

The didec package

Manual for version 1.2.1 (2026/06/01)

Thomas F. Sturm¹

<https://www.ctan.org/pkg/didec>

<https://github.com/T-F-S/didec>

Abstract

The `didec` package provides fixed-point arithmetic with two decimal places (“*di-decimal*”), as typically used in financial transactions for many currencies. Its primary intended use is (personal) bookkeeping.

Contents

1	Quick start	2
2	Didec kernels and didec expressions	4
3	Creating didec variables	7
4	Setting didec variables	7
5	Using didecs	11
6	Option settings	14
7	Didec conditionals	17
8	Viewing didecs	19
	Index	20

¹Prof. Dr. Dr. Thomas F. Sturm, Institut für Mathematik und Informatik, University of the Bundeswehr Munich, D-85577 Neubiberg, Germany; email: thomas.sturm@unibw.de

1 Quick start

For readers who prefer a brief summary: this package provides fixed-point arithmetic with two decimal places. It can be used in any context where exactly two decimal places are required or sufficient; its primary application, however, is (personal) bookkeeping.

Suppose John wants to keep track of his money. With

```
\didecnew{John}
```

a so-called *didec variable* is created to store currency values.

Now, lets fill in some money:

```
\didecset{John}{1000}
```

How much money does John have now?

```
\didecuse{John}
```

1.000,00 €

The amount is given in euros and formatted according to German conventions. Of course, this can be adapted to your liking, see `\didecsetup`^{P.14}. But now, let us spend some money.

```
\didecsub{John}{19.75}
\didecuse{John}
```

980,25 €

Preferably, the package functions are used inside convenient user commands. You can choose between L^AT_EX₂ε and L³ programming layer functions.

For this quick start, we make some convenience commands for John. `\transaction` shall be one purchase made in cash and `\transaction*` one made by credit card.

```
%\usepackage{booktabs}
\didecsetup{english,
  currency          = {\pounds}{},
  currency-negative = {-\pounds}{},
}

%\didecnew{John}
\didecnew{cash}
\didecnew{credit}
\didecnew{transaction}

% Let's keep record in a table
\NewDocumentEnvironment{householdbook}{}{%
  \begin{center}
    \begin{tabular}{lp{9cm}rr}\toprule%
      & Transaction & Expenses & Budget\\ \midrule
    \end{tabular}%
    \midrule%
    \multicolumn{3}{r}{Cash} & \dideccoluse{cash} \\ %
    \multicolumn{3}{r}{Credit card} & \dideccoluse{credit} \\ \bottomrule%
  \end{tabular}%
  \end{center}
```

```

}

% One transaction
\NewDocumentCommand \transaction{ s m m }
{
  \didecset{transaction}{#3}%
  \didecsub{John}{transaction}%
  \IfBooleanTF {#1}%
    {\didecsub{credit}{transaction}CC}%
    {\didecsub{cash}{transaction}}%
  & #2
  & \dideccolinvuse{transaction}
  & \dideccoluse{John}\\%
}

% Ready to start our tiny accountancy

\didecset{John}{1000}      % John's money
\didecsetequal{cash}{John} % in cash
\didecset{credit}{0}       % blank credit card

\begin{householdbook}
  \transaction{Coffee break with snack}{19.75}
  \transaction*{Refuel}{62.87}
  \transaction{Gift from Aunt Mary for helping her}{-30}
  \transaction{Parking meter}{4.50}
  \transaction*{Shopping for weekend}{147.23}
  \transaction*{Fancy thing on the Internet}{270}
  \transaction*{Cinema}{17.70}
\end{householdbook}

```

Transaction		Expenses	Budget
	Coffee break with snack	£19.75	£980.25
CC	Refuel	£62.87	£917.38
	Gift from Aunt Mary for helping her	-£30.00	£947.38
	Parking meter	£4.50	£942.88
CC	Shopping for weekend	£147.23	£795.65
CC	Fancy thing on the Internet	£270.00	£525.65
CC	Cinema	£17.70	£507.95
		Cash	£1,005.75
		Credit card	-£497.80

2 Didec kernels and didec expressions

All calculations are done on cent basis as integer operations, but all displayed figures have two decimal places which give the name for the package (*di-decimal*).

The package provides two numerical kernels which can be selected mutually by package options `didec/int` or `didec/fp`. The `didec/int` kernel is faster, but provides a smaller number range, while the `didec/fp` kernel is slower with a larger number range. For ordinary people like you and me, the `didec/int` kernel will suffice to do all personal financial calculations. Upgrading later from `didec/int` to `didec/fp` is a matter of just switching the package option setting.

didec/int (no value, initially set)

Selects the $\langle int \rangle$ (integer) based numerical kernel with up to 9.33² significant figures and fast computation. The number range for valid figures n is:

$$-21\,474\,836.47 \leq n \leq 21\,474\,836.47$$

```
\usepackage[int]{didec}
```

Using figures outside the valid range will result in L^AT_EX errors complaining about too large numbers.

didec/fp (no value, initially unset)

Selects the $\langle fp \rangle$ (floating-point) based numerical kernel with up to 16 significant figures and somewhat slower computation. The number range for valid figures n is:

$$-99\,999\,999\,999\,999.99 \leq n \leq 99\,999\,999\,999\,999.99$$

```
\usepackage[fp]{didec}
```

Using figures outside the valid range will result in *silent calculation errors*, because L^AT_EX3 $\langle fp \rangle$ can use much larger numbers but is restricted to 16 significant figures.

In the following, a $\langle didec\ expr \rangle$ (didec expression) denotes one of the following:

- a number in floating-point notation, e.g. 123.45
- a number in floating-comma notation, e.g. 123,45
- a $\langle didec\ var \rangle$ (didec variable), e.g. `expenses`.

Note that the notation $\langle didec\ expr \rangle$ is inspired by $\langle int\ expr \rangle$ and $\langle fp\ expr \rangle$ from L^AT_EX3 but is in comparison very restricted and allows only the three choices above. A $\langle didec\ expr \rangle$ is always expanded and spaces are trimmed.

Many provided commands or functions of the package come in three flavors, for example:

- `\didecadd`^{P.9}: This is a user command where arguments are space trimmed and some check on variable existence is done. Not existing variables are reported by speaking error messages (not in all cases!).
- `\didec_gadd_check:nn`^{P.9}: This a programming layer function with no space trimming for arguments, but some check on variable existence is done. Not existing variables are reported by speaking error messages (not in all cases!).
- `\didec_gadd:nn`^{P.9}: This a programming layer function with no space trimming for arguments and no check on variable existence. Not existing variables give strange errors. This is the fastest function and base of the others above.

²joke for the mathematicians

The following tables compare the computation time for selected functions of the package for the two numerical kernels. Time values will differ on other computers and also depend on selected values for the examples calculations. Nevertheless, you get an impression of the differences.

kernel: int, engine: pdftex		kernel: fp, engine: pdftex	
<code>\didec_gset:nn</code>	7 μ s	<code>\didec_gset:nn</code>	36 μ s
<code>\didec_gset_check:nn</code>	10 μ s	<code>\didec_gset_check:nn</code>	39 μ s
<code>\didecset</code>	11 μ s	<code>\didecset</code>	40 μ s
<code>\didec_gset_eq:nn</code>	1 μ s	<code>\didec_gset_eq:nn</code>	1 μ s
<code>\didec_gset_eq_check:nn</code>	6 μ s	<code>\didec_gset_eq_check:nn</code>	6 μ s
<code>\didecsetequal</code>	8 μ s	<code>\didecsetequal</code>	8 μ s
<code>\didec_gset_fp:nn</code>	240 μ s	<code>\didec_gset_fp:nn</code>	260 μ s
<code>\didec_gset_fp_check:nn</code>	246 μ s	<code>\didec_gset_fp_check:nn</code>	262 μ s
<code>\didecsetfp</code>	246 μ s	<code>\didecsetfp</code>	266 μ s
<code>\didec_gadd:nn</code>	6 μ s	<code>\didec_gadd:nn</code>	59 μ s
<code>\didec_gadd_check:nn</code>	8 μ s	<code>\didec_gadd_check:nn</code>	61 μ s
<code>\didecadd</code>	27 μ s	<code>\didecadd</code>	84 μ s
<code>\didec_gadd_to:nnn</code>	11 μ s	<code>\didec_gadd_to:nnn</code>	79 μ s
<code>\didec_gadd_to_check:nnn</code>	14 μ s	<code>\didec_gadd_to_check:nnn</code>	83 μ s
<code>\didecadd</code>	51 μ s	<code>\didecadd</code>	123 μ s
<code>\didec_gmul_fp:nn</code>	246 μ s	<code>\didec_gmul_fp:nn</code>	287 μ s
<code>\didec_gmul_fp_check:nn</code>	249 μ s	<code>\didec_gmul_fp_check:nn</code>	288 μ s
<code>\didecmulfp</code>	272 μ s	<code>\didecmulfp</code>	318 μ s
<code>\didec_if_positive:nTF</code>	1 μ s	<code>\didec_if_positive:nTF</code>	15 μ s
<code>\didecifpositive</code>	2 μ s	<code>\didecifpositive</code>	17 μ s
<code>\didec_compare:nNnTF</code>	9 μ s	<code>\didec_compare:nNnTF</code>	77 μ s
<code>\didecifgreaterthan</code>	10 μ s	<code>\didecifgreaterthan</code>	77 μ s
<code>\didec_to_fp:n</code>	13 μ s	<code>\didec_to_fp:n</code>	26 μ s
<code>\didec_to_fp_check:n</code>	15 μ s	<code>\didec_to_fp_check:n</code>	29 μ s
<code>\didectofp</code>	16 μ s	<code>\didectofp</code>	30 μ s
<code>\didec_use:n</code>	27 μ s	<code>\didec_use:n</code>	41 μ s
<code>\didec_use_check:n</code>	29 μ s	<code>\didec_use_check:n</code>	44 μ s
<code>\didecuse</code>	49 μ s	<code>\didecuse</code>	65 μ s
<code>\didecformat</code>	55 μ s	<code>\didecformat</code>	107 μ s
<code>\didec_color_use:n</code>	71 μ s	<code>\didec_color_use:n</code>	87 μ s
<code>\didec_color_use_check:n</code>	73 μ s	<code>\didec_color_use_check:n</code>	91 μ s
<code>\dideccoluse</code>	95 μ s	<code>\dideccoluse</code>	114 μ s
<code>\dideccolformat</code>	103 μ s	<code>\dideccolformat</code>	157 μ s

If needed, the selected kernel can be questioned by the following:

`\c_didec_kernel_str`

The current kernel is given as a lower case string: one of `int` or `fp`.

```
\ExplSyntaxOn
\c_didec_kernel_str
\ExplSyntaxOff
```

int

```
\didec_if_kernel_int_p:
\didec_if_kernel_int:T {\true code}
\didec_if_kernel_int:TF {\true code}{\false code}
\didec_if_kernel_fp_p:
\didec_if_kernel_fp:T {\true code}
\didec_if_kernel_fp:TF {\true code}{\false code}
```

Conditionals which allow kernel-specific code to be used. The names follow naturally from those of the kernels.

```
\ExplSyntaxOn
\didec_if_kernel_int:T { Integer~kernel~used. }\par
Floating~point~kernel
\didec_if_kernel_fp:TF { ~used. }{ ~not~used. }
\ExplSyntaxOff
```

Integer kernel used.
Floating point kernel not used.

3 Creating didec variables

```
\didecnew{<didec var>}  
\didec_new:n{<didec var>}
```

Creates a new $\langle didec var \rangle$ or raises an error if the name is already taken. `\didecnew` trims spaces while `\didec_new:n` does not.

```
\didecnew{konto}  
\didecset{konto}{99.75}  
\didecuse{konto}
```

99,75 €

4 Setting didec variables

```
\didecset{<didec var>}{<didec expr>}  
\didec_gset:nn{<didec var>}{<didec expr>}  
\didec_gset_check:nn{<didec var>}{<didec expr>}
```

Sets $\langle didec var \rangle$ to the value of $\langle didec expr \rangle$ which can be

- a number in floating-point notation,
- a number in floating-comma notation,
- another $\langle didec var \rangle$.

`\didecset` trims spaces and performs an existence check for $\langle didec var \rangle$.

`\didec_gset_check:nn` performs an existence check for $\langle didec var \rangle$. Decimals places 3 and beyond are cut not rounded. If rounding is an issue, use `\didecsetfp`^{P.8} instead.

```
\didecset{A}{1234.56}  
\didecuse{A}  
  
\didecset{A}{2345,6789}  
\didecuse{A}  
  
\didecset{B}{-3500}  
\didecset{A}{B}  
\didecuse{A}
```

1.234,56 €
2.345,67 €
-3.500,00 €

```
\didecsetequal{<didec var1>}{<didec var2>}  
\didec_gset_eq:nn{<didec var1>}{<didec var2>}  
\didec_gset_eq_check:nn{<didec var1>}{<didec var2>}
```

Sets the $\langle didec var_1 \rangle$ to the current value of $\langle didec var_2 \rangle$

```
\didecset{A}{1234.56}  
\didecsetequal{B}{A}  
\didecuse{A}  
\didecuse{B}
```

1.234,56 € 1.234,56 €

```

\didecsetnegative{<didec var>}{<didec expr>}
\didec_gset_negative:nn{<didec var>}{<didec expr>}
\didec_gset_negative_check:nn{<didec var>}{<didec expr>}

```

Sets $\langle didec var \rangle$ to the negated (opposite) value of $\langle didec expr \rangle$. `\didecsetnegative` trims spaces and performs an existence check for $\langle didec var \rangle$.

```

\didecsetnegative{A}{1234.56}
\didecuse{A}

\didecsetnegative{A}{-42.55}
\didecuse{A}

\didecset{B}{-3500}
\didecsetnegative{A}{B}
\didecuse{A}

```

-1.234,56 €
42,55 €
3.500,00 €

```

\didecsetfp{<didec var>}{<fp expr>}
\didec_gset_fp:nn{<didec var>}{<fp expr>}
\didec_gset_fp_check:nn{<didec var>}{<fp expr>}

```

Sets $\langle didec var \rangle$ to the value of $\langle fp expr \rangle$ which can be any L^AT_EX3 floating-point expression. `\didecsetfp` trims spaces and performs an existence check for $\langle didec var \rangle$. Other `didec` variables can be used inside $\langle fp expr \rangle$ if guarded with `\didectofp`^{P.11}. The result is rounded to 2 decimal places according to the rounding rule set by `didec/rounding`^{P.16}. `\didec_gset_fp_check:nn` performs an existence check for $\langle didec var \rangle$.

```

\didecsetfp{A}{2345.6789}
\didecuse{A}

\didecsetfp{A}{ ln( 12345678 ) }
\didecuse{A}

\didecset{A}{123456,78}
\didecsetfp{B}{ \didectofp{A} * 2.35 / 100 }
2.35\% of \didecuse{A} are \didecuse{B}.

```

2.345,68 €
16,33 €
2.35% of 123.456,78 € are 2.901,23 €.

```

\didecadd[⟨didec var⟩]{⟨didec expr1⟩}{⟨didec expr2⟩}
\didec_gadd:nn{⟨didec var⟩}{⟨didec expr⟩}
\didec_gadd_check:nn{⟨didec var⟩}{⟨didec expr⟩}
\didec_gadd_to:nnn{⟨didec var⟩}{⟨didec expr1⟩}{⟨didec expr2⟩}
\didec_gadd_to_check:nnn{⟨didec var⟩}{⟨didec expr1⟩}{⟨didec expr2⟩}

```

- Adds the result of computing the $\langle didec expr \rangle$ to the $\langle didec var \rangle$
- or sets $\langle didec var \rangle$ to the sum of $\langle didec expr_1 \rangle$ and $\langle didec expr_2 \rangle$.
- For `\didecadd`, if the optional $\langle didec var \rangle$ is not available, the sum is stored into $\langle didec expr_1 \rangle$ which has to be a $\langle didec var \rangle$ in this case.

```

\didecset{A}{123}
\didecset{B}{5,88}
\didecadd{A}{B}
\didecuse{A}

\didecadd[A]{B}{1000}
\didecuse{A}

\didecadd{A}{-2750}
\didecuse{A}

```

128,88 €
 1.005,88 €
 -1.744,12 €

```

\didecsub[⟨didec var⟩]{⟨didec expr1⟩}{⟨didec expr2⟩}
\didec_gsub:nn{⟨didec var⟩}{⟨didec expr⟩}
\didec_gsub_check:nn{⟨didec var⟩}{⟨didec expr⟩}
\didec_gsub_to:nnn{⟨didec var⟩}{⟨didec expr1⟩}{⟨didec expr2⟩}
\didec_gsub_to_check:nnn{⟨didec var⟩}{⟨didec expr1⟩}{⟨didec expr2⟩}

```

- Subtracts the result of computing the $\langle didec expr \rangle$ to the $\langle didec var \rangle$
- or sets $\langle didec var \rangle$ to the difference of $\langle didec expr_1 \rangle$ and $\langle didec expr_2 \rangle$.
- For `\didecsub`, if the optional $\langle didec var \rangle$ is not available, the difference is stored into $\langle didec expr_1 \rangle$ which has to be a $\langle didec var \rangle$ in this case.

```

\didecset{A}{123}
\didecset{B}{5,88}
\didecsub{A}{B}
\didecuse{A}

\didecsub[A]{B}{1000}
\didecuse{A}

\didecsub{A}{-2750}
\didecuse{A}

```

117,12 €
 -994,12 €
 1.755,88 €

```

\didecmulfp[⟨didec var2⟩]{⟨didec var⟩}{⟨fp expr⟩}
\didec_gmul_fp:nn{⟨didec var⟩}{⟨fp expr⟩}
\didec_gmul_fp_check:nn{⟨didec var⟩}{⟨fp expr⟩}
\didec_gmul_fp_to:nnn{⟨didec var2⟩}{⟨didec var⟩}{⟨fp expr⟩}
\didec_gmul_fp_to_check:nnn{⟨didec var2⟩}{⟨didec var⟩}{⟨fp expr⟩}

```

- Multiplies $\langle didec var \rangle$ with the result of computing the $\langle fp expr \rangle$
- and sets $\langle didec var \rangle$ or respectively $\langle didec var_2 \rangle$ to the result.
- The result is rounded to 2 decimal places according to the rounding rule set by `didec/rounding`^{→ P. 16}.

```

\didecset{A}{123}
\didecmulfp{A}{0.9675}
\didecuse{A}

\didecmulfp[B]{A}{ln(42)}
\didecuse{B}

```

119,00 €
444,78 €

```

\didecdivfp[⟨didec var2⟩]{⟨didec var⟩}{⟨fp expr⟩}
\didec_gdiv_fp:nn{⟨didec var⟩}{⟨fp expr⟩}
\didec_gdiv_fp_check:nn{⟨didec var⟩}{⟨fp expr⟩}
\didec_gdiv_fp_to:nnn{⟨didec var2⟩}{⟨didec var⟩}{⟨fp expr⟩}
\didec_gdiv_fp_to_check:nnn{⟨didec var2⟩}{⟨didec var⟩}{⟨fp expr⟩}

```

- Divides $\langle didec var \rangle$ by the result of computing the $\langle fp expr \rangle$
- and sets $\langle didec var \rangle$ or respectively $\langle didec var_2 \rangle$ to the result.
- The result is rounded to 2 decimal places according to the rounding rule set by `didec/rounding`^{→ P. 16}.

```

\didecset{A}{123}
\didecdivfp{A}{0.9675}
\didecuse{A}

\didecdivfp[B]{A}{ln(42)}
\didecuse{B}

```

127,13 €
34,01 €

```

\didecsetsum[⟨didec var⟩]{⟨sum of didec exp⟩}

```

Sets $\langle didec var \rangle$ to the result of computing the given $\langle sum of didec exp \rangle$.
Here, $\langle sum of didec exp \rangle = \langle didec exp_1 \rangle + \langle didec exp_2 \rangle + \dots + \langle didec exp_n \rangle$

```

\didecset{A}{123}
\didecset{B}{-32.15}
\didecsetsum{A}{ A + B + -22.5 }
\didecuse{A}

```

68,35 €

5 Using didecs

```
\didectoint{\<didec var>}  
\didec_to_int:n{\<didec var>}  
\didec_to_int_check:n{\<didec var>}
```

Expresses the $\langle didec\ var \rangle$ as Cent integer value, i.e. 100 times the value. All functions are expandable.

```
\didecset{A}{27123.45}  
\didectoint{A}  
  
\didecset{A}{-17}  
\didectoint{A}
```

```
2712345  
-1700
```

```
\didectofp{\<didec var>}  
\didec_to_fp:n{\<didec var>}  
\didec_to_fp_check:n{\<didec var>}
```

Expresses the $\langle didec\ var \rangle$ as floating-point value. All functions are expandable.

```
\didecset{A}{27123.45}  
\didectofp{A}  
  
\didecset{A}{-17}  
\didectofp{A}
```

```
27123.45  
-17.00
```

```
\didectofc{\<didec var>}  
\didec_to_fc:n{\<didec var>}  
\didec_to_fc_check:n{\<didec var>}
```

Expresses the $\langle didec\ var \rangle$ as floating-comma value. All functions are expandable.

```
\didecset{A}{27123.45}  
\didectofc{A}  
  
\didecset{A}{-17}  
\didectofc{A}
```

```
27123,45  
-17,00
```

```

\didecuse[⟨key list⟩]{⟨didec var⟩}
\didec_use:n{⟨didec var⟩}
\didec_use_check:n{⟨didec var⟩}

```

Expresses the $\langle didec var \rangle$ as formatted value. With $\backslash didecsetup^{\rightarrow P.14}$, the standard format can be set. This standard format can be overwritten by $\langle key list \rangle$.

```

\didecset{A}{123456.78}
\didecuse{A}

\didecuse[ english, currency={\pounds}{} ]{A}

```

123.456,78 €
£123,456.78

```

\dideccoluse[⟨key list⟩]{⟨didec var⟩}
\didec_color_use:n{⟨didec var⟩}
\didec_color_use_check:n{⟨didec var⟩}

```

Expresses the $\langle didec var \rangle$ as colorized formatted value. With $\backslash didecsetup^{\rightarrow P.14}$, the standard format can be set. This standard format can be overwritten by $\langle key list \rangle$.

```

\didecset{A}{123456.78}
\dideccoluse{A}

\didecset{A}{-125}
\dideccoluse{A}

\dideccoluse[color-negative=didec-blue]{A}

```

123.456,78 €
-125,00 €
-125,00 €

```

\dideccolinvuse[⟨key list⟩]{⟨didec var⟩}
\didec_color_inverse_use:n{⟨didec var⟩}
\didec_color_inverse_use_check:n{⟨didec var⟩}

```

Expresses the $\langle didec var \rangle$ as colorized formatted value. The coloring is switched between positive and negative. The standard coloring and format can be overwritten by $\langle key list \rangle$.

```

\didecset{A}{123456.78}
\dideccolinvuse{A}

\didecset{A}{-125}
\dideccolinvuse{A}

```

123.456,78 €
-125,00 €

```

\didformat[⟨key list⟩]{⟨didec expr⟩}
\dideccolformat[⟨key list⟩]{⟨didec expr⟩}
\dideccolinvformat[⟨key list⟩]{⟨didec expr⟩}

```

Like $\backslash\text{didecuse}^{\rightarrow P.12}$, $\backslash\text{dideccoluse}^{\rightarrow P.12}$, $\backslash\text{dideccolinvuse}^{\rightarrow P.12}$, but accepts a $\langle didec \text{ expr} \rangle$ instead of a $\langle didec \text{ var} \rangle$. If the $\langle didec \text{ expr} \rangle$ is a $\langle didec \text{ var} \rangle$, $\backslash\text{didecuse}^{\rightarrow P.12}$, $\backslash\text{dideccoluse}^{\rightarrow P.12}$, $\backslash\text{dideccolinvuse}^{\rightarrow P.12}$ are more efficient.

```

\didformat{123456.78}
\didformat{A}
\didformat[english]{123456.78}
\dideccolformat{123456.78}
\dideccolinvformat{123456.78}

```

```

123.456,78 €
-125,00 €
123,456.78 €
123.456,78 €
123.456,78 €

```

```

\didcwrite{⟨didec var⟩}{⟨stream⟩}
\didc_write:nn{⟨didec var⟩}{⟨stream⟩}
\didc_write_check:nn{⟨didec var⟩}{⟨stream⟩}

```

Writes $\backslash\text{didecset}^{\rightarrow P.7}\{\langle didec \text{ var} \rangle\}\{\langle \text{current value} \rangle\}$ to the given already opened output $\langle \text{stream} \rangle$.

```

\didcwrite{A}{output}
% writes to output:
% \didecset{A}{VALUE}

```

6 Option settings

`\didecsetup``{⟨key list⟩}`

Sets all keys of the given *⟨key list⟩*. See the following for available settings.

```
\didecsetup{
  currency          = {\pounds}{},
  decimal-separator = {.,},
  grouping-separator = {,},
}
\didecset{A}{123456.78}
\didecuse{A}
```

£123,456.78

U 2026-02-20

`didec/currency``={⟨prefix⟩}{⟨postfix⟩}` (initially `{ }{\;€}`)

Sets some *⟨prefix⟩* and *⟨postfix⟩* to denote the currency of the didec variable. This also sets `didec/currency-negative{⟨prefix⟩}{⟨postfix⟩}`

```
\didecset{A}{123456.78}
\didecuse[ currency = {\pounds}{} ]{A} \par
\didecuse[ currency = {}{\:Gulden} ]{A}
```

£123.456,78
123.456,78 Gulden

U 2026-02-20

`didec/currency-negative``={⟨prefix⟩}{⟨postfix⟩}` (initially `{-}{\;€}`)

Sets some *⟨prefix⟩* and *⟨postfix⟩* to denote the currency of the didec variable, if the resulting value is negative. Otherwise, the settings of `didec/currency` are used. Note that you need to set a minus sign `-` explicitly, if you want to see it. Also note that setting `didec/currency` overwrites values given by `didec/currency-negative`.

```
\didecset{A}{-123456.78}
\didecuse{A}\par
\didecuse[ currency-negative = {\$-}{\;€} ]{A} \par
\didecuse[ currency-negative = {()}{\;€} ]{A} \par
\didecuse[ currency-negative = {(){} }\;€ }{A} \par
\didecuse[ currency-negative = {-€}{} ]{A} \par
```

-123.456,78 €
-123.456,78 €
(123.456,78 €)
(123.456,78) €
-€123.456,78

didec/decimal-separator= $\langle separator \rangle$ (initially ,)

Sets some $\langle separator \rangle$ as decimal separator.

```
\didecset{A}{123456.78}
\didecuse[ decimal-separator={\#} ]{A} \par

123.456#78 €
```

didec/grouping-separator= $\langle separator \rangle$ (initially .)

Sets some $\langle separator \rangle$ as grouping separator.

```
\didecset{A}{123456.78}
\didecuse[ grouping-separator={'} ]{A}

123'456,78 €
```

<u>U 2026-02-20</u>	didec/german	(style)
<u>U 2026-02-20</u>	didec/english	(style)
<u>U 2026-02-20</u>	didec/french	(style)
<u>U 2026-02-20</u>	didec/float	(style)

Styles to set some format preferences combined. Note that $\backslash\text{didectofp}^{\rightarrow P.11}$ is more efficient than $\backslash\text{didecuse}^{\rightarrow P.12}$ with style **didec/float**.

	german	english	french	float
decimal-separator	,	.	,	.
grouping-separator	.	,	\;	




```
\didecset{A}{12345678.90}
\didecuse[german]{A} \par
\didecuse[english]{A} \par
\didecuse[french]{A} \par
\didecuse[float]{A} \par

12.345.678,90 €
12,345,678.90 €
12 345 678,90 €
12345678.90 €
```

didec/color-positive= $\langle positive color \rangle$ (initially **didec-green**)

didec/color-negative= $\langle negative color \rangle$ (initially **didec-red**)

Sets $\langle positive color \rangle$ to denote positive (and zero) values and $\langle negative color \rangle$ to denote negative values. Any valid **13color** $\langle color expression \rangle$ can be used. The package defines additional colors

-  **didec-green**
-  **didec-red**
-  **didec-blue**

```
\didecset{A}{123456.78}
\dideccoluse[ color-positive = magenta ]{A}

123.456,78 €
```

didec/rounding={ $\langle rule \rangle$ } (initially **half-away-from-zero**)

Sets the rounding $\langle rule \rangle$ for floating-point expressions used inside `\didecsetfp`^{→P.8}, `\didecmulfp`^{→P.10}, `\didecdivfp`^{→P.10}, and friends. There are two two choices (each providing several synonyms):

- **half-away-from-zero**, **commercial**, **DIN-1333**:

Round to the nearest value; ties are rounded away from zero.

Common: “commercial rounding”, “Kaufmännisches Runden”

```
\didecsetup{rounding=half-away-from-zero}
\didecsetfp{A}{-2.015} \didecuse{A}\par
\didecsetfp{A}{-2.005} \didecuse{A}\par
\didecsetfp{A}{2.005} \didecuse{A}\par
\didecsetfp{A}{2.015} \didecuse{A}\par
```

-2,02 €
-2,01 €
2,01 €
2,02 €

- **half-to-even**, **mathematical**, **IEEE-754**:

Round to the nearest value; ties are rounded to the nearest even integer.

Common: “Banker’s rounding”, “Mathematisches Runden”

```
\didecsetup{rounding=half-to-even}
\didecsetfp{A}{-2.015} \didecuse{A}\par
\didecsetfp{A}{-2.005} \didecuse{A}\par
\didecsetfp{A}{2.005} \didecuse{A}\par
\didecsetfp{A}{2.015} \didecuse{A}\par
```

-2,02 €
-2,00 €
2,00 €
2,02 €

Note that **half-to-even** was the default (and only) setting before version 1.2.0. The default changed to **half-away-from-zero** with version 1.2.0.

7 Didec conditionals

```
\didecifpositive{<didec var>}{<true code>}{<false code>}
\didec_if_positive_p:n{<didec var>}
\didec_if_positive:nTF{<didec var>}{<true code>}{<false code>}
\didec_if_positive:nT{<didec var>}{<true code>}
\didec_if_positive:nF{<didec var>}{<false code>}
```

Evaluates the $\langle didec var \rangle$ and returns **true** or executes the $\langle true code \rangle$ if the value is positive, otherwise returns **false** or executes the $\langle false code \rangle$.

```
\didecset{A}{2799.50}
\didecuse{A} is \didecifpositive{A}{positive}{not positive}.

\didecset{B}{-584}
\didecuse{B} is \didecifpositive{B}{positive}{not positive}.

\didecset{A}{0}
\didecuse{A} is \didecifpositive{A}{positive}{not positive}.

2.799,50 € is positive.
-584,00 € is not positive.
0,00 € is not positive.
```

```
\didecifnegative{<didec var>}{<true code>}{<false code>}
\didec_if_negative_p:n{<didec var>}
\didec_if_negative:nTF{<didec var>}{<true code>}{<false code>}
\didec_if_negative:nT{<didec var>}{<true code>}
\didec_if_negative:nF{<didec var>}{<false code>}
```

Evaluates the $\langle didec var \rangle$ and returns **true** or executes the $\langle true code \rangle$ if the value is negative, otherwise returns **false** or executes the $\langle false code \rangle$.

```
\didecset{A}{2799.50}
\didecuse{A} is \didecifnegative{A}{negative}{not negative}.

\didecset{B}{-584}
\didecuse{B} is \didecifnegative{B}{negative}{not negative}.

\didecset{A}{0}
\didecuse{A} is \didecifnegative{A}{negative}{not negative}.

2.799,50 € is not negative.
-584,00 € is negative.
0,00 € is not negative.
```

```

\didеcifzero{<didеc var>}{<true code>}{<false code>}
\didеc_if_zero_p:n{<didеc var>}
\didеc_if_zero:nTF{<didеc var>}{<true code>}{<false code>}
\didеc_if_zero:nT{<didеc var>}{<true code>}
\didеc_if_zero:nF{<didеc var>}{<false code>}

```

Evaluates the $\langle didеc var \rangle$ and returns **true** or executes the $\langle true code \rangle$ if the value is zero, otherwise returns **false** or executes the $\langle false code \rangle$.

```

\didеcset{A}{2799.50}
\didеcuse{A} is \didеcifzero{A}{zero}{not zero}.

\didеcset{B}{-584}
\didеcuse{B} is \didеcifzero{B}{zero}{not zero}.

\didеcset{A}{0}
\didеcuse{A} is \didеcifzero{A}{zero}{not zero}.

```

2.799,50 € is not zero.
-584,00 € is not zero.
0,00 € is zero.

```

\didеciflowerthan{<didеc expr1>}{<didеc expr2>}{<true code>}{<false code>}
\didеcifequal{<didеc expr1>}{<didеc expr2>}{<true code>}{<false code>}
\didеcifgreaterthan{<didеc expr1>}{<didеc expr2>}{<true code>}{<false code>}
\didеc_compare_p:nNn{<didеc expr1>}{<relation>}{<didеc expr2>}
\didеc_compare:nNnTF{<didеc expr1>}{<relation>}{<didеc expr2>}{<true code>}{<false code>}
\didеc_compare:nNnT{<didеc expr1>}{<relation>}{<didеc expr2>}{<true code>}
\didеc_compare:nNnF{<didеc expr1>}{<relation>}{<didеc expr2>}{<false code>}

```

Compares the $\langle didеc expr_1 \rangle$ and the $\langle didеc expr_2 \rangle$, and returns **true** or executes the $\langle true code \rangle$ if the $\langle relation \rangle$ (given by function respectively) is obeyed, otherwise returns **false** or executes the $\langle false code \rangle$.

```

\didеcset{A}{2799.50}
\didеcset{B}{-584}
\didеcuse{A} is \didеciflowerthan{A}{B}{lower}{not lower} than \didеcuse{B}

\didеcuse{A} is \didеcifequal{A}{B}{equal}{not equal} to \didеcuse{B}

\didеcuse{A} is \didеcifequal{A}{A}{equal}{not equal} to \didеcuse{A}

\didеcuse{A} is \didеcifgreaterthan{A}{B}{greater}{not greater} than
↪ \didеcuse{B}

```

2.799,50 € is lower than -584,00 €
2.799,50 € is equal to -584,00 €
2.799,50 € is equal to 2.799,50 €
2.799,50 € is greater than -584,00 €

8 Viewing didecs

```
\didec_show:n{⟨didec var⟩}
```

Displays the content of $\langle didec\ var \rangle$ in the terminal.

```
\didecset{A}{2799.50}  
\ExplSyntaxOn  
\didec_show:n{A}  
\ExplSyntaxOff
```

Index

`\c_didec_kernel_str`, 6
`color-negative` key, 15
`color-positive` key, 15
Colors
 `didec-blue`, 15
 `didec-green`, 15
 `didec-red`, 15
Commands
 `\c_didec_kernel_str`, 6
 `\didec_color_inverse_use:n`, 12
 `\didec_color_inverse_use_check:n`, 12
 `\didec_color_use:n`, 12
 `\didec_color_use_check:n`, 12
 `\didec_compare:nNnF`, 18
 `\didec_compare:nNnT`, 18
 `\didec_compare:nNnTF`, 18
 `\didec_compare_p:nNn`, 18
 `\didec_gadd:nn`, 9
 `\didec_gadd_check:nn`, 9
 `\didec_gadd_to:nnn`, 9
 `\didec_gadd_to_check:nnn`, 9
 `\didec_gdiv_fp:nn`, 10
 `\didec_gdiv_fp_check:nn`, 10
 `\didec_gdiv_fp_to:nnn`, 10
 `\didec_gdiv_fp_to_check:nnn`, 10
 `\didec_gmul_fp:nn`, 10
 `\didec_gmul_fp_check:nn`, 10
 `\didec_gmul_fp_to:nnn`, 10
 `\didec_gmul_fp_to_check:nnn`, 10
 `\didec_gset:nn`, 7
 `\didec_gset_check:nn`, 7
 `\didec_gset_eq:nn`, 7
 `\didec_gset_eq_check:nn`, 7
 `\didec_gset_fp:nn`, 8
 `\didec_gset_fp_check:nn`, 8
 `\didec_gset_negative:nn`, 8
 `\didec_gset_negative_check:nn`, 8
 `\didec_gsub:nn`, 9
 `\didec_gsub_check:nn`, 9
 `\didec_gsub_to:nnn`, 9
 `\didec_gsub_to_check:nnn`, 9
 `\didec_if_kernel_fp:T`, 6
 `\didec_if_kernel_fp:TF`, 6
 `\didec_if_kernel_fp_p:`, 6
 `\didec_if_kernel_int:T`, 6
 `\didec_if_kernel_int:TF`, 6
 `\didec_if_kernel_int_p:`, 6
 `\didec_if_negative:nF`, 17
 `\didec_if_negative:nT`, 17
 `\didec_if_negative:nTF`, 17
 `\didec_if_negative_p:n`, 17
 `\didec_if_positive:nF`, 17
 `\didec_if_positive:nT`, 17
 `\didec_if_positive:nTF`, 17
 `\didec_if_positive_p:n`, 17
 `\didec_if_zero:nF`, 18
 `\didec_if_zero:nT`, 18
 `\didec_if_zero:nTF`, 18
 `\didec_if_zero_p:n`, 18
 `\didec_new:n`, 7
 `\didec_show:n`, 19
 `\didec_to_fc:n`, 11
 `\didec_to_fc_check:n`, 11
 `\didec_to_fp:n`, 11
 `\didec_to_fp_check:n`, 11
 `\didec_to_int:n`, 11
 `\didec_to_int_check:n`, 11
 `\didec_use:n`, 12
 `\didec_use_check:n`, 12
 `\didec_write:nn`, 13
 `\didec_write_check:nn`, 13
 `\didecadd`, 9
 `\dideccolformat`, 13
 `\dideccolinvformat`, 13
 `\dideccolinvuse`, 12
 `\dideccoluse`, 12
 `\didecdivfp`, 10
 `\didecformat`, 13
 `\didecifequal`, 18
 `\didecifgreaterthan`, 18
 `\dideciflowerthan`, 18
 `\didecifnegative`, 17
 `\didecifpositive`, 17
 `\didecifzero`, 18
 `\didecmulfp`, 10
 `\didecnew`, 7
 `\didecset`, 7
 `\didecsetequal`, 7
 `\didecsetfp`, 8
 `\didecsetnegative`, 8
 `\didecsetsum`, 10
 `\didecsetup`, 14
 `\didecsub`, 9
 `\didectofofc`, 11
 `\didectofofp`, 11
 `\didectooint`, 11
 `\didecuse`, 12
 `\didecwrite`, 13
`commercial` value, 16
`currency` key, 14
`currency-negative` key, 14

`decimal-separator` key, 15
`didec-blue` color, 15
`didec-green` color, 15
`didec-red` color, 15
 `\didec_color_inverse_use:n`, 12
 `\didec_color_inverse_use_check:n`, 12
 `\didec_color_use:n`, 12
 `\didec_color_use_check:n`, 12
 `\didec_compare:nNnF`, 18
 `\didec_compare:nNnT`, 18

`\didec_compare:nNnTF`, 18
`\didec_compare_p:nNn`, 18
`\didec_gadd:nn`, 9
`\didec_gadd_check:nn`, 9
`\didec_gadd_to:nnn`, 9
`\didec_gadd_to_check:nnn`, 9
`\didec_gdiv_fp:nn`, 10
`\didec_gdiv_fp_check:nn`, 10
`\didec_gdiv_fp_to:nnn`, 10
`\didec_gdiv_fp_to_check:nnn`, 10
`\didec_gmul_fp:nn`, 10
`\didec_gmul_fp_check:nn`, 10
`\didec_gmul_fp_to:nnn`, 10
`\didec_gmul_fp_to_check:nnn`, 10
`\didec_gset:nn`, 7
`\didec_gset_check:nn`, 7
`\didec_gset_eq:nn`, 7
`\didec_gset_eq_check:nn`, 7
`\didec_gset_fp:nn`, 8
`\didec_gset_fp_check:nn`, 8
`\didec_gset_negative:nn`, 8
`\didec_gset_negative_check:nn`, 8
`\didec_gsub:nn`, 9
`\didec_gsub_check:nn`, 9
`\didec_gsub_to:nnn`, 9
`\didec_gsub_to_check:nnn`, 9
`\didec_if_kernel_fp:T`, 6
`\didec_if_kernel_fp:TF`, 6
`\didec_if_kernel_fp_p:`, 6
`\didec_if_kernel_int:T`, 6
`\didec_if_kernel_int:TF`, 6
`\didec_if_kernel_int_p:`, 6
`\didec_if_negative:nF`, 17
`\didec_if_negative:nT`, 17
`\didec_if_negative:nTF`, 17
`\didec_if_negative_p:n`, 17
`\didec_if_positive:nF`, 17
`\didec_if_positive:nT`, 17
`\didec_if_positive:nTF`, 17
`\didec_if_positive_p:n`, 17
`\didec_if_zero:nF`, 18
`\didec_if_zero:nT`, 18
`\didec_if_zero:nTF`, 18
`\didec_if_zero_p:n`, 18
`\didec_new:n`, 7
`\didec_show:n`, 19
`\didec_to_fc:n`, 11
`\didec_to_fc_check:n`, 11
`\didec_to_fp:n`, 11
`\didec_to_fp_check:n`, 11
`\didec_to_int:n`, 11
`\didec_to_int_check:n`, 11
`\didec_use:n`, 12
`\didec_use_check:n`, 12
`\didec_write:nn`, 13
`\didec_write_check:nn`, 13
`\didecadd`, 9
`\dideccolformat`, 13
`\dideccolinvformat`, 13

`\dideccolinvuse`, 12
`\dideccoluse`, 12
`\didecdivfp`, 10
`\didecformat`, 13
`\didecifequal`, 18
`\didecifgreaterthan`, 18
`\dideciflowerthan`, 18
`\didecifnegative`, 17
`\didecifpositive`, 17
`\didecifzero`, 18
`\didecmulfp`, 10
`\didecnew`, 7
`\didecset`, 7
`\didecsetequal`, 7
`\didecsetfp`, 8
`\didecsetnegative`, 8
`\didecsetsum`, 10
`\didecsetup`, 14
`\didecsub`, 9
`\didectofc`, 11
`\didectofp`, 11
`\didectoint`, 11
`\didecuse`, 12
`\didecwrite`, 13
DIN-1333 value, 16

english key, 15

float key, 15

fp package option, 4

french key, 15

german key, 15

grouping-separator key, 15

half-away-from-zero value, 16

half-to-even value, 16

IEEE-754 value, 16

int package option, 4

Keys

`didec/`
`color-negative`, 15
`color-positive`, 15
`currency`, 14
`currency-negative`, 14
`decimal-separator`, 15
`english`, 15
`float`, 15
`french`, 15
`german`, 15
`grouping-separator`, 15
`rounding`, 16

mathematical value, 16

Package options

`didec/`
`fp`, 4
`int`, 4

`rounding` key, 16

Values

`commercial`, 16

`DIN-1333`, 16

`half-away-from-zero`, 16

`half-to-even`, 16

`IEEE-754`, 16

`mathematical`, 16