

This Way

ConTEXt magazine #3
May 2003

Pasting digits together
Hans Hagen
PRAGMA ADE

Each language, discipline (and sometimes designer) has its own conventions with respect to separating digits by commas, periods and/or spaces. This can be semi-automated by the `\digits` command.

The format of digits not only may differ per language, it also can depend on the content. Instead of hardcoding the punctuation in the source document, you can fall back on the `\digits` command. This command takes one argument:

This will never be a `\digits{1.000.000}` seller.

This will never be a 1.000.000 seller.

Instead of an braced argument, you can use the space as delimiter.

I will never grow longer than `\digits 1.86 \Meter`.

I will never grow longer than 1.86 m.

This command is a rather low level one and (for the moment) has no high level interface. The interpretation is controlled by `\digitmode`:

```
\setdigitmode 1 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 2 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 3 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 4 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 5 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 6 \digits 12.345,90 \digits 12.345.000 \digits 1,23
```

The six modes show up as follows:

```
12.345,90 12.345.000 1,23
12,345.90 12,345,000 1.23
12 345,90 12 345 000 1,23
12 345.90 12 345 000 1.23
12 345,90 12 345 000 1,23
12 345.90 12 345 000 1.23
```

The sign can be typeset as is or within the space of a digit.

```
\setdigitmode 0 \digits +12.345,90
\setdigitmode 1 \digits +12.345,90
\setdigitmode 2 \digits +12.345,90
\setdigitmode 3 \digits +12.345,90
```

This is typeset as:

```
+12.345,90
+12,345,90
+12,345.90
+12 345,90
```

The digit modes are:

- periods & comma
- commas & period
- thinmuskip & comma
- thinmuskip & period
- thickmuskip & comma
- thickmuskip & period

The digit parser handles a bunch of special characters as well as different formats. We strongly suggest you to use the grouped call.

- . ,. comma or period
- , ,. comma or period
- @ invisible space
- invisible space
- / invisible sign
- - minus sign
- + + plus sign
- s invisible high sign
- p + high plus sign
- m - high minus sign
- n - high minus (negative) sign
- = - zero padding

These triggers are used in the following examples.

<code>\digits 12</code>	<code>\boxed{12}</code>	12
<code>\digits {~~~.~~~.~~~.68.712,34}</code>	<code>\boxed{.....68.712,34}</code>	68.712,34
<code>\digits ~~~.~~~.~~~.68.712,34</code>	<code>\boxed{.....68.712,34}</code>	68.712,34
<code>\digits _____.____.111.68.712,34</code>	<code>\boxed{.....111.68.712,34}</code>	111.68.712,34
<code>\digits 111.111.111.68.712,34</code>	<code>\boxed{111.111.111.68.712,34}</code>	111.111.111.68.712,34
<code>\digits 12.345,90</code>	<code>\boxed{12.345,90}</code>	12.345,90
<code>\digits 12.345.000</code>	<code>\boxed{12.345.000}</code>	12.345.000
<code>\digits 12,34</code>	<code>\boxed{12,34}</code>	12,34
<code>\digits {392.857.230.68.712,34}</code>	<code>\boxed{392.857.230.68.712,34}</code>	392.857.230.68.712,34
<code>\digits 1234</code>	<code>\boxed{1234}</code>	1234
<code>\digits {1234}</code>	<code>\boxed{1234}</code>	1234
<code>\digits 1234\relax</code>	<code>\boxed{1234}</code>	1234
<code>\$_digits 123.222,00\$</code>	<code>\boxed{123.222,00}</code>	123.222,00
<code>\digits 123.222,00</code>	<code>\boxed{123.222,00}</code>	123.222,00
<code>\digits 123.222,==</code>	<code>\boxed{123.222,-...}</code>	123.222,-
<code>\digits 123.222,00^10</code>	<code>\boxed{123.222,00\cdot 10^{10}}</code>	$123.222,00 \cdot 10^{10}$
<code>\digits 123.222,00e10</code>	<code>\boxed{123.222,00\cdot 10^{10}}</code>	$123.222,00 \cdot 10^{10}$
<code>\digits /123.222,00e-12</code>	<code>\boxed{...123.222,00\cdot 10^{-12}}</code>	$123.222,00 \cdot 10^{-12}$
<code>\digits -123.222,00e-12</code>	<code>\boxed{-123.222,00\cdot 10^{-12}}</code>	$-123.222,00 \cdot 10^{-12}$
<code>\digits +123.222,00e-12</code>	<code>\boxed{+123.222,00\cdot 10^{-12}}</code>	$+123.222,00 \cdot 10^{-12}$

<code>\digits n123.222,00e-12</code>	$\overline{-123.222,00 \cdot 10^{-12}}$	$-123.222,00 \cdot 10^{-12}$
<code>\digits s123.222,00e-12</code>	$\underline{123.222,00 \cdot 10^{-12}}$	$123.222,00 \cdot 10^{-12}$
<code>\digits p123.222,00e-12</code>	$\overline{+123.222,00 \cdot 10^{-12}}$	$+123.222,00 \cdot 10^{-12}$

The `\digits` command behaves well under different circumstances, like:

```
{\digits1234}
\digits{1234}
\digits 1234\whatever
$\digits 123.222,00$
\digits 123.222,00.
```

The latter case shows us that trailing non digits are to be passed untreated.

Another interesting case is:

```
\digits 123.222,00^10
```

or typeset:

$$123.222,00 \cdot 10^{10}$$

The separator is defined as:

```
\def\digitpowerseparator{\cdot10}
```

You can change this in for instance:

```
\def\digitpowerseparator{\times10}
```

Users can specify the way they enter those digits by saying something like:

```
\digittemplate 12.000.000,00
```

or

```
\digittemplate .,
```

The tabulate environment has digits support built in by means of `\MN`. This feature should not be confused with the `g` key and associated `\NG` column starter, which can be used to align on a symbol.

```
\starttabulate[|l|c|r|g{.}|g{!}|r|]
\NC 1      \NC 1      \NC 1      \NG 1      \NG 1      \NN 1.@@ \NC \NR
\NC 1.1    \NC 1.1    \NC 1.1    \NG 1.1    \NG 1!1    \NN 1.1@ \NC \NR
\NC 1.11   \NC 1.11   \NC 1.11   \NG 1.11   \NG 1!11   \NN 1.11 \NC \NR
\NC 11.11  \NC 11.11  \NC 11.11  \NG 11.11  \NG 11!11  \NN 11.11 \NC \NR
\stoptabulate
```

The cell data entries must be delimited by a space or enclosed in braces.

1	1	1	1	1	1
1.1	1.1	1.1	1.1	1!1	1.1
1.11	1.11	1.11	1.11	1!11	1.11
11.11	11.11	11.11	11.11	11!11	11.11

source code of this document

```
\usemodule[mag-01,units]
```

```
\setvariables
```

```
[magazine]
[title={Pasting digits together},
author=Hans Hagen,
affiliation=PRAGMA ADE,
date=May 2003,
number=3]
```

```
\startbuffer[abstract]
```

Each language, discipline (and sometimes designer) has its own conventions with respect to separating digits by commas, periods and/or spaces. This can be semi-automated by the `\type {\digits}` command.

```
\stopbuffer
```

```
\starttext \setups [titlepage] \setups [title]
```

The format of digits not only may differ per language, it also can depend on the content. Instead of hardcoding the punctuation in the source document, you can fall back on the `\type {\digits}` command. This command takes one argument:

```
\startbuffer
```

This will never be a `\digits{1.000.000}` seller.

```
\stopbuffer
```

```
\typebuffer \getbuffer
```

Instead of an braced argument, you can use the space as delimiter.

```
\startbuffer
```

I will never grow longer than `\digits 1.86 \Meter`.

```
\stopbuffer
```

```
\typebuffer \getbuffer
```

This command is a rather low level one and (for the moment) has no high level interface. The interpretation is controlled by `\type {\digitmode}`:

source code of this document

```

\startbuffer
\setdigitmode 1 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 2 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 3 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 4 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 5 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\setdigitmode 6 \digits 12.345,90 \digits 12.345.000 \digits 1,23
\stopbuffer

```

```
\typebuffer
```

The six modes show up as follows:

```

\startlines
\getbuffer
\stoptlines

```

The sign can be typeset as is or within the space of a digit.

```

\startbuffer
\setdigitmode 0 \digits +12.345,90
\setdigitmode 1 \digits +12.345,90
\setdigitmode 2 \digits +12.345,90
\setdigitmode 3 \digits +12.345,90
\stopbuffer

```

```
\typebuffer
```

This is typset as:

```

\startlines
\getbuffer
\stoptlines

```

The digit modes are:

```

\startitemize[packed]
\item periods \& comma
\item commas \& period
\item thinmuskip \& comma
\item thinmuskip \& period
\item thickmuskip \& comma

```

source code of this document

```
\item thickmuskip \& period
\stopitemize
```

The digit parser handles a bunch of special characters as well as different formats. We strongly suggest you to use the grouped call.

```
\starttabulate[|l|l|l|]
\NC \type{.} \NC , . \NC comma or period \NC \NR
\NC \type{,} \NC , . \NC comma or period \NC \NR
\NC \type{@} \NC \NC invisible space \NC \NR
\NC \type{ } \NC \NC invisible space \NC \NR
\NC \type{/} \NC \NC invisible sign \NC \NR
\NC \type{-} \NC $-$ \NC minus sign \NC \NR
\NC \type{+} \NC $+$ \NC plus sign \NC \NR
\NC \type{s} \NC \NC invisible high sign \NC \NR
\NC \type{p} \NC $\positive$ \NC high plus sign \NC \NR
\NC \type{m} \NC $\negative$ \NC high minus sign \NC \NR
\NC \type{n} \NC $\negative$ \NC high minus (negative) sign \NC \NR
\NC \type{=} \NC $\zeroamount$ \NC zero padding \NC \NR
\stoptabulate
```

These triggers are used in the following examples.

```
\def\Sample#1%
  {\NC \convertargument#1\to\ascii\tttf\ascii
  \NC \ruledhbox{#1}
  \NC #1 \NC \NR}

\starttabulate[|l|l|l|]
\Sample {\digits 12 }
\Sample {\digits{~~~.~~~.~~~.68.712,34} }
\Sample {\digits ~~~.~~~.~~~.68.712,34 }
\Sample {\digits ___.___.111.68.712,34 }
\Sample {\digits 111.111.111.68.712,34 }
\Sample {\digits 12.345,90 }
\Sample {\digits 12.345.000 }
\Sample {\digits 12,34 }
\Sample {\digits{392.857.230.68.712,34}}
\Sample {{\digits1234}}
\Sample {\digits{1234}}
\Sample {\digits 1234\relax}
\Sample {$\digits 123.222,00$ }
\Sample {\digits 123.222,00 }
```


source code of this document

```
\Sample {\digits 123.222,== }
\Sample {\digits 123.222,00^10 }
\Sample {\digits 123.222,00e10 }
\Sample {\digits /123.222,00e-12 }
\Sample {\digits -123.222,00e-12 }
\Sample {\digits +123.222,00e-12 }
\Sample {\digits n123.222,00e-12 }
\Sample {\digits s123.222,00e-12 }
\Sample {\digits p123.222,00e-12 }
\stoptabulate
```

The `\type {\digits}` command behaves well under different circumstances, like:

```
\starttyping
{\digits1234}
\digits{1234}
\digits 1234\whatever
$\digits 123.222,00$
\digits 123.222,00.
\stoptyping
```

The latter case shows us that trailing non digits are to be passed untreated.

Another interesting case is:

```
\startbuffer
\digits 123.222,00^10
\stopbuffer

\typebuffer
```

or typeset:

```
\getbuffer
```

The separator is defined as:

```
\starttyping
\def\digitpowerseparator{\cdot10}
\stoptyping
```

You can change this in for instance:

source code of this document

```
\starttyping
\def\digitpowerseparator{\times10}
\stoptyping
```

Users can specify the way they enter those digits by saying something like:

```
\starttyping
\digittemplate 12.000.000,00
\stoptyping
```

or

```
\starttyping
\digittemplate .,
\stoptyping
```

The tabulate environment has digits support built in by means of `\type {\NN}`. This feature should not be confused with the `\type {g}` key and associated `\type {\NG}` column starter, which can be used to align on a symbol.

```
\startbuffer
\starttabulate[l|c|r|g{.}|g{!}|r|]
\NC 1      \NC 1      \NC 1      \NG 1      \NG 1      \NN 1.@@ \NC \NR
\NC 1.1    \NC 1.1    \NC 1.1    \NG 1.1    \NG 1!1    \NN 1.1@ \NC \NR
\NC 1.11   \NC 1.11   \NC 1.11   \NG 1.11   \NG 1!11   \NN 1.11 \NC \NR
\NC 11.11  \NC 11.11  \NC 11.11  \NG 11.11  \NG 11!11  \NN 11.11 \NC \NR
\stoptabulate
\stopbuffer
```

```
\typebuffer
```

The cell data entries must be delimited by a space or enclosed in braces.

```
\getbuffer
```

```
\setups [listing] \setups [lastpage] \stoptext
```

source code of this document

