4 kg
\cunum{10_3/4}{kg}
**cutext-space**  \(\text{cutext-space} = \{(\text{string})\}\)

\(\text{string}\) is inserted between the numeral part and the unit part when using \texttt{cutext} and \texttt{Cutext}. By default it is set to \texttt{space}. Use this option if you want to e.g. insert an unbreakable space.

- 1 kilogramme: \texttt{\cutext{1}{kg}}
- 10 kilogramme: \texttt{\Cutext{10}{kg}}
- 1 kilogramme: \texttt{\cutext{1}{kg}}
- 10 kilogramme: \texttt{\Cutext{10}{kg}}
- 1 kilogramme: \texttt{\cutext{1}{kg}}
- 10 kilogramme: \texttt{\Cutext{10}{kg}}
- 1 kilogramme: \texttt{\cutext{1}{kg}}
- 10 kilogramme: \texttt{\Cutext{10}{kg}}
- 1 kilogramme: \texttt{\cutext{1}{kg}}
- 10 kilogramme: \texttt{\Cutext{10}{kg}}

**phrase-space**  \(\text{phrase-space} = \{(\text{string})\}\)

\(\text{string}\) is inserted between the numeral part and the phrase part while using \texttt{cuam}. By default it is set to \texttt{space}. Use this option if you want to e.g. insert an unbreakable space.

(Switching to german)

- 1 Dutzend: \texttt{\cuam{12}}
- 12 Dutzend: \texttt{\cuam{144}}
- 1 Dutzend: \texttt{\cuam{12}}
- 12 Dutzend: \texttt{\cuam{144}}
- 1 Dutzend: \texttt{\cuam{12}}
- 12 Dutzend: \texttt{\cuam{144}}
- 1 Dutzend: \texttt{\cuam{12}}
- 12 Dutzend: \texttt{\cuam{144}}
- 1 Dutzend: \texttt{\cuam{12}}
- 12 Dutzend: \texttt{\cuam{144}}

**amount-unit-space**  \(\text{amount-unit-space} = \{(\text{string})\}\)

Change the spacing for \texttt{cunum} between the printed amount(s) and the unit. The default value is \texttt{thinspace}:

---

28
8.2.8 label & refs

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>recalculate-amount</code></td>
<td><code>recalculate-amount = (true/false)</code></td>
</tr>
<tr>
<td></td>
<td>Set this option to <code>true</code> if you want to change your recipes to the given number of people set by <code>set-number-of-persons</code>. Note that only those values who have a label are changed.</td>
</tr>
<tr>
<td><code>set-number-of-persons</code></td>
<td><code>set-number-of-persons = (integer)</code></td>
</tr>
<tr>
<td></td>
<td>With this option you can determine the number of people your recipes are for. Note that this option only has an effect on those who have a <code>(label)</code> given. It is set to <code>4</code> by default. Please also note the use of <code>recalculate-amount</code>.</td>
</tr>
<tr>
<td>Label</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>cup of tea</td>
<td>1 person</td>
</tr>
<tr>
<td>2 persons</td>
<td>1 kg</td>
</tr>
<tr>
<td>1 kg</td>
<td>1</td>
</tr>
<tr>
<td>1 kilogramme</td>
<td>2 persons</td>
</tr>
<tr>
<td>2 kilogramme</td>
<td>4 persons</td>
</tr>
<tr>
<td>2 kg</td>
<td>2</td>
</tr>
<tr>
<td>20 kilogramme</td>
<td>3 persons</td>
</tr>
<tr>
<td>1.5 kg</td>
<td>1.5</td>
</tr>
<tr>
<td>1.5 kilogramme</td>
<td>1.5</td>
</tr>
<tr>
<td>15 kilogramme</td>
<td>1.5</td>
</tr>
<tr>
<td>2 persons</td>
<td>1 kg</td>
</tr>
<tr>
<td>1 kg</td>
<td>1</td>
</tr>
<tr>
<td>1 kilogramme</td>
<td>2</td>
</tr>
<tr>
<td>10 kilogramme</td>
<td>2</td>
</tr>
<tr>
<td>1 person</td>
<td>0.5 kg</td>
</tr>
<tr>
<td>0.5 kg</td>
<td>0.5</td>
</tr>
<tr>
<td>0.5 kilogramme</td>
<td>0.5</td>
</tr>
<tr>
<td>5 kilogramme</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

The key-value version of \texttt{\textbackslash{label}}. It defines the label \texttt{⟨string⟩} which is originally for \texttt{⟨integer⟩} people. Please note that the \texttt{*} is mandatory as it separates the string from the integer. Each label is defined globally and must be unique.
The key-value version of \texttt{\curef}. Note that this key doesn’t save the value inside a macro but rather prints it directly into the document.

\texttt{\culabel{Schinken}{3}}
\texttt{\cusetup{get-label=Schinken}}
\texttt{\curef{Schinken}}
\texttt{\cusetup{recalculate-amount=true}}
\texttt{\curef{Schinken}}

\textbf{Note:} \texttt{\curef} is expendable.

Instead of using the first optional arguments of the commands in section 2 you may use this option. It requires a valid value and throws an error if \langle label\rangle is not defined.

\texttt{\culabel{Kaese}{3}}
\texttt{\cunum<Kaese>[m=dm]{1}{m}}
\texttt{\cunum[ref=Kaese,m=dm]{1}{m}}
\texttt{\cusetup{recalculate-amount=true}}
\texttt{\cunum[ref=Kaese,m=dm]{1}{m}}

There are units which do not depend on the number of folks you are cooking for, units measuring the temperature are an example. Changing those units with the label & ref system would be accidental and in the best case throw an error. With the following options you can add units to the “forbidden unit list”, remove them and clear the whole list entirely.

By default the list contains C, F, K and Re.
8.3 Weird options

**check-temperature**

Specify if the used temperature is below absolute zero. Currently C, F, K and Re are supported. While \cunum{0}{K} is ok, \cunum{-1}{K} raises an error, same for the others. Is set to false by default. To add new units see add-temperature-to-check.

**add-temperature-to-check**

This option adds \(\langle\text{unit-key-1}\rangle = \langle\text{minimum-value-1}\rangle\), \(\langle\text{unit-key-2}\rangle = \langle\text{minimum-value-2}\rangle\), etc. to the list of units to be checked if check-temperature is active. The argument can be a comma-separated list of \(\langle\text{unit-key}\rangle = \langle\text{minimum-value}\rangle\). This sets the allowed minimum value of \(\langle\text{unit-key}\rangle\) to \(\langle\text{minimum-value}\rangle\).

**Example:** This package implements the allowed minimum values for the temperatures C, F, K and Re to be checked if check-temperature is active using:

\begin{verbatim}
\cusetup
{
  add-temperature-to-check =
  {                      
    K = 0,            
    C = -273.15,          
    F = -459.67,         
    Re = -218.52         
  }
}
\end{verbatim}
If you want to add a new value, for example degree Rømer (which has been defined in another example) you can write:

```
cusetup
{
  add-temperature-to-check = { Ro = -135.90375 }
}
cumenu{convert-to-eV = (true/false)}
```

Converts (nearly) every unit in table 1 to electron volt or the respective derivative (if possible). Note that this option is: a) experimental and probably will forever be and b) just a joke, you are not supposed to use this units in a cookery book (and as you see this package doesn’t support the arrangement of such huge numbers). Also you may want to check the values if you really want to use them, just to be sure (I’ve checked them several times and hope they are finally correct, but mistakes happen).

```
cusetup{convert-to-eV=true}
```

---

```
cumenu{convert-to-eV = (true/false)}
```

```
cumenu{42 = (true/false)}
```

Take a good guess.

```
cumenu{42=15.5}
```

```
cumenu{42=15.5}
```

9 Bugs & Feedback

Bug reports are always welcome. If you are sending a bug report please include a minimal working example showing the bug and a short description. If you use mail please add cooking-units to the e-mail header. GMX has the habit of putting e-mails into the spam account and adding cooking-units to the header makes it easier to recognize those e-mails. It can also take longer of GitHub, but I hope I figured out how to get a mail if a new issue is created (by not me).
Feedback and requests (commands, units, etc.) are also welcome. Please also add (if possible) an example of the desired output into the minimal example (and – if by mail – add cooking-units to the header).

Furthermore, as you can see I am not able to speak too many languages (German and English to be precise; I managed to add French with the help of the internet, which is not optimal) so if you are able to speak a language not yet implemented and would like to help you can send me the translations known to you. A list of all units (and their current translations) is given in appendix A.
10 Bens Einheitensammelsurium (Bens unit Almanac)

Units are a fascinating mess. There are so many different ones which are different and the few ones which are the same (in name at least) are also different, depending on geographical position, time period and probably pure spite. We can be glad that SI-units exist.

So for those units which didn’t make it into table 1 and table 2, this section exists. Please note that this list is intended to be a just-for-fun list and not a compilation of every unit in existence with its exact value ordered by geographical and chronological position. I am sadly neither a historian nor very good in regards to languages. It would sound like fun, but ultimately, I wouldn’t have the time. Therefore I am only taking units into account which I either found in literature (stone, canna, etc.), are well known (foot) or have some other experience with them (ell) (exception: Batman). The reason I am not including units which I found in the internet is that I would like to see those units in their “natural environment”.

**unit (translation) [abbreviation]** Description, containing a quote or not. Please note that most of the units are country dependent! So the translation may not have the same amount as the word it is translated to.

**Batman** So ... You wanna be Batman? Be like Bruce Wayne? Having a secret identity? Then congratulations! You are Batman! How much Batman depends on the location, but Wikipedia is your friend in this matter.

**Rotolo**

*sicilian (Rottel*\(^\text{de}\)) Around 0.850 kg

*Auf den Fußboden lagen vier ungereifte Käse zu je zwölf Rottel, jeder ungefähr zehn Kilo schwer.* (see [1] page 51)

**Canna**

*sicilian (Rute*\(^\text{de}\), rod*\(^\text{en}\)) About 2 m bzw. about 6 foot.

*“Unsinn, Stella, Unsinn; was soll mir zustoßen? Sie kennen mich alle: Männer, die eine Rute lange sind, gibt es wenige in Palermo.”* (see [1] page 25)

**Stone** [st] 6.35 kg. According to a fellow student this unit is still used in Great Britain. I’ve also recently found it in a video game; in the german translation of said video game to be precise. Why is the german translation using stone and not kilogram (at least in braces)?

*As we had expected, the telegramm was soon followed by its sender, and the card of Mr. Cyril Overton, Trinity College, Cambridge, announced the arrival of an enormous young man, sixteen stone [101.6 kg] of solid bone and muscle, who spanned the doorway with his broad shoulders [ ... ]* (see [2] page 988)

(Story “The missing Three Quarters”)

**Foot** [ft] Equals exactly 0.3048 m or 12 in.

A bit of a strange unit (for me at least). Where I am from, people tend to have different feet sizes. Also present in the german translation of the video game that uses “Stone”.

**degree Réamur** [°Ré] Like degree Celsius, but instead of having the water boiling at 100° (Celsius), water boils at 80°. Water thankfully still freezes at 0°. Don’t think that this unit is used anymore. I think I learned about in physics.
Ell Just read the Wikipedia article.

Fun Fact: At the Stephansdom in Vienna left of the main entrance are two metal bars. One is the “Tuchelle” (drapery ell, circa 78 cm), the other the “Leinenelle” (linen ell, around 89.6 cm).

cup I think the idea of having a “cup” and it not being equal to 250 ml is a bit strange, for me at least. What other sizes can a cup have? I can imagine 500 ml, but are there other sizes?

stick A unit I’ve made fun of because it is quite regional and doesn’t make any sense for foreigners. Then I realized that I am using the unit “Packerl” in my cookery book which is also quite locally* and – even worse – the weight changes depending the content (See Packerl).

Packerlde (small bag) I’m a bit split on this unit as I don’t actually know if it exists. The reason I have the unit Packerl for my cookery book is that in Austria you can buy baking powder, (dry) Germ, Natrium, etc. in small bags (similar to stick). The problem: Depending on the content, the weight of Packerl differs. Not only that, but it can also differ between different producers (but not more than 2 g bzw. 0.07 oz). Here is a table:

| 1 Packerl Backpulver (baking powder) | 16 g (0.56 oz) |
| Natrium | 14 g (0.49 oz) |
| Vanillin(-zucker) (vanillin(-sugar)) | 8 g (0.28 oz) |
| Germ* | 7 g (0.25 oz) |

*Tockengerm (dry Germ) to be precise

For what kind of thing do I need Natrium for?

A Translations

This section contains the list of available translations. Each table shows the available translations regarding the unit symbol, the unit name (printed if \text{cutext} or \text{Cutext} is used) and the plural form (if different from the singular form). A second table shows the translations used for phrases (if given).

If a translation is not available a “—” is shown.

---

*And maybe doesn’t even exist outside my family
### A.1 English

<table>
<thead>
<tr>
<th>(unit-key)</th>
<th>printed unit</th>
<th>unitname</th>
<th>(plural)</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>kg</td>
<td>kg</td>
<td>kilogramme</td>
<td>(pounds)</td>
<td>m</td>
</tr>
<tr>
<td>dag</td>
<td>dag</td>
<td>decagramme</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>gramme</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>oz</td>
<td>oz</td>
<td>ounce</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>lb</td>
<td>lb</td>
<td>pound</td>
<td>(pounds)</td>
<td>m</td>
</tr>
<tr>
<td>ºC</td>
<td>degree Celsius</td>
<td>degree Celsius</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ºF</td>
<td>degree Fahrenheit</td>
<td>degree Fahrenheit</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ºRé</td>
<td>degree Réaumur</td>
<td>degree Réaumur</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>K</td>
<td>Kelvin</td>
<td>Kelvin</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>day</td>
<td>(days)</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
<td>hour</td>
<td>(hours)</td>
<td>m</td>
</tr>
<tr>
<td>min</td>
<td>min</td>
<td>minute</td>
<td>(minutes)</td>
<td>m</td>
</tr>
<tr>
<td>s</td>
<td>s</td>
<td>second</td>
<td>(seconds)</td>
<td>m</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>metre</td>
<td>(metres)</td>
<td>m</td>
</tr>
<tr>
<td>dm</td>
<td>dm</td>
<td>decimetre</td>
<td>(decimetres)</td>
<td>m</td>
</tr>
<tr>
<td>cm</td>
<td>cm</td>
<td>centimetre</td>
<td>(centimetres)</td>
<td>m</td>
</tr>
<tr>
<td>mm</td>
<td>mm</td>
<td>millimetre</td>
<td>(millimetres)</td>
<td>m</td>
</tr>
<tr>
<td>in</td>
<td>in</td>
<td>inch</td>
<td>(inches)</td>
<td>m</td>
</tr>
<tr>
<td>l</td>
<td>ℓ</td>
<td>litre</td>
<td>(litres)</td>
<td>m</td>
</tr>
<tr>
<td>dl</td>
<td>dl</td>
<td>decilitre</td>
<td>(decilitres)</td>
<td>m</td>
</tr>
<tr>
<td>cl</td>
<td>cl</td>
<td>centilitre</td>
<td>(centilitres)</td>
<td>m</td>
</tr>
<tr>
<td>ml</td>
<td>ml</td>
<td>millilitre</td>
<td>(millilitres)</td>
<td>m</td>
</tr>
<tr>
<td>cal</td>
<td>cal</td>
<td>calorie</td>
<td>(calories)</td>
<td>m</td>
</tr>
<tr>
<td>kcal</td>
<td>kcal</td>
<td>kilocalorie</td>
<td>(kilocalories)</td>
<td>m</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>joule</td>
<td>(joules)</td>
<td>m</td>
</tr>
<tr>
<td>kJ</td>
<td>kJ</td>
<td>kilojoule</td>
<td>(kilojoules)</td>
<td>m</td>
</tr>
<tr>
<td>eV</td>
<td>eV</td>
<td>electron volt</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>pn</td>
<td>pinch</td>
<td>pinch</td>
<td>(pinches)</td>
<td>m</td>
</tr>
<tr>
<td>EL</td>
<td>tbsp.</td>
<td>tablespoon</td>
<td>(tablespoons)</td>
<td>m</td>
</tr>
<tr>
<td>TL</td>
<td>tsp.</td>
<td>teaspoon</td>
<td>(teaspoons)</td>
<td>m</td>
</tr>
<tr>
<td>csp</td>
<td>csp.</td>
<td>coffeespoonful</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>dsp</td>
<td>dsp.</td>
<td>dessertspoonful</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ssp</td>
<td>ssp.</td>
<td>saltspoonful</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Msp</td>
<td>Msp.</td>
<td>—</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>decimal-mark</td>
<td>—</td>
<td>.</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>one(m)</td>
<td>—</td>
<td>one</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>one(f)</td>
<td>—</td>
<td>one</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>one(n)</td>
<td>—</td>
<td>one</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>(unit-key)</td>
<td>printed unit</td>
<td>unitname</td>
<td>(plural)</td>
<td>gender</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------</td>
<td>----------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>kg</td>
<td>kg</td>
<td>kilogram</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>dag</td>
<td>dag</td>
<td>decagram</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>gram</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>oz</td>
<td>oz</td>
<td>ounce</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>lb</td>
<td>lb</td>
<td>pound</td>
<td>(pounds)</td>
<td>m</td>
</tr>
<tr>
<td>°C</td>
<td>°C</td>
<td>degree Celsius</td>
<td>(degrees Celsius)</td>
<td>m</td>
</tr>
<tr>
<td>°F</td>
<td>°F</td>
<td>degree Fahrenheit</td>
<td>(degrees Fahrenheit)</td>
<td>m</td>
</tr>
<tr>
<td>°Ré</td>
<td>°Ré</td>
<td>degree Réaumur</td>
<td>(degrees Réaumur)</td>
<td>m</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
<td>kelvin</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>day</td>
<td>(days)</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
<td>hour</td>
<td>(hours)</td>
<td>m</td>
</tr>
<tr>
<td>min</td>
<td>min</td>
<td>minute</td>
<td>(minutes)</td>
<td>m</td>
</tr>
<tr>
<td>s</td>
<td>s</td>
<td>second</td>
<td>(seconds)</td>
<td>m</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>meter</td>
<td>(meters)</td>
<td>m</td>
</tr>
<tr>
<td>dm</td>
<td>dm</td>
<td>decimeter</td>
<td>(decimeters)</td>
<td>m</td>
</tr>
<tr>
<td>cm</td>
<td>cm</td>
<td>centimeter</td>
<td>(centimeters)</td>
<td>m</td>
</tr>
<tr>
<td>mm</td>
<td>mm</td>
<td>millimeter</td>
<td>(millimeters)</td>
<td>m</td>
</tr>
<tr>
<td>in</td>
<td>in</td>
<td>inch</td>
<td>(inches)</td>
<td>m</td>
</tr>
<tr>
<td>ℓ</td>
<td>ℓ</td>
<td>liter</td>
<td>(liters)</td>
<td>m</td>
</tr>
<tr>
<td>dl</td>
<td>dl</td>
<td>deciliter</td>
<td>(deciliters)</td>
<td>m</td>
</tr>
<tr>
<td>cl</td>
<td>cl</td>
<td>centiliter</td>
<td>(centiliters)</td>
<td>m</td>
</tr>
<tr>
<td>ml</td>
<td>ml</td>
<td>milliliter</td>
<td>(milliliters)</td>
<td>m</td>
</tr>
<tr>
<td>cal</td>
<td>cal</td>
<td>calorie</td>
<td>(calories)</td>
<td>m</td>
</tr>
<tr>
<td>kcal</td>
<td>kcal</td>
<td>kilocalorie</td>
<td>(kilocalories)</td>
<td>m</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>joule</td>
<td>(joules)</td>
<td>m</td>
</tr>
<tr>
<td>kJ</td>
<td>kJ</td>
<td>kilojoule</td>
<td>(kilojoules)</td>
<td>m</td>
</tr>
<tr>
<td>eV</td>
<td>eV</td>
<td>electron volt</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>pn</td>
<td>pn.</td>
<td>pinch</td>
<td>(pinches)</td>
<td>m</td>
</tr>
<tr>
<td>EL</td>
<td>tbsp.</td>
<td>tablespoon</td>
<td>(tablespoons)</td>
<td>m</td>
</tr>
<tr>
<td>TL</td>
<td>tsp.</td>
<td>teaspoon</td>
<td>(teaspoons)</td>
<td>m</td>
</tr>
<tr>
<td>csp</td>
<td>csp.</td>
<td>coffeespoonful</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>dsp</td>
<td>dsp.</td>
<td>dessertspoonful</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ssp</td>
<td>ssp.</td>
<td>saltspoonful</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Msp</td>
<td>Msp.</td>
<td>—</td>
<td>—</td>
<td>m</td>
</tr>
<tr>
<td>decimal-mark</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>m</td>
</tr>
<tr>
<td>one(m)</td>
<td>—</td>
<td>one</td>
<td>—</td>
<td>m</td>
</tr>
<tr>
<td>one(f)</td>
<td>—</td>
<td>one</td>
<td>—</td>
<td>m</td>
</tr>
<tr>
<td>one(n)</td>
<td>—</td>
<td>one</td>
<td>—</td>
<td>m</td>
</tr>
</tbody>
</table>
### A.3 German

<table>
<thead>
<tr>
<th>unit-key</th>
<th>printed unit</th>
<th>unitname</th>
<th>(plural)</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>kg</td>
<td>kg</td>
<td>Kilogramm</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>dag</td>
<td>dag</td>
<td>Dekagramm</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>Gramm</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>oz</td>
<td>oz</td>
<td>Unze</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>lb</td>
<td>lb</td>
<td>Pfund</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>C</td>
<td>°C</td>
<td>Grad Celsius</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>F</td>
<td>°F</td>
<td>Grad Fahrenheit</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Re</td>
<td>°Ré</td>
<td>Grad Réamur</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
<td>Kelvin</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>Tag</td>
<td>(Tage)</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
<td>Stunde</td>
<td>(Stunden)</td>
<td>f</td>
</tr>
<tr>
<td>min</td>
<td>min</td>
<td>Minute</td>
<td>(Minuten)</td>
<td>f</td>
</tr>
<tr>
<td>s</td>
<td>s</td>
<td>Sekunde</td>
<td>(Sekunden)</td>
<td>f</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>Meter</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>dm</td>
<td>dm</td>
<td>Dezimeter</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>cm</td>
<td>cm</td>
<td>Centimeter</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>mm</td>
<td>mm</td>
<td>Millimeter</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>in</td>
<td>in</td>
<td>Zoll</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>l</td>
<td>l</td>
<td>Liter</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>dl</td>
<td>dl</td>
<td>Deziliter</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>cl</td>
<td>cl</td>
<td>Centiliter</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ml</td>
<td>ml</td>
<td>Milliliter</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>cal</td>
<td>cal</td>
<td>Kalorie</td>
<td>(Kalorien)</td>
<td>f</td>
</tr>
<tr>
<td>kcal</td>
<td>kcal</td>
<td>Kilokalorie</td>
<td>(Kilokalorien)</td>
<td>f</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>Joule</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>kJ</td>
<td>kJ</td>
<td>Kilojoule</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>eV</td>
<td>eV</td>
<td>Elektronenvolt</td>
<td></td>
<td>f</td>
</tr>
<tr>
<td>pn</td>
<td>Prise</td>
<td>Prise</td>
<td>(Prisen)</td>
<td>f</td>
</tr>
<tr>
<td>EL</td>
<td>EL</td>
<td>Esslöffel</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>TL</td>
<td>TL</td>
<td>Teelöffel</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>csp</td>
<td>KL</td>
<td>Mokkalöffel</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>dsp</td>
<td>dsp.</td>
<td></td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>ssp</td>
<td>ssp.</td>
<td></td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>Msp</td>
<td>Msp.</td>
<td></td>
<td>(Messerspitzen)</td>
<td>f</td>
</tr>
<tr>
<td>decimal-mark</td>
<td>—</td>
<td></td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>one(m)</td>
<td>—</td>
<td>ein</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>one(f)</td>
<td>—</td>
<td>eine</td>
<td></td>
<td>m</td>
</tr>
<tr>
<td>one(n)</td>
<td>—</td>
<td>ein</td>
<td></td>
<td>m</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Phrase-key</th>
<th>phrase</th>
<th>(plural)</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Dutzend</td>
<td></td>
<td>n</td>
</tr>
</tbody>
</table>
Some further phrases, just to write them down (they are not implemented, as they are barely used).

<table>
<thead>
<tr>
<th>(number)</th>
<th>name</th>
<th>Note</th>
<th>(plural)</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>Schock</td>
<td>(5 Dutzend, 12 * 5)</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>144</td>
<td>Gros</td>
<td>(12 Dutzend, 12 * 12)</td>
<td></td>
<td>n</td>
</tr>
<tr>
<td>1728</td>
<td>Großgros</td>
<td>(12 Groß, 12 * 144)</td>
<td></td>
<td>n</td>
</tr>
</tbody>
</table>

Note that Großgros has other (probably more common) synonyms.
### A.4 French

<table>
<thead>
<tr>
<th>(unit-key)</th>
<th>printed unit</th>
<th>unitname</th>
<th>(plural)</th>
<th>gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>kg</td>
<td>kg</td>
<td>kilogramme</td>
<td>(kilogrammes)</td>
<td>m</td>
</tr>
<tr>
<td>dag</td>
<td>dag</td>
<td>décagramme</td>
<td>(décagrammes)</td>
<td>m</td>
</tr>
<tr>
<td>g</td>
<td>g</td>
<td>gramme</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>oz</td>
<td>oz</td>
<td>once</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>lb</td>
<td>lb</td>
<td>livre</td>
<td>(livres)</td>
<td>f</td>
</tr>
<tr>
<td>C</td>
<td>°C</td>
<td>degré Celsius</td>
<td>(degrés Celsius)</td>
<td>m</td>
</tr>
<tr>
<td>F</td>
<td>°F</td>
<td>kelvin</td>
<td>(kelvins)</td>
<td>m</td>
</tr>
<tr>
<td>Re</td>
<td>°Ré</td>
<td>échelle Réaumur</td>
<td>(degrés Réaumur)</td>
<td>m</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
<td>degré Fahrenheit</td>
<td>(degrés Fahrenheit)</td>
<td>m</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>jour</td>
<td>(jours)</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>h</td>
<td>heure</td>
<td>(heures)</td>
<td>f</td>
</tr>
<tr>
<td>min</td>
<td>min</td>
<td>minute</td>
<td>(minutes)</td>
<td>f</td>
</tr>
<tr>
<td>s</td>
<td>s</td>
<td>seconde</td>
<td>(secondes)</td>
<td>f</td>
</tr>
<tr>
<td>m</td>
<td>m</td>
<td>mètre</td>
<td>(mètres)</td>
<td>m</td>
</tr>
<tr>
<td>dm</td>
<td>dm</td>
<td>décimètre</td>
<td>(décimètres)</td>
<td>m</td>
</tr>
<tr>
<td>cm</td>
<td>cm</td>
<td>centimètre</td>
<td>(centimètres)</td>
<td>m</td>
</tr>
<tr>
<td>mm</td>
<td>mm</td>
<td>millimètre</td>
<td>(millimètres)</td>
<td>m</td>
</tr>
<tr>
<td>in</td>
<td>po</td>
<td>pouce</td>
<td>(pouces)</td>
<td>m</td>
</tr>
<tr>
<td>l</td>
<td>L</td>
<td>litre</td>
<td>(litres)</td>
<td>m</td>
</tr>
<tr>
<td>dl</td>
<td>dL</td>
<td>décilitre</td>
<td>(décilitres)</td>
<td>m</td>
</tr>
<tr>
<td>cl</td>
<td>cL</td>
<td>centilitre</td>
<td>(centilitres)</td>
<td>m</td>
</tr>
<tr>
<td>ml</td>
<td>mL</td>
<td>millilitre</td>
<td>(millilitres)</td>
<td>m</td>
</tr>
<tr>
<td>cal</td>
<td>cal</td>
<td>calorie</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>kcal</td>
<td>kcal</td>
<td>kilocalorie</td>
<td>(kilocalories)</td>
<td>m</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>joule</td>
<td>(joules)</td>
<td>m</td>
</tr>
<tr>
<td>kJ</td>
<td>kJ</td>
<td>kilojoule</td>
<td>(kilojoules)</td>
<td>m</td>
</tr>
<tr>
<td>eV</td>
<td>eV</td>
<td>électron-volt</td>
<td>(électron-volts)</td>
<td>m</td>
</tr>
<tr>
<td>pn</td>
<td>pinch</td>
<td>pincée</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>EL</td>
<td>c.à.s.</td>
<td>cuillère à soupe</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>TL</td>
<td>c.à.c.</td>
<td>cuillère à café</td>
<td>f</td>
<td></td>
</tr>
<tr>
<td>csp</td>
<td>csp.</td>
<td>—</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>dsp</td>
<td>dsp.</td>
<td>—</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>ssp</td>
<td>ssp.</td>
<td>—</td>
<td>m</td>
<td></td>
</tr>
<tr>
<td>Msp</td>
<td>Msp.</td>
<td>—</td>
<td>m</td>
<td></td>
</tr>
</tbody>
</table>

If the spoons should be extra full:
- cuillère à soupe rase
- cuillère à café rase
B US, Imperial and Other units

As source [5] has been used for imperial units, while [4] and [3] were used for U.S. units. I hope someone will find this bringing together useful.

1 yard = 0.9144 m (exact)
1 yard = 3 foot
1 yard = 36 Inch
1 Inch = 0.0254 m (also exact)

1 liter = 1 dm$^3$
1 gallon = 4 Quart
1 gallon = 8 Pint
1 gallon = 32 Gill
1 gallon = 160 fl.oz
1 fl.oz = 0.028 413 062 5 liter
1 fl.oz U.S. = 0.029 573 529 562 5 liter

Note 1: I think the American fl.oz U.S. is more common. Maybe. Most bottles have something like 10 fl.oz, which they say is equal to 30 ml. This would work really well with fl.oz U.S.

Note 2: Sometimes “fl. oz” is written without the dot. I am also not sure what kind of spacing has to be between “fl.” and “oz” (currently using \thinspace).

Note 3: This maybe sounds stupid, but could we introduce something like “flouz”, “floiz” and “floez”? “flouz” would be “fl. oz U.S.”, “floiz” would be “Imperial fl. oz.” and “floez” would simply be equal to 30 ml?

For “stick” see [6].

1 lb = 0.453 592 37 kg (exact)
1 lb = 16 oz
1 lb = $\frac{1}{14}$ st
1 lb = $\frac{1}{12}$ ounce troy
1 lb = 4 stick

1 cup $\approx$ 0.25 litre = 250 ml
1 cup U.S. = 8 fl. oz U.S.
1 tablespoon $\approx$ 0.015 litre = 15 ml
1 tablespoon U.S. = $\frac{1}{2}$ fl. oz U.S.
1 teaspoon $\approx$ 0.005 litre = 5 ml
1 teaspoon U.S. = $\frac{1}{6}$ fl. oz U.S.

Note 1: I tested the approximation for tablespoon with water (1 mg $\approx$ 1 mg) and the approximation looks good enough. It of course depends on how full you fill your spoon.

References


Change History

2016/06/11
General: Added the package option to
load 'fmtcount'. ........................ 1

2016/08/31
General: Fixed calculation: degree
Reamur to eV ............................ 1
Initial version ............................ 1

2016/09/03
General: Added units 'ssp', 'csp', 'dsp' 1
British English: 'pinch' is written in
full ................................... 1
English unit: litre (and only litre)
uses the curly l ℓ now ............... 1
Separated Messerspitze and pinch ... 1

2016/09/05
General: New message:
'obsolete-command' ..................... 1
Replaced \cufrac by \cuam .......... 1

2016/09/09
General: \@@_calculate_input_and_-
store_in:nN optimiert durch neue
property-key: single. ................. 1
Add 'single' to property list of
singlekeys. ............................. 1
Changed name from
\@@_cunum_parse_range (and
derivatives) to
\@@_cutext_parse_range. ............ 1
Changed name from \@@_parse_-
fraction_in_input:www to
\@@_parse_mixed_fraction_in_-
input:www ................................ 1
Corrected mistake: 'Elektronenvolt'
(note uppercase L) to
'Elektronenvolt' in german. ......... 1
Delete 'single' from property lists of
singlekeys cause it is not as safe as
I thought. .............................. 1

In \@@_cutext_default:nnn it is
only checked once if a range is
inside. .................................. 1

2016/09/16
General: Only use \phantom if the
argument (for \phantom) is not
empty. .................................. 1

2016/09/26
General: \cuaddsinglekeys now tests
if the unit exists (it didn’t before). . 1
New option (and needed macros):
add-temperature-to-check. ............ 1
New option: ‘round-half’. ............. 1
Recalculated all electron volt values
for conversion (as 'kg' was wrong
before). Let’s hope they are correct
this time. ................................. 1
Replaced \prop_clear_new:c by
\prop_clear:c ............................ 1

2016/10/19
General: 'convert-to-eV’ now also as
optional argument available. ........ 1
Option 'load-time-option' now spells
'available' correct. ..................... 1
Update of documentation. ............. 1
Use \keys_set:nn only if second
argument is not empty. ............... 1

2016/10/28
General: \cutext (and \Cutext) and
\cuam now parse their input like
\cunum. This is needed as they also
need to be changed. .................... 1
Start implementation of “Change
recipe from n to m persons.” ........ 1

2016/10/29
General: Tiding code: Now every
command is separated into a “calc”
function, a “print numeric value”
and a “print unit” (if there) function. At least, that’s the plan.  

2016/10/30  
General: Fractions should now deal correctly with minus signs.  

2016/11/07  
General: Finished writing v1.10.  

2016/11/13  
General: \cutext, \Cutext and \cuam check their input, allows conversion of units.  
Change amounts for specific number of persons.  
New commands: \culabel and \curef.  
New commands: \declarecookingunit and \providecookingunit.  
New options: \cuam-version and \cutext-version  
New options: \cutext-change-to-cumum, \cutext-change-unit and \cutext-space.  
New options: \recalculate-amount and \set-number-of-persons, \label, get-label, \ref.  
2017/03/10  
General: \curef is now defined by \NewExpandableDocumentCommand instead of the Declare variant.  
Removed \translate and others from code and replaced them with wrapper-macros.  
Removed things like ‘cu-unit’ from translate input and placed them into separate t\'s.  
2017/10/23  
General: Added “phrases”.  
Added unit “stick” (of butter).  
New option: \amount-unit-space.  
New option: \phrase-space.  
New option: \print-numerals.  
New option: \set-cutext-translation-message.  
New option: \use-phrases.  
Now checks for ranges if both values can be printed as numerals (if activated) (bug fix).  
Replaced translator by translations.  
Reworked quite a lot of code.  
2018/04/20  
Allow round precision to be negative.  
Change large portions of code.  
Cooking Units-keys are not allowed to contain either “,” or “/”.  
Fix argument specifiers.  
Introduce key-groups (weight, volume, etc.).  
New feature: Hooks.  
New Option: 42.  
New option: add-unit-to-group.  
New option: erase-all-options-for.  
New options: expand-both, expand-amount, expand-unit.  
New parsing algorithm. Hopefully better error recovery (if signs for fractions are in wrong order e.g.)  
Option: add-natural-unit.  
2018/06/05  
General: \set-unknown-message: Fix default value.  
Add “range-sign” for translations (not usable yet).  
Bugfix (phrases): Use the phrase from the first amount to check the second (and don’t parse through the second amount).  
Bugfix (unit-change): convert-to-eV can be again used as a local argument.  
true) will print the second word small.  
Change (amount-not-known): Message change a bit.  
Convert clist to seq if possible.  
Fix some more argument specifiers.  
Improve error-recovery by a lot.  
Reremoving of some internal commands.  
Reworking of internal commands.  
This version introduces mayor internal changes. For users not many things change.  
2018/09/24  
General: Changes prefix from cooking_units to cookingunits.  
Improved french (not in general, only for this package).
Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols
<group> .................................................. 17
<unit> .................................................. 16
@@ commands:
@@_calculate_input_and_store_in:nN ............................................. 42
@@_cunum_parse_range .................................................. 42
@@_cutext_default:nnn .................................................. 42
@@_cutext_parse_range .................................................. 42
@@_parse_fraction_in_input:www ............................................. 42
@@_parse_mixed_fraction_in_input ............................................
\Cuaddkeys .................................................. 8, 10, 11
\cuaddoptionfor .................................................. 15, 18, 19
\cuaddsinglekeys .................................................. 8, 10
\cuaddtokeys .................................................. 8, 11
\cuaddtounitgroup .................................................. 17, 18
\cualr .................................................. 3, 14, 14, 20, 21, 21, 23, 23, 28
\cudeclareunitgroup .................................................. 17, 17
\cudefinekeys .................................................. 8, 9
\cudefinename .................................................. 12
\cudefinephase .................................................. 14
\cudefinesinglekey .................................................. 8, 9
\cudefinesymbol .................................................. 13
\culabel .................................................. 5, 29
\cunum .................................................. 3, 4, 19, 21, 21, 21, 21, 23, 23, 28
\curef .................................................. 5, 30
\cusetoptionfor .................................................. 15, 18, 19
\cusetup .................................................. 3, 15
\cutext .................................................. 3, 4, 6, 12, 16,
19, 20, 20, 21, 21, 21, 21, 22, 23, 23, 27, 35
\declarecookingderivatives .................................................. 8
\declarecookingunit .................................................. 6, 7
\newcookingunit .................................................. 6, 7
\providecookingunit .................................................. 6, 7

Numbers
42 .................................................. 32

Add: add-natural-unit .................................................. 32
add-option-for .................................................. 19
add-option-for<unit-key> .................................................. 18
add-special-sign .................................................. 21
add-temperature-to-check .................................................. 31
add-unit-to-group .................................................. 18
amount-unit-space .................................................. 28

Check: check-temperature .................................................. 31
\command .................................................. 26
commands-add-hook .................................................. 20
convert-fraction .................................................. 26
convert-to-eV .................................................. 32
\cuaddkeys .................................................. 10
\cuaddoptionfor .................................................. 15
\cuaddsinglekeys .................................................. 10, 42
\cuaddtokeys .................................................. 11
\cuaddtounitgroup .................................................. 17
\cualr .................................................. 42, 42
\cuaddoptionfor .................................................. 20
\cuaddversion .................................................. 20
\cunsetoptionfor .................................................. 15
\cudeclareunitgroup .................................................. 17
\cudedefinename .................................................. 9
\cudedefinesinglekey .................................................. 9
\cudefinesymbol .................................................. 13
\cufrac .................................................. 42
\culabel .................................................. 43
\cunum .................................................. 42
\cuaddoptionfor .................................................. 20
\curef .................................................. 43
\curef-add-forbidden-unit .................................................. 30
\curef-clear-forbidden-units .................................................. 30
\curef-remove-forbidden-unit .................................................. 30
\cusetoptionfor .................................................. 15
\cusetup .................................................. 15
\cutext .................................................. 42, 43
\cutext .................................................. 42, 43

A

C

B