The euro package

Melchior FRANZ

March 7, 2005

Abstract

This article describes the euro package[1], which converts arbitrary currency units into each other using the *Euro* (official symbol €[2]) as base unit, and formats monetary amounts in almost any desired way. The respective conversion rates are predefined for the national currency units of the *Eurozone* countries, every other currency unit can easily be added. The package allows to type e. g. \ATS{17.6} to get something like 17,60 öS (1,28 €). All necessary calculations are with high precision done by the great fp package[3] by MICHAEL MEHLICH, which you have to install to get this package working. The package complies with the concerned rules and orders of the European Council.

Contents

[1] This package has version number 1.1, last revised 2005/03/07.

[2] The € symbol in this documentation is 'handmade'—it is really ugly and not to be used in real documents. Take the well designed symbol defined e. g. in the eurosym package by HENRIK THEILING eventually using the eurofont package by ROWLAND MCDONNELL to select the proper fo(u)nt/shape.

[3] ⟨CTAN//macros/latex/contrib/other/fp/⟩

`EUR`	Europe	`GRD`	Greece
`ATS`	Austria	`IEP`	Ireland
`BEF`	Belgium	`ITL`	Italy
`DEM`	Germany	`LUF`	Luxembourg
`ESP`	Spain	`NLG`	The Netherlands
`FIM`	Finland	`PTE`	Portugal
`FRF`	France		

Table 1: ISO currency codes of the *Euro* and of the twelve *Euro-zone* countries.

1 Introduction

Twelve of the fifteen members of the European Union have agreed to replace their national currencies by a common currency, the *Euro.* Before the 1st of January 2001 the Euro was only a unit for calculations, with fixed conversion rates for each of the old currencies. Since then the Euro is actual money with coins and bank notes. The Francs, the D-Mark, the Schilling, they are history. Several further countries will join the European Union over the next years, and sooner or later also the so-called *Euro-zone.* It's still common practice in Europe to write monetary amounts both in the national currency and in the new € currency, thus helping to become familiar with the *Euro.*

But it's a nuisance to convert every value by hand or even with a calculator. Let our computers do all the hard work. Although TEX is unable to do such calculations with its basic arithmetic functions, there's a way out: The `fp` package allows to deal with numbers ranging from −999999999999999999,999999999999999999 to +999999999999999999,999999999999999999 with high precision.

Since there are so many currencies and different possible formats, this package provides only a customizable interface. All parameters are predefined in a common way, though, so that no further definitions will be necessary in most cases. The package tries to load an optional file '`euro.cfg`' where such local customizing commands may be placed in (see section 3.7 for further explanation).

2 How to use the package

2.1 The main macro

`\EURO`

The `\EURO` command does the main work. It takes the *source currency unit,* an optional *destination currency unit,* both as three-digits ISO currency codes (see table 1), and the respective *money amount.*

```
\EURO{⟨source currency unit⟩}[⟨destination currency unit⟩]{⟨amount⟩}
```

A dot is to be used as *decimal 'point'* separating a number's integer and fractional part. Numbers may of course be negative. If the optional destination currency unit is omitted, the *Euro* is taken.

`\EURO{BEF}[FIM]{70}`	70 BEF (10,32 FIM)	
`\EURO{FIM}[BEF]{70}`	70 FIM (474,93 BEF)	
`\EURO{BEF}[BEF]{70}`	70 BEF (70 BEF)	[surprised?]
`\EURO{FIM}{17.8}`	17,80 FIM (2,99 Euro)	

2.2 How to change the local style

`\EUROSYM`

The ISO currency codes don't really look good in running text. You can replace them by some other currency symbol or tag using the `\EUROSYM` command.

`\EUROSYM{`⟨*currency unit*⟩`}{`⟨*symbol*⟩`}`

We Austrians, for example, prefer to write 'öS' instead of 'ATS'. We don't have to write `\EUROSYM{ATS}{\"oS}` though, because this national style (like those of some other countries) is already predefined. If your national style is still missing, please drop me a line, and I will insert it. Meanwhile the ISO currency codes are used in such cases.

`\EURO{FRF}{16.3}`	16,30 FRF (2,48 Euro)
`\EUROSYM{FRF}{Francs}`	
`\EURO{FRF}{16.3}`	16,30 Francs (2,48 Euro)

The official Euro symbol is not provided by default, since there are a lot of different fonts and methods to produce an € sign. Type `\EUROSYM{EUR}{`⟨*euro symbol*⟩`}` to replace the word 'Euro' with your local € sign. (I'm still hoping that some day the text companion fonts will contain the official in addition or instead of the phantasy *Euro* symbols.) This documentation, for example, says `\EUROSYM{EUR}{\euro}` because it has its own `\euro` symbol built-in. (Isn't it beautiful?)

`\EURO{LUF}{137.8}`	137,80 LUF (3,42 Euro)
`\EUROSYM{EUR}{\euro}`	
`\EURO{LUF}{137.8}`	137,80 LUF (3,42 €)

Writing e. g. `\EURO{ATS}{...}` again and again is cumbersome, so it's recommended to define macros for the most often used currency units. You may put such definitions into your local configuration file (see section 3.7).

`\newcommand*\ATS{\EURO{ATS}}`	
`\ATS{9.90}`	9,90 öS (0,72 €)
`\ATS[ITL]{19.90}`	19,90 öS (2 800 Lit.)

2.3 The package options

The `euro` package provides six package options. See table 2 for a short demonstration.

eco The eco option typesets monetary amounts as, I think, economists prefer it, with ISO codes preceding the values.

dots The dots option inserts dots as three-digits group separators. (Default is a little space like ‖.)

`\EURO{ATS}{1000}` with	
(default settings)	1 000 öS (72,67 €)
dots	1.000 öS (72,67 €)
endash	1 000,– öS (72,67 €)
emdash	1 000,— öS (72,67 €)
zeros	1 000,00 öS (72,67 €)
eco	ATS 1 000 (EUR 72,67)
eco,endash	ATS 1 000,– (EUR 72,67)
eco,emdash	ATS 1 000,— (EUR 72,67)

Table 2: Package options

table Turns the *table mode* globally on. While integers are normally printed without decimal point and fractional part (sic!), it may especially in tables be desirable to typeset a decimal point followed by a dash. See section 3.4 for how to keep this behavior local to a particular table.

emdash Lets the table dashes be '—'. Doesn't turn on *table mode.* (This option is selected by default, so it doesn't make much sense at the moment. But this might change some day.)

endash Lets the table dashes be '–'. Doesn't turn on *table mode.*

zeros Lets zeros be printed instead of dashes. Doesn't turn on *table mode.*

2.4 Adding currency units

If some day further countries will join the *Euro-zone,* you can easily add their currency units to your document files or, better, to your local '`euro.cfg`' configuration file. The `euro` package will then, of course, be updated as soon as possible.

```
\EUROADD{⟨ISO code⟩}{⟨national style/symbol⟩}{⟨conversion rate⟩}
```

The conversion rate describes, how many units of the new currency equal 1 €. You can also add currencies, that will probably never become member of the *Euro-zone,* like the US-$. But be aware that such conversion rates may change daily/hourly! See ⟨`http://europa.eu.int/`⟩ for the actual rates. The used ISO code serves as identifier for the respective currency unit.

```
%  1 Euro = 0.6694 Pound Sterling    (March 11, 1999)
%
\EUROADD{GBP}{\textsterling}{0.6694}
\newcommand*\GBP{\EURO{GBP}}
```

`\GBP{47.11}`	47,11 *£* (70,38 €)
`\GBP[FRF]{47.11}`	47,11 *£* (461,64 FRF)
`\EURO{EUR}[GBP]{1}`	1 € (0,67 *£*)

The pound symbol in this example looks a bit too slanted. You'd better take `\textsterling` from the `textcomp` package, which I didn't.

3 Further customization

The desirable format for monetary amounts may differ from country to country and from purpose to purpose. The `euro` package uses three kinds of format strings to customize the format, each of which may consist of some *reserved keywords* and any other TEX commands. It's a good idea to put such commands in your local configuration file (see section 3.7).

3.1 The main format

The main format defines how source and destination currency shall be arranged.

`\EUROFORMAT{main}{`⟨*main format string*⟩`}`

The *reserved keywords* `\in` and `\out` stand for the source and the destination currency. The following table shows some possible formats with the first line showing the default definition.

	`\EURO{ATS}{30}` yields
`\EUROFORMAT{main}{\in\ (\out)}`	30 öS (2,18 €)
`\EUROFORMAT{main}{\in\slash\out}`	30 öS/2,18 €
`\EUROFORMAT{main}{\in\ ($=$\,\out)}`	30 öS (= 2,18 €)
`\EUROFORMAT{main}{\in\ (that is \out)}`	30 öS (that is 2,18 €)
`\EUROFORMAT{main}{\out}`	2,18 €

3.2 The currency format

The currency format defines how the source and the destination currency shall be formatted.

`\EUROFORMAT{in}{`⟨*currency format string*⟩`}`
`\EUROFORMAT{out}{`⟨*currency format string*⟩`}`

The *reserved keywords* `\val`, `\iso` and `\sym` have a special meaning, but almost every other TEX command may be used as well.

`\val`	monetary amount
`\iso`	ISO currency code
`\sym`	national currency style/tag (if defined; ISO code else)

For some currency units the `\sym` command produces only the ISO code instead of the national style, which I didn't know. This will (with your help!) be changed in future releases. The following table shows the most obvious arrangements. All said about the `in` format string is also true for the `out` format.

with `\EUROFORMAT{main}{\in}` and	`\EURO{ATS}{30}` yields
`\EUROFORMAT{in}{\val~\sym}`	30 öS
`\EUROFORMAT{in}{\val~\iso}`	30 ATS
`\EUROFORMAT{in}{\sym~\val}`	öS 30
`\EUROFORMAT{in}{\iso~\val}`	ATS 30

3.3 The number format

The number format, finally, defines how the numbers themselves should be typeset considering positive/negative numbers, rounding and so on.

```
\EUROFORMAT{all}{⟨number format⟩}
\EUROFORMAT{⟨ISO currency code⟩}{⟨number format⟩}
```

The definitions labeled with `all` will be used for all currency units. By using `\EUROFORMAT` with one of the three-digits ISO codes as the first argument, these general settings can be overridden for this currency. Only the *Italian Lira* is defined this way by default, because it wouldn't make sense to write *Lira* amounts with two fractional digits like the other European currencies.[4]

The *number format string* may contain the commands `\form`, `\round`, `\zero`, `\plus`, and `\minus`, each of which takes one or more arguments. You have only to mention the points that shall differ from the default behavior. Such a number format might look horrible like this

```
\EUROFORMAT{all}{%
  \form{\,}{,}{\,}%
  \round{-2}%
  \zero{0}{0}{}%
  \plus{}{}%
  \minus{\(-\)}{}}
```

which, by fortune, is the predefined default format[5], but it might as well be empty or contain only one of the commands and its arguments. Here again all arguments are predefined quite reasonably, so you won't have to bother with these verbose definitions.

[4] The package, for example, says something like `\EUROFORMAT{ITL}{\round{0}}`.

[5] Well, that is a lie: In fact, it is only the equivalent of the default definition, which is defined in another way.

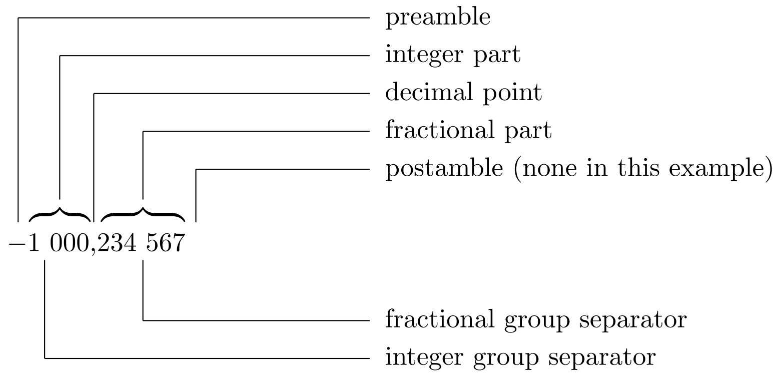

Figure 1: Elements that a number may consist of.

To make further explanations easier, figure 1 describes the elements that a number may consist of (using my own nomenclature).

3.3.1 The `\form` command

`\form` The `\form` command takes three arguments: the *integer group separator*, the *decimal point* and the *fractional group separator*.

```
\form
  {⟨integer group separator⟩}
  {⟨decimal point⟩}
  {⟨fractional group separator⟩}
```

group separators: separate groups of three decimal digits to improve legibility of bigger numbers. It's a small space by default, but taking a space, a dot or just nothing might be desirable in some cases.

decimal point: separates a number's integer part from the fractional part. It's a comma by default.

`\form{.}{,}{.}`	1.000,12
`\form{\,}{,}{\,}`	1 000,12
`\form{~}{,}{~}`	1 000,12
`\form{}{,}{}`	1000,12
`\form{,}{.}{,}`	1,000.12
`\form{,}{$\cdot$}{,}`	1,000·12

3.3.2 The `\round` command

`\round` The `\round` command takes the *rounding number* as argument.

`\round{`⟨*rounding number*⟩`}`

It describes where to round the given money amount. Positive values round to integer digits, negative numbers round to fractional digits. Numbers lower than −9 suppress rounding.

rounding	*result*
`-10`	12 345,678 987 65
`-4`	12 345,679 0
`-3`	12 345,679
`-2`	12 345,68
`-1`	12 345,7
`0`	12 346
`1`	12 350
`2`	12 300
`3`	12 000
`4`	10 000
`5`	0

3.3.3 The `\zero` command

`\zero` The `\zero` command describes how to deal with zeros. It takes three arguments: the *overall zero* part, the *integer zero* part and the *fractional zero* part.

`\zero{`⟨*overall zero*⟩`}{`⟨*integer zero*⟩`}{`⟨*fractional zero*⟩`}`

emdash The following table shows some of the possible settings. The first entry is the predefined one, which should be used for running text. The last line shows the number format that is obtained in table mode with the default style emdash selected.

	`0.0`	`0.3`	`3.0`
`\zero{0}{0}{}`	0	0,30	3
`\zero{0}{0}{,00}`	0	0,30	3,00
`\zero{0}{}{}`	0	,30	3
`\zero{---\EURO@align{,---}}{---}{,---}`	—,—	—,30	3,—
`\zero{0}{---}{,---}`	0	—,30	3,—

The `\EURO@align` command will get replaced by `\rlap` if alignment was requested. You only have to state it in the first argument, if you want to align zero amounts in tables.

3.3.4 The `\plus` and `\minus` commands

`\plus` `\minus` The `\plus` and `\minus` commands allow to define a preamble/postamble for positive and negative numbers, respectively.

```
\plus{⟨positive preamble⟩}{⟨positive postamble⟩}
\minus{⟨negative preamble⟩}{⟨negative postamble⟩}
```

While `\plus` isn't used by default, `\minus` is defined to provide a preceding minus sign for negative numbers.

	positive numbers	negative numbers
`\minus{\(-\)}{}`	12 000	−12 000
`\minus{(}{)}`	0,79	(0,79)
`\minus{\color{red}(}{)}`	123,46	(123,46)
`\minus{\color{red}\(-\)}{}`	10	−10

3.4 Tables and alignment

To achieve proper alignment in tables there's an `\align` command that can be used in any of the different format strings no matter if it's a *main*, an *in*, *out*, or one of the *number* formats. This command is to be stated before the concerned `\in`, `\out`, or `\val` command, respectively. It lets a number's fractional part have no width, thus allowing to align such numbers properly in a flush-right table column `\table` (`r` column). The example below also shows how to use the `\table` keyword. It turns the table mode on for the current table. To turn it on for the whole document, use the table package option. If you find the '—' dashes too long, you endash can load the package with the endash option.

```
\newcommand*\ALNUM{%
  \EUROFORMAT{main}{\table\in}%     % no \out currency
  \EUROFORMAT{in}{\align\val}%      % no symbol/ISO code
  \EURO{EUR}}%

\begin{tabular}{r}
\textbf{costs}\\
\ALNUM{123456}\\
\ALNUM{1234.56}\\
\ALNUM{12.3456}\\
\ALNUM{0.123456}
\end{tabular}
```

costs
123 456,—
1 234,56
12,35
—,12

It may be necessary to put a `\hspace` after such an aligned number, especially when it is applied more than once. The second column would have to be `\llap`ped in this case.

```
\newcommand*\IEP{%
  \EUROFORMAT{main}{\align\in\hspace{1.8cm}\llap{\align\out}}%
  \EUROFORMAT{in}{\val}%
  \EUROFORMAT{out}{\val}%
  \EURO{IEP}}

\begin{tabular}{lr}
\bf Month& \bf Irish Pound/Euro\\
January&   \IEP{6070.83}\\
February&  \IEP{1200}\\
March&     \IEP{0.123}\\
April&     \IEP{441.0358}
\end{tabular}
```

Month	**Irish Pound/Euro**	
January	6 070,83	7 708,36
February	1 200	1 523,69
March	0,12	0,16
April	441,04	560

It would have been much simpler, though, not to combine two columns in one `\EURO` command, but to use two separate entries.

3.5 Dirty tricks

Of course, the format doesn't have to deal with the €, nor with monetary amounts at all. (The `EUR` currency in the following examples has no meaning, we could also have used any other currency, provided that no `\out` command is used within the *format string*.)

```
\newcommand*\USD[1]{{%
  \EUROFORMAT{main}{\in}%
  \EUROFORMAT{in}{\val\,\$}%
  \EUROFORMAT{EUR}{\form{,}{.}{,}\round{-2}}%
  \EURO{EUR}{#1}}}

\USD{-123450.6789}
```

−123,450.68 $

These examples are probably too complicated for the simple things they do. But they may serve as examples of how to use some of the customization commands.

```
\newcommand*\NUM[1]{{%
  \EUROFORMAT{main}{\in}%
  \EUROFORMAT{in}{\val}%
  \EUROFORMAT{EUR}{\form{,}{.}{,}\round{-10}}%
  \EURO{EUR}{#1}}}

\NUM{1000} red ants
```

1,000 red ants

3.6 The default settings

The following lines show how the `\EURO` command's default behavior could have been defined. The real settings, however, were defined on a lower programming level.

```
\EUROFORMAT{main}{\in\ (\out)}
\EUROFORMAT{in}{\val~\sym}
\EUROFORMAT{out}{\val~\sym}
\EUROFORMAT{all}{%
  \form{\,}{,}{\,}%
  \round{-2}%
  \zero{0}{0}{}%
  \plus{}{}%
  \minus{\(-\)}{}}
\EUROFORMAT{ITL}{%
  \round{0}}
```

3.7 The configuration file

If you want to change the predefined settings, create a file named '`euro.cfg`' and put it in a directory, where TEX can find it. This configuration file will then be loaded at the end of the `euro.sty` file, so you may redefine any settings or commands, select package options and even introduce new package options. But if you intend to give your documents to others, don't forget to give them the required configuration files, too! That's how such a file could look like:

```
% introduce the US-Dollar and the British Pound Sterling
\EUROADD{USD}{\$}{1.0891}
\EUROADD{GBP}{\textsterling}{0.6694}

% let Italian Lire be rounded to 10s
\EUROFORMAT{ITL}{\round{1}}
% and let their symbol be only 'L.'
\EUROSYM{ITL}{L.}

% declare a new package option 'bold',
% which typesets the source currency in boldface type
\DeclareOption{bold}{%
  \EUROFORMAT{main}{\textbf{\in} (\out)}}

% always select the 'endash' option
\ExecuteOptions{endash}
\endinput
```

The European Union & Patents on Software

Because this package is dedicated to the European Union's official currency, I feel the obligation – in the name of freedom and democracy, and as a citizen of the EU – to comment on today's decision on the "Computer Implemented Inventions Directive":

Today, 2005/03/07, unelected bureaucrats – the European Council of ministers – decided to ignore the objections and clear will of the elected European

Parliament, those of the Committee on Legal Affairs (JURI), the presidency of the European Parliament, of several national parliaments, and nothing less than the interests of the citizens of the European Union. Instead, they chose to fulfill the wishes of software industry giants (some of which have been convicted of abusing their monopoly power in several countries), and the wishes of bureaucrats from the European Patent Office, which has illegally granted 30.000 software patents already.

This is an incredible offense and disgusting abuse of power. It has yet to be investigated, which rôle corruption played in this plot, for which there seem to be strong signs. We are facing nothing less than an oligarchy of unelected bureaucrats who feel more obliged to maintaining the wealth of the richest man on earth, than the interests of the European citizens and the European economy. There is still hope that the European Parliament will reject the directive in the second reading. This appeal will remain in the documentation, though, as long as the European Parliament is treated as a fig leaf for undemocratic decisions. And should this directive become effective, and should I ever encounter conflicts with software patent claims in *any* of my software packages, the `euro` package will be the first that I will withdraw.

This is a sad day. Shame on the European Council! Shame on the European Commission!

References

[1] Duden, Volume 1: *Die Rechtschreibung.* Bibliographisches Institut, Mannheim–Wien–Zürich, 1986, 19th edition.

[2] European Council: *Council Regulation (EC) No. 2866/98 of 31 December 1998 on the conversion rates between the euro and the currencies of the Member States adopting the euro.* Official Journal L 359. 31 December 1998.

[3] European Council: *Council Regulation (EC) No. 1478/2000 of 19 June 2000 amending Regulation (EC) No. 2866/98 on the conversion rates between the euro and the currencies of the Member States adopting the euro.* Official Journal L 167. 7 July 2000.

[4] European Council: *Certain provisions related to the introduction of the Euro.* Council Regulation 1103/97, 17 June 1997.

[5] Knuth, Donald Ervin: *The TEXbook.* Addison–Wesley Publishing Company, Reading/Massachusetts, 1989, 16th edition.

[6] Tschichold, Jan: *Ausgewählte Aufsätze über Fragen der Gestalt des Buches und der Typographie.* Birkhäuser, Basel, 1987, 2nd edition.

4 The implementation

```
⟨*package⟩
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{euro}[2003/02/19 v1.1 euro  (mf)]
\RequirePackage{fp-basic}
\RequirePackage{fp-snap}
\FPmessagesfalse
```

4.1 The engine

\EURO@zero
\EURO@lzero
\EURO@rzero
\EURO@lminus
\EURO@rminus
\EURO@lplus
\EURO@rplus
\EURO@align
\EURO@lsep
\EURO@lsep
\EURO@point
\EURO@round

These macros are for internal use only. They partly predefine the format strings. Never change them here, but create a '`euro.cfg`' file, where you can redefine them without causing undesirable side effects.

```
\newcommand*\EURO@zero{0}
\newcommand*\EURO@lzero{0}
\newcommand*\EURO@rzero{}
\newcommand*\EURO@lminus{\(-\)}
\newcommand*\EURO@rminus{}
\newcommand*\EURO@lplus{}
\newcommand*\EURO@rplus{}
\newcommand*\EURO@align{}
\newcommand*\EURO@lsep{\,}
\newcommand*\EURO@rsep{\,}
\newcommand*\EURO@point{,}
\newcommand*\EURO@round{-2}
```

\EURO
\EURO@@
\EURO@scan

These macros manage the whole package. All the *format string* macros (also called *keywords*) are defined here locally. Note the beautiful `####`s! `;-)`

```
\newcommand*\EURO[1]{\bgroup\def\EURO@icurr{#1}\EURO@@}
\newcommand*\EURO@@[2][EUR]{\FPset\EURO@ival{#2}%
  \def\EURO@set##1##2##3##4##5{\FPset##5{##1}}%
  \csname EURO@@\EURO@icurr\endcsname\EURO@ifac
  \csname EURO@@#1\endcsname\EURO@ofac
  \FPdiv\EURO@oval\EURO@ival\EURO@ifac
  \FPmul\EURO@oval\EURO@oval\EURO@ofac
  \def\EURO@set##1##2##3##4##5{\def##5{##4}}%
  \EURO@@main\EURO@
  \EURO@@all\EURO@dflt
  \let\EURO@set\EURO@scan
  \def\in{{\csname EURO@@\EURO@icurr\endcsname\EURO@ival
    \def\EURO@set####1####2####3{}\EURO@@in}}%
  \def\out{{\csname EURO@@#1\endcsname\EURO@oval
    \def\EURO@set####1####2####3{}\EURO@@out}}%
  \def\align{\def\EURO@align####1{\rlap{####1}\let\EURO@align\relax}}%
  \let\table\EURO@table
  \def\form##1##2##3{\def\EURO@lsep{##1}\def\EURO@rsep{##3}%
    \def\EURO@point{##2}}%
  \def\round##1{\def\EURO@round{##1}}%
  \let\zero\EURO@setzero
  \def\minus##1##2{\def\EURO@lminus{##1}\def\EURO@rminus{##2}}%
  \def\plus##1##2{\def\EURO@lplus{##1}\def\EURO@rplus{##2}}%
  \EURO@\egroup}
\newcommand*\EURO@scan[5]{%
  \def\val{\FPifzero#5\EURO@zero\else\EURO@num#5\fi}%
  \def\iso{#2}\def\sym{#3}\def\EURO@form{\EURO@dflt#4}%
  \ifx\EURO@form\empty\let\EURO@form\EURO@dflt\fi
  \EURO@form}%
\newcommand*\EURO@setzero[3]{\def\EURO@zero{#1}%
  \def\EURO@lzero{#2}\def\EURO@rzero{#3}}%
```

\EUROSYM
\EUROFORMAT
\EUROADD

These macros replace the third/fourth parameter of the respective currency entry with another currency code/format string, or add a new currency entry.

```
\newcommand*\EUROSYM[2]{%
  \def\EURO@set##1##2##3##4{\toks@={\EURO@set{##1}{##2}{#2}{##4}}%
    \expandafter\edef\csname EURO@@#1\endcsname{\the\toks@}}%
  \csname EURO@@#1\endcsname}
\newcommand*\EUROFORMAT[2]{%
  \def\EURO@set##1##2##3##4{\toks@={\EURO@set{##1}{##2}{##3}{#2}}%
    \expandafter\edef\csname EURO@@#1\endcsname{\the\toks@}}%
  \csname EURO@@#1\endcsname}
\newcommand*\EUROADD[3]{%
  \expandafter\def\csname EURO@@#1\endcsname{\EURO@set{#3}{#1}{#2}{}}}
```

\EURO@num
\EURO@split

Takes a macro containing a number, passes the integer part to \EURO@grp and outputs everything else, cares about rounding and such…

```
\newcommand*\EURO@num[1]{{%
  \FPifneg#1%
    \EURO@lminus\let\EURO@post\EURO@rminus\FPneg#1#1%
  \else
    \EURO@lplus\let\EURO@post\EURO@rplus
  \fi
  \EURO@rround#1%
  \let\EURO@aux#1%
  \FPtrunc#1#10%
  \FPifzero#1\EURO@lzero\else\EURO@grp#1\fi
  \EURO@align{%
    \FPsub\EURO@aux\EURO@aux#1%
    \FPifzero\EURO@aux
      \ifnum\EURO@round<0 \EURO@rzero\fi
    \else
      \ifnum\EURO@round<0
        \ifnum\EURO@round<-9
          \FPclip\EURO@aux\EURO@aux
        \else
          \FPtrunc\EURO@aux\EURO@aux{-\EURO@round}%
        \fi
        \EURO@point
        \expandafter\EURO@split\EURO@aux\relax\relax\relax\EURO@
      \fi
```

```
     \fi
     \EURO@post}}%
  \let\EURO@align\relax}
\newcommand*\EURO@split{}
\def\EURO@split0.#1#2#3#4\EURO@{#1#2#3%
  \EURO@threedig#4\relax\relax\relax\EURO@}
\newcommand*\EURO@threedig[3]{%
  \if#1\relax
    \let\EURO@\relax
  \else
    \EURO@rsep#1#2#3\let\EURO@\EURO@threedig
  \fi\EURO@}
```

`\EURO@rround` `\EURO@rfac` Takes a macro containing a number, returns it rounded. `\EURO@rfac` calculates the rounding factor 10^r for positive r (*rounding number*).

```
\newcommand*\EURO@rround[1]{%
  \count@=\EURO@round\relax
  \ifnum\count@<1%
    \count@-\count@
    \FPround#1#1\count@
  \else
    \FPset\EURO@aux1%
    \EURO@rfac\EURO@aux\count@
    \FPdiv#1#1\EURO@aux
    \FPround#1#10%
    \FPmul#1#1\EURO@aux
  \fi}
\newcommand*\EURO@rfac[2]{%
  \ifnum#2>0 \advance#2-1 \EURO@rfac#1#2\FPmul#1#1{10}\fi}
```

`\EURO@grp` Let's do some recursion—it's fun! (This macro does the three-digit grouping.)

```
\newcommand*\EURO@grp[1]{\bgroup%
  \FPtrunc#1#10%
  \FPiflt#1{1000}%
    #1%
  \else
    \let\EURO@aux=#1%
    \FPdiv#1#1{1000}%
    \FPtrunc#1#10%
    \EURO@grp#1%
    \FPmul#1#1{1000}%
    \FPsub\EURO@aux\EURO@aux#1%
    \FPtrunc\EURO@aux\EURO@aux0%
    \count0=\EURO@aux\relax
    \EURO@lsep
    \ifnum\count0<100 0\ifnum\count0<10 0\fi\fi\EURO@aux
  \fi\egroup}
```

4.2 The facts

\EURO@@ATS \EURO@@BEF \EURO@@DEM \EURO@@ESP \EURO@@FIM \EURO@@FRF \EURO@@GRD \EURO@@IEP \EURO@@ITL \EURO@@LUF \EURO@@NLG \EURO@@PTE \EURO@@EUR

These values have been irrevocably fixed by the European Council on suggestion by the European Commission and been published in the *Council Regulation (EC) No. 2866/98* [2] and *No. 1478/2000* [3] (taken from `http://europa.eu.int/`).

```
\newcommand*\EURO@@ATS{\EURO@set{13.7603}{ATS}{\"oS}{}}
\newcommand*\EURO@@BEF{\EURO@set{40.3399}{BEF}{BEF}{}}
\newcommand*\EURO@@DEM{\EURO@set{1.95583}{DEM}{DM}{}}
\newcommand*\EURO@@ESP{\EURO@set{166.386}{ESP}{Pt.}{}}
\newcommand*\EURO@@FIM{\EURO@set{5.94573}{FIM}{FIM}{}}
\newcommand*\EURO@@FRF{\EURO@set{6.55957}{FRF}{FRF}{}}
\newcommand*\EURO@@GRD{\EURO@set{340.750}{GRD}{GRD}{}}
\newcommand*\EURO@@IEP{\EURO@set{.787564}{IEP}{IEP}{}}
\newcommand*\EURO@@ITL{\EURO@set{1936.27}{ITL}{Lit.}{}}
\newcommand*\EURO@@LUF{\EURO@set{40.3399}{LUF}{LUF}{}}
\newcommand*\EURO@@NLG{\EURO@set{2.20371}{NLG}{NLG}{}}
\newcommand*\EURO@@PTE{\EURO@set{200.482}{PTE}{PTE}{}}
\newcommand*\EURO@@EUR{\EURO@set{1.00000}{EUR}{Euro}{}}
```

\EURO@@main \EURO@@in \EURO@@out \EURO@@all

Define the generic format macros and 'fill' them with some default settings. Define the number format for the *Italian Lira* and the package options, load the optional '`euro.cfg`' file and, finally: Exit.

```
\newcommand*\EURO@@main{\EURO@set0{}{}{}}
\newcommand*\EURO@@in{\EURO@set0{}{}{}}
\newcommand*\EURO@@out{\EURO@set0{}{}{}}
\newcommand*\EURO@@all{\EURO@set0{}{}{}}
\EUROFORMAT{main}{\in\ (\out)}
\EUROFORMAT{in}{\val~\sym}
\EUROFORMAT{out}{\val~\sym}
\EUROFORMAT{ITL}{\round0}
\DeclareOption{eco}%
  {\EUROFORMAT{in}{\iso~\val}%
   \EUROFORMAT{out}{\iso~\val}}
\newcommand*\EURO@emdash{\EURO@setzero
  {0}{---}{\EURO@point---}}
\newcommand*\EURO@endash{\EURO@setzero
  {0}{--\kern.1em}{\EURO@point\kern.1em--}}
\newcommand*\EURO@zeros{\EURO@setzero
  {0\EURO@align{\EURO@point00}}{0}{\EURO@point00}}
\DeclareOption{emdash}{\let\EURO@table\EURO@emdash}
\DeclareOption{endash}{\let\EURO@table\EURO@endash}
\DeclareOption{zeros}{\let\EURO@table\EURO@zeros}
\DeclareOption{table}{\EURO@table}
\DeclareOption{dots}{\let\EURO@lsep.\let\EURO@rsep.}
\ExecuteOptions{emdash}
\InputIfFileExists{euro.cfg}%
  {\PackageInfo{euro}{Local config file euro.cfg used}}{}
\ProcessOptions
\endinput
```

```
⟨/package⟩
```