Contents

	
texdimens
		Copyright and License ............................................ 1
		Usage ................................................................. 1
		Aim of this package .................................................. 1
		Macros of this package (summary) ................................. 1
		Quick review of basics: TeX points and scaled points ............ 2
		Further units known to TeX on input ............................... 3
		Macros of this package (full list) ................................... 5
		Acknowledgements ..................................................... 13


texdimens

Copyright and License

Copyright (c) 2021 Jean-François Burnol

This file is part of the texdimens package distributed under the LPPL 1.3c. See file LICENSE.md.

Repository: https://github.com/jfbu/texdimens

Release: 1.0 2021/11/10

Usage

Utilities and documentation related to TeX dimensional units, usable:

• with Plain TeX: \input texdimens

• with LaTeX: \usepackage{texdimens}

Aim of this package

The aim of this package is to provide facilities to express dimensions (or dimension expressions evaluated by \dimexpr) using the various available TeX units, to the extent possible.

Macros of this package (summary)

This package provides expandable macros:

• \textdimenUU with UU standing for one of pt, bp, cm, mm, in, pc, cc, nc, dd and nd,
• \textdimenUUup and \textdimenUUdown with UU as above except pt,
• \textdimenbothincm and relatives,
• \textdimenbothbpm and relatives,
\textdimenwithunit.

\textdimenbp takes on input some dimension or dimension expression and produces on output a decimal \( D \) such that \( D \ \text{bp} \) is guaranteed to be the same dimension as the input, if it admits any representation as \( E \ \text{bp} \); else it will be either the closest match from above or from below. For this unit, as well as for nd and dd the difference is at most 1sp. For other units (not pt of course) the distance will usually be larger than 1sp and one does not know if the approximant from the other direction would have been better or worst.

The variants \textdimenbpup and \textdimenbpdown expand slightly less fast than \textdimenbp but they allow to choose the direction of approximation (in absolute value).

The macros associated to the other units have the same descriptions.

\textdimenbothincm, respectively \textdimenbothbmm, find the largest (in absolute value) dimension not exceeding the input and exactly representable both with the in and cm units, respectively exactly representable both with the bp and mm units.

\textdimenwithunit{<dim1>}{<dim2>} produces a decimal \( D \) such that \( D \ \dimexpr \ dimen2 \relax \) is parsed by TeX into the same dimension as \( \dimen1 \) if this is at all possible. If \( \dimen2<1\text{pt} \) all TeX dimensions \( \dimen1 \) are attainable. If \( \dimen2>1\text{pt} \) not all \( \dimen1 \) are attainable. If not attainable, the decimal \( D \) will ensure a closest match from below or from above but one does not know if the approximation from the other direction is better or worst.

In a sense, this macro divides \( \dimen1 \) by \( \dimen2 \), see additional details in the complete macro description.

**Quick review of basics: TeX points and scaled points**

This project requires the e-TeX extensions \texttt{\dimexpr} and \texttt{\numexpr}. The notation \texttt{<dim. expr.>} in the macro descriptions refers to a *dimensional expression* as accepted by \texttt{\dimexpr}. The syntax has some peculiarities: among them the fact that \texttt{-(...)} (for example \texttt{-3pt}) is illegal, one must use alternatives such as \texttt{0pt-(...)} or a sub-expression \texttt{-\dimexpr...\relax} for example.

TeX dimensions are represented internally by a signed integer which is in absolute value at most \( 0x3FFFFFFF \), i.e. \( 1073741823 \). The corresponding unit is called the "scaled point", i.e. 1sp is \( 1/65536 \) of one TeX point 1pt, or rather 1pt is represented internally as 65536.

If \texttt{\foo} is a dimen register:

- \texttt{\number\foo} produces the integer \( N \) such as \texttt{\foo} is the same as \( N\text{sp} \),
- inside \texttt{\numexpr}, \texttt{\foo} is replaced by \( N \),
• \the\foo produces a decimal D (with at most five places) followed with pt (catcode 12 tokens) and this output Dpt can serve as input in a dimen assignment to produce the same dimension as \foo. One can also use the catcode 11 characters pt for this. Digits and decimal mark must have their standard catcode 12.

When TeX encounters a dimen denotation of the type Dpt it will compute N in a way equivalent to N = round(65536 D) where ties are rounded away from zero. Only 17 decimal places of D are kept as it can be shown that going beyond can not change the result.

When \foo has been assigned as Dpt, \the\foo will produce some Ept where E is not necessarily the same as D. But it is guaranteed that Ept defines the same dimension as Dpt.

Further units known to TeX on input

TeX understands on input further units: bp, cm, mm, in, pc, cc, nc, dd and nd. It also understands font-dependent units ex and em, and PDFTeX adds the px dimension unit. Japanese engines also add specific units.

The ex, em, and px units are handled somewhat differently by (pdf)TeX than bp, cm, mm, in, pc, cc, nc, dd and nd units. For the former (let’s use the generic notation uu), the exact same dimensions are obtained from an input D uu where D is some decimal or from D <dimen> where <dimen> stands for some dimension register which records 1uu or \dimexpr1uu\relax. In contrast, among the latter, i.e. the core TeX units, this is false except for the pc unit.

TeX associates (explicitly for the core units, implicitly for the units corresponding to internal dimensions) to each unit uu a fraction phi which is a conversion factor. For the internal dimensions ex, em, px or in the case of multiplying a dimension by a decimal, this phi is morally $f/65536$ where $f$ is the integer such that 1 uu=$f$ sp. For core units however, the hard-coded ratio n/d never has a denominator d which is a power of 2, except for the pc whose associated ratio factor is 12/1 (and arguably for the sp for which morally phi is 1/65536 but we keep it separate from the general discussion; as well as pt with its unit conversion factor).

Here is a table with the hard-coded conversion factors:

<table>
<thead>
<tr>
<th>uu</th>
<th>phi</th>
<th>reduced</th>
<th>real approximation (Python output)</th>
<th>\the&lt;uu&gt;</th>
<th>luu in sp= [65536phi]</th>
</tr>
</thead>
<tbody>
<tr>
<td>bp 7227/7200</td>
<td>803/800</td>
<td>1.00375</td>
<td>65781</td>
<td>1.00374pt</td>
<td></td>
</tr>
<tr>
<td>nd 685/642</td>
<td>same</td>
<td>1.0669781931464175</td>
<td>69925</td>
<td>1.06697pt</td>
<td></td>
</tr>
<tr>
<td>dd 1238/1157</td>
<td>same</td>
<td>1.070008643042351</td>
<td>70124</td>
<td>1.07pt</td>
<td></td>
</tr>
<tr>
<td>mm 7227/2540</td>
<td>same</td>
<td>2.8452755905511813</td>
<td>186467</td>
<td>2.84526pt</td>
<td></td>
</tr>
<tr>
<td>pc 12/1</td>
<td>12</td>
<td>12.0</td>
<td>786432</td>
<td>12.0pt</td>
<td></td>
</tr>
</tbody>
</table>
The values of \texttt{1uu} in the \texttt{sp} and \texttt{pt} units are irrelevant and even misleading regarding the \TeX{} parsing of \texttt{D uu} input. Notice for example that \texttt{\the\dimexpr1bp\relax} gives \texttt{1.00374pt} but the actual conversion factor is \texttt{1.00375} (and \texttt{1.00375pt=65782sp>1bp}). Similarly \texttt{\the\dimexpr1in\relax} outputs \texttt{72.26999pt} and is represented internally as \texttt{4736286sp} but the actual conversion factor is \texttt{72.27=7227/100}, and \texttt{72.27pt=4736287sp>1in}. And for the other units except the \texttt{pc}, the conversion factors are not decimal numbers, so even less likely to match \texttt{\the<1uu>} as listed in the last column. Their denominators are not powers of \texttt{2} so they don’t match exactly either \texttt{(1uu in sp)/65536} but are only close.

When \TeX{} parses an assignment \texttt{U uu} with a decimal \texttt{U} and a unit \texttt{uu}, be it a core unit, or a unit corresponding to an internal dimension, it first handles \texttt{U} as with the \texttt{pt} unit. This means that it computes \texttt{N = round(65536*U)}. It then multiplies this \texttt{N} by the conversion factor \texttt{phi} and truncates towards zero the mathematically exact result to obtain an integer \texttt{T}: \texttt{T=trunc(N*phi)}. The assignment \texttt{Uuu} is concluded by defining the value of the dimension to be \texttt{Tsp}.

Regarding the core units, we always have \texttt{phi>1}. The increasing sequence \texttt{0<=trunc(phi)<=trunc(2phi)<=...} is thus \textit{strictly increasing} and, as \texttt{phi} is never astronomically close to \texttt{1}, it \textbf{always has jumps}: no \TeX{} dimensions can be obtained from an assignment using a core unit distinct from the \texttt{pt} (and \texttt{sp} of course, but we already said it was kept out of the discussion here).

On the other hand when \texttt{phi<1}, then the sequence \texttt{trunc(N phi)} is not strictly increasing, already because \texttt{trunc(phi)=0} and besides here \texttt{phi=f/65536}, so the \texttt{65536} integers \texttt{0...65535} are mapped to \texttt{f} integers \texttt{0...(f-1)} inducing non one-to-oneness. But all integers in the \texttt{0...(2**30-1)} range will be attained for some input, so there is surjectivity.

The “worst” unit is the largest i.e. the \texttt{in} whose conversion factor is \texttt{72.27}. The simplest unit to understand is the \texttt{pc} as it corresponds to an integer ratio \texttt{12}: only dimensions which in scaled points are multiple of \texttt{12} are exactly representable in the \texttt{pc} unit.

This also means that some dimensions expressible in one unit may not be available with another unit. For example, and perhaps surprisingly, there is no decimal \texttt{D} which would achieve \texttt{1in==Dcm}: the “step” between attainable dimensions is \texttt{72--73sp} for the \texttt{in} and \texttt{28--29sp} for the \texttt{cm}, and as \texttt{1in} differs internally from \texttt{2.54cm} by only \texttt{12sp} it is impossible to adjust either the \texttt{in} side or the \texttt{cm} side to obtain equality.
In particular 1\text{in}==2.54\text{cm} is \textbf{false} in \TeX, but it is true that 100\text{in}==254\text{cm} . . . (it is already true that 50\text{in}==127\text{cm}). It is also false that 10\text{in}==25.4\text{cm} but it is true that 100\text{in}==254\text{mm} . . . It is false though that 1\text{in}==25.4\text{mm}!

>>> (\dimexpr1\text{in}, \dimexpr2.54\text{cm});
@_1 4736286, 4736274

>>> (\dimexpr10\text{in}, \dimexpr25.4\text{cm});
@_2 47362867, 47362855

>>> (\dimexpr100\text{in}, \dimexpr254\text{cm});
@_3 473628672, 473628672

>>> (\dimexpr1\text{in}, \dimexpr25.4\text{mm});
@_4 4736286, 4736285

>>> (\dimexpr10\text{in}, \dimexpr254\text{mm});
@_5 47362867, 47362867

\texttt{\textbackslash maxdimen} can be expressed only with pt, bp, and nd. For the other core units the maximal attainable dimensions in \texttt{sp} unit are given in the middle column of the next table.

\begin{tabular}{lcll}
maximal allowed & the corresponding & minimal \TeX dimen denotation & causing "Dimension too large"
\hline
(with 5 places) & maximal attainable dim. & & \\
\hline
16383.99999pt & 1073741823sp & 16383.99999237060546875pt & \\
16322.78954bp & 1073741823sp & 16322.78954315185546875bp & \\
15355.51532nd & 1073741823sp & 15355.51532437060546875nd & \\
15312.02584dd & 1073741822sp & 15312.02584075927734375dd & \\
5758.31742mm & 1073741822sp & 5758.31742096240234375mm & \\
1365.33333pc & 1073741820sp & 1365.33333587646484375pc & \\
1279.62627nc & 1073741814sp & 1279.62627410888671875nc & \\
1276.00215cc & 1073741821sp & 1276.00215896240234375cc & \\
575.83174cm & 1073741822sp & 575.83174896240234375cm & \\
226.70540in & 1073741768sp & 226.70540618896484375in & \\
\end{tabular}

Perhaps for these various peculiarities with dimensional units, \TeX does not provide an output facility for them similar to what \texttt{\textbackslash the} achieves for the pt.

**Macros of this package (full list)**

The macros are all expandable, and most are f-expandable (check the source code). They parse their arguments via \texttt{\dimexpr} so can be nested (with appropriate units added, as the outputs always are bare decimal numbers).

Negative dimensions behave as if replaced by their absolute value, then at last step the sign (if result is not zero) is applied (so “down” means “towards zero”, and “up” means “away from zero”).
Remarks about “Dimension too large” issues:

1. For input $X$ equal to $\maxdimen$ (or differing by a few sp’s) and those units $uu$ for which $\maxdimen$ is not exactly representable (i.e. all core units except pt, bp and nd), the output $D$ of the “up” macros $\text{texdimen}<uu>\text{up}\{X\}$, if used as $Duu$ in a dimension assignment or expression, will (as is logical) trigger a “Dimension too large” error.

2. For $dd$, $nc$ and $in$, it turns out that $\text{texdimen}<uu>\{X\}$ chooses the “up” approximant for $X$ equal to or very near $\maxdimen$ (check the respective macro documentations), i.e. the output $D$ is such that $Duu$ is the first virtually attainable dimension beyond $\maxdimen$. Hence $Duu$ will trigger on use a “Dimension too large error”. With the other units for which $\maxdimen$ is not attainable exactly, $\text{texdimen}<uu>\{\maxdimen\}$ output is by luck the “down” approximant.

3. Similarly the macro $\text{texdimenwithunit}\{D1pt\}\{D2pt\}$ covers the entire dimension range, but its output $F$ for $D1pt$ equal to or very close to $\maxdimen$ may be such that $F<D2pt>$ represents a dimension beyond $\maxdimen$, if the latter is not exactly representable. Hence $F<D2pt>$ would trigger “Dimension too large” on use. This can only happen if $D2pt>1pt$ and (roughly) $D1pt>\maxdimen-D2sp$. As $D2sp$ is less than 0.25pt, this is not likely to occur in real life practice except if deliberately targeting $\maxdimen$. For $D2pt<1pt$, all dimensions $D1pt$ are exactly representable, in particular $\maxdimen$, and the output $F$ will always be such that TeX parses $F<D2pt>$ into exactly the same dimension as $D1pt$.

$\text{texdimenpt}\{\text{dim. expr.}\}$

Does $\text{the}\dimexpr <\text{dim. expr.}> \relax$ then removes the pt.

$\text{texdimenbp}\{\text{dim. expr.}\}$

Produces a decimal (with up to five decimal places) $D$ such that $Dbp$ represents the dimension exactly if possible. If not possible it will differ by 1sp from the original dimension, but it is not known in advance if it will be above or below.

$\maxdimen$ on input produces 16322.78954 and indeed is realized as 16322.78954bp.

$\text{texdimenbpdown}\{\text{dim. expr.}\}$

Produces a decimal (with up to five decimal places) $D$ such that $Dbp$ represents the dimension exactly if possible. If not possible it will be smaller by 1sp from the original dimension.

$\text{texdimenbpup}\{\text{dim. expr.}\}$

Produces a decimal (with up to five decimal places) $D$ such that $Dbp$ represents the dimension exactly if possible. If not possible it will be
larger by 1sp from the original dimension.

\textdimennd{<dim. expr.>}

Produces a decimal (with up to five decimal places) D such that Dnd represents the dimension exactly if possible. If not possible it will differ by 1sp from the original dimension, but it is not known in advance if it will be above or below.

\maxdimen on input produces 15355.51532 and indeed is realized as 15355.51532nd.

\textdimennddown{<dim. expr.>}

Produces a decimal (with up to five decimal places) D such that Dnd represents the dimension exactly if possible. If not possible it will be smaller by 1sp from the original dimension.

\textdimenndup{<dim. expr.>}

Produces a decimal (with up to five decimal places) D such that Dnd represents the dimension exactly if possible. If not possible it will be larger by 1sp from the original dimension.

\textdimendd{<dim. expr.>}

Produces a decimal (with up to five decimal places) D such that Ddd represents the dimension exactly if possible. If not possible it will differ by 1sp from the original dimension, but it is not known in advance if it will be above or below.

Warning: the output for \maxdimen is 15312.02585 but 15312.02585dd will trigger on use “Dimension too large” error. \maxdimen−1sp is the maximal input for which the output remains less than \maxdimen (max attainable dimension: \maxdimen−1sp).

\textdimenddddown{<dim. expr.>}

Produces a decimal (with up to five decimal places) D such that Ddd represents the dimension exactly if possible. If not possible it will be smaller by 1sp from the original dimension.

\textdimendddup{<dim. expr.>}

Produces a decimal (with up to five decimal places) D such that Ddd represents the dimension exactly if possible. If not possible it will be larger by 1sp from the original dimension.

If input is \maxdimen, then Ddd virtually represents \maxdimen+1sp and will trigger on use “Dimension too large”.

\textdimenmm{<dim. expr.>}

7
Produces a decimal (with up to five decimal places) $D$ such that \( D_{mm} \) represents the dimension exactly if possible. If not possible it will either be the closest from below or from above, but it is not known in advance which one (and it is not known if the other choice would have been closer).

\( \texttt{\textbackslash maxdimen} \) as input produces on output 5758.31741 and indeed the maximal attainable dimension is 5758.31741mm (\( \texttt{\textbackslash maxdimen-1sp} \)).

\texttt{\textbackslash texdimen\textbackslash m\textbackslash m\textbackslash d\textbackslash o\textbackslash w\textbackslash n\textbackslash d\textbackslash \{<\text{dim. expr.}>\}}

Produces a decimal (with up to five decimal places) $D$ such that $D_{mm}$ represents the dimension exactly if possible. If not possible it will be largest representable dimension smaller than the original one.

\texttt{\textbackslash texdimen\textbackslash m\textbackslash m\textbackslash u\textbackslash p\textbackslash \{<\text{dim. expr.}>\}}

Produces a decimal (with up to five decimal places) $D$ such that $D_{mm}$ represents the dimension exactly if possible. If not possible it will be smallest representable dimension larger than the original one.

If input is \( \texttt{\textbackslash maxdimen} \), then $D_{mm}$ virtually represents $\texttt{\textbackslash maxdimen+2sp}$ and will trigger on use “Dimension too large”.

\texttt{\textbackslash texdimen\textbackslash p\textbackslash c\textbackslash \{<\text{dim. expr.}>\}}

Produces a decimal (with up to five decimal places) $D$ such that $D_{pc}$ represents the dimension exactly if possible. If not possible it will be the closest representable one (in case of tie, the approximant from above is chosen).

\( \texttt{\textbackslash maxdimen} \) as input produces on output 1365.33333 and indeed the maximal attainable dimension is 1365.33333pc (\( \texttt{\textbackslash maxdimen-3sp} \)).

\texttt{\textbackslash texdimen\textbackslash p\textbackslash c\textbackslash d\textbackslash o\textbackslash w\textbackslash n\textbackslash d\textbackslash \{<\text{dim. expr.}>\}}

Produces a decimal (with up to five decimal places) $D$ such that $D_{pc}$ represents the dimension exactly if possible. If not possible it will be largest representable dimension smaller than the original one.

\texttt{\textbackslash texdimen\textbackslash p\textbackslash c\textbackslash u\textbackslash p\textbackslash \{<\text{dim. expr.}>\}}

Produces a decimal (with up to five decimal places) $D$ such that $D_{pc}$ represents the dimension exactly if possible. If not possible it will be smallest representable dimension larger than the original one.

If input is \( \texttt{\textbackslash maxdimen-3sp} \), then $D_{pc}$ virtually represents $\texttt{\textbackslash maxdimen+9sp}$ and will trigger on use “Dimension too large”.

\texttt{\textbackslash texdimen\textbackslash n\textbackslash c\textbackslash \{<\text{dim. expr.}>\}}
Produces a decimal (with up to five decimal places) \( D \) such that \( \texttt{Dnc} \) represents the dimension exactly if possible. If not possible it will either be the closest from below or from above, but it is not known in advance which one (and it is not known if the other choice would have been closer).

Warning: the output for \texttt{\textbackslash maxdimen-1sp} is 1279.62628 but 1279.62628\texttt{nc} will trigger on use “Dimension too large” error. \texttt{\textbackslash maxdimen-2sp} is the maximal input for which the output remains less than \texttt{\textbackslash maxdimen} (max attainable dimension: \texttt{\textbackslash maxdimen-9sp}).

\texttt{\textbackslash texdimenncdown{<dim. expr.>}}

Produces a decimal (with up to five decimal places) \( D \) such that \( \texttt{Dnc} \) represents the dimension exactly if possible. If not possible it will be largest representable dimension smaller than the original one.

\texttt{\textbackslash texdimenncup{<dim. expr.>}}

Produces a decimal (with up to five decimal places) \( D \) such that \( \texttt{Dnc} \) represents the dimension exactly if possible. If not possible it will be smallest representable dimension larger than the original one.

If input is \texttt{>\textbackslash maxdimen-9sp}, then \( \texttt{Dnc} \) virtually represents \texttt{\textbackslash maxdimen+4sp} and will trigger on use “Dimension too large”.

\texttt{\textbackslash texdimencc{<dim. expr.>}}

Produces a decimal (with up to five decimal places) \( D \) such that \( \texttt{Dcc} \) represents the dimension exactly if possible. If not possible it will either be the closest from below or from above, but it is not known in advance which one (and it is not known if the other choice would have been closer).

\texttt{\textbackslash maxdimen} as input produces on output 1276.00215 and indeed the maximal attainable dimension is 1276.00215\texttt{cc} (\texttt{\textbackslash maxdimen-2sp}).

\texttt{\textbackslash texdimenccdown{<dim. expr.>}}

Produces a decimal (with up to five decimal places) \( D \) such that \( \texttt{Dcc} \) represents the dimension exactly if possible. If not possible it will be largest representable dimension smaller than the original one.

\texttt{\textbackslash texdimenccup{<dim. expr.>}}

Produces a decimal (with up to five decimal places) \( D \) such that \( \texttt{Dcc} \) represents the dimension exactly if possible. If not possible it will be smallest representable dimension larger than the original one.

If input is \texttt{>\textbackslash maxdimen-2sp}, then \( \texttt{Dcc} \) virtually represents \texttt{\textbackslash maxdimen+11sp} and will trigger on use “Dimension too large”.
\textdimencm{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $D_{cm}$ represents the dimension exactly if possible. If not possible it will either be the closest from below or from above, but it is not known in advance which one (and it is not known if the other choice would have been closer).

\maxdimen\text{ as input produces on output 575.83174 and indeed the maximal attainable dimension is 575.83174cm (\maxdimen-1sp).}

\textdimencmdown{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $D_{cm}$ represents the dimension exactly if possible. If not possible it will be largest representable dimension smaller than the original one.

\textdimencmup{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $D_{cm}$ represents the dimension exactly if possible. If not possible it will be smallest representable dimension larger than the original one.

If input is $\maxdimen$, then $D_{cm}$ virtually represents $\maxdimen+28sp$ and will trigger on use “Dimension too large”.

\textdimenin{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $D_{in}$ represents the dimension exactly if possible. If not possible it will either be the closest from below or from above, but it is not known in advance which one (and it is not known if the other choice would have been closer).

Warning: the output for $\maxdimen-18sp$ is 226.70541 but 226.70541in will trigger on use “Dimension too large” error. $\maxdimen-19sp$ is the maximal input for which the output remains less than $\maxdimen$ (max attainable dimension: $\maxdimen-55sp$).

\textdimenindown{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $D_{in}$ represents the dimension exactly if possible. If not possible it will be largest representable dimension smaller than the original one.

\textdimeninup{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $D_{in}$ represents the dimension exactly if possible. If not possible it will be smallest representable dimension larger than the original one.
If input is $>\maxdimen-55sp$, then Din virtually represents $\maxdimen+17sp$ and will trigger on use “Dimension too large”.

\textdimenbothcmin{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that Din is the largest dimension not exceeding the original one (in absolute value) and exactly representable both in the in and cm units.

\textdimenbothincm{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $Dcm$ is the largest dimension not exceeding the original one (in absolute value) and exactly representable both in the in and cm units. Thus both expressions \textdimenbothcmin{<dim. expr.>} in and \textdimenbothincm{<dim. expr.>} cm represent the same dimension.

\textdimenbothcminpt{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $Dpt$ is the largest dimension not exceeding the original one (in absolute value) and exactly representable both in the in and cm units. It thus represents the same dimension as the one determined by \textdimenbothcmin and \textdimenbothincm.

\textdimenbothincmpt{<dim. expr.>}

Alias for \textdimenbothcminpt.

\textdimenbothcminsp{<dim. expr.>}

Produces an integer (explicit digit tokens) $N$ such that $Nsp$ is the largest dimension not exceeding the original one in absolute value and exactly representable both in the in and cm units.

\textdimenbothincmsp{<dim. expr.>}

Alias for \textdimenbothcminsp.

\textdimenbothbpm{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $Dmm$ is the largest dimension smaller (in absolute value) than the original one and exactly representable both in the bp and mm units.

\textdimenbothmmbp{<dim. expr.>}

Produces a decimal (with up to five decimal places) $D$ such that $Dbp$ is the largest dimension smaller (in absolute value) than the original one and exactly representable both in the bp and mm units. Thus
\textdimenbothbmpbp{<dim. expr.>}bp is the same dimension as \textdimenbothbpmmpmm{<dim. expr.>}mm.

\textdimenbothbpmmpmpt{<dim. expr.>}

Produces a decimal (with up to five decimal places) \(D\) such that \(D\text{pt}\) is the largest dimension not exceeding the original one and exactly representable both in the bp and mm units.

\textdimenbothbmpmbppt{<dim. expr.>}

Alias for \textdimenbothbpmmpmpt.

\textdimenbothbmpmmmsp{<dim. expr.>}

Produces an integer (explicit digit tokens) \(N\) such that \(N\text{sp}\) is the largest dimension not exceeding the original one and exactly representable both in the bp and mm units.

\textdimenbothbmpbpsp{<dim. expr.>}

Alias for \textdimenbothbmpmmmsp.

\textdimenwithunit{<dim. expr. 1>}{<dim expr. 2>}

Produces a decimal \(D\) such that \(D\dimexpr <\text{dim expr. 2}>\relax\) is considered by TeX the same as \(<\text{dim. expr. 1}>\) if at all possible. If the (assumed non zero) second argument \(<\text{dim2}>\) is at most \(1\text{pt}\) (in absolute value), then this is always possible. If the second argument \(<\text{dim2}>\) is \(>1\text{pt}\) then this is not always possible and the output \(D\) will ensure for \(D<\text{dim2}>\) to be a closest match to the first argument \(<\text{dim1}>\) either from above or below, but one does not know if the other direction would have given a better or worst match.

\textdimenwithunit{<dim>}{1bp} and \textdimenbp{<dim>} are not the same: The former produces a decimal \(D\) such that \(D\dimexpr \text{1bp}\relax\) is represented internally as \(<\text{dim}>\) if at all possible, whereas the latter produces a decimal \(D\) such that \(D\text{ bp}\) is the one aiming at being the same as \(<\text{dim}>\). Using \(D\dimexpr \text{1bp}\relax\) implies a conversion factor equal to \(65781/65536\), whereas \(D\text{ bp}\) involves the \(803/800\) conversion factor.

\textdimenwithunit{D1pt}{D2pt} output is close to the mathematical ratio \(D1/D2\). But notwithstanding the various unavoidable “errors” arising from conversion of decimal inputs to binary internals, and from the latter to the former, the output \(R\) will tend to be on average slightly larger (in its last decimal) than mathematical \(D1/D2\). The root cause being that the specification for \(R\) is that \(R<\text{D2pt}>\) must be exactly \(<\text{D1pt}>\) after TeX parsing, if at all possible; and it turns out this is always possible for \(\text{D2pt}<1\text{pt}\). The final step in the TeX parsing of a multiplication
of a dimension by a scalar is a truncation to an integer multiple of the \( \text{sp}=1/65536\text{pt} \) unit, not a rounding. So \( R \) is basically (i.e. before conversion to a decimal) \( \text{ceil}(D_1/D_2, 16) \), or to be more precise it is obtained as \( \text{ceil}(N_1/N_2, 16) \) with \( D_1\text{pt} \to N_1\text{sp} \), \( D_2\text{pt} \to N_2\text{sp} \) and the second argument of \( \text{ceil} \) means that 16 binary places are used. This formula is the one used for \( D_2\text{pt}<1\text{pt} \), for \( D_2\text{pt}>1\text{pt} \) the mathematics is different, but the implication that \( R \) has a (less significant) bias to be “shifted upwards” (in its last decimal place) compared to the (rounded) value \( D_1/D_2 \) or rather \( N_1/N_2 \) still stands.

Acknowledgements

Thanks to Denis Bitouzé for raising an issue on the LaTeX3 tracker which became the initial stimulus for this package.

Thanks to Ruixi Zhang for reviving the above linked-to thread and opening up on the package issue tracker the issue #2 asking to add handling of the ex and em cases. This was done at release 0.99 via the addition of \texttt{\textdimenwithunit}.

Renewed thanks to Ruixi Zhang for analyzing at issue #10 what is at stake into finding dimensions exactly representable both in the bp and mm units. Macros \texttt{\texdimenbothbpm} and \texttt{\texdimenbothmmbp} now address this (release 1.0).