

Contents

texdimens	1
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	2
Macros of this package (full list)	5
Extras?	11
Acknowledgements	11

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: 0.99d 2021/11/04

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:

- `\texdimenpt`,
- `\texdimenUU`, `\texdimenUUup` and `\texdimenUUdown` with UU standing for one of bp, cm, mm, in, pc, cc, nc, dd and nd,
- `\texdimenbothincm` (and relatives not listed here, see below),

- `\texdimenwithunit`.

For example `\texdimenbp` 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 that one admits any representation as $E \text{ bp}$; else it will be either the closest match from above or from below (for this unit the error is at most 1sp).

The variants `\texdimenbpup` and `\texdimenbpdown` allow to choose the direction of approximation.

`\texdimenwithunit{<dimen1>}{<dimen2>}` produces a decimal D such that `\dimexpr dimen2\relax` is parsed by TeX into the same dimension as `dimen1` if this is at all possible. If `dimen2 < 1pt` all TeX dimensions `dimen1` are attainable. If `dimen2 > 1pt` 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>` but please continue reading this documentation for relevant information on how TeX handles dimensions.

Quick review of basics: TeX points and scaled points

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

- `\number\foo` produces the integer N such as `\foo` is the same as `Nsp`,
- inside `\numexpr`, `\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 $N = \text{round}(65536 \cdot D)$ 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 `\dimexpr 1uu\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).

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

uu	phi	reduced	real approximation (Python output)	1uu in sp= [65536phi]	\the<1uu> [65536phi]
bp	7227/7200	803/800	1.00375	65781	1.00374pt
nd	685/642	same	1.0669781931464175	69925	1.06697pt
dd	1238/1157	same	1.070008643042351	70124	1.07pt
mm	7227/2540	same	2.8452755905511813	186467	2.84526pt
pc	12/1	12	12.0	786432	12.0pt
nc	1370/107	same	12.80373831775701	839105	12.80373pt
cc	14856/1157	same	12.84010371650821	841489	12.8401pt
cm	7227/254	same	28.45275590551181	1864679	28.45274pt
in	7227/100	same	72.27	4736286	72.26999pt

The values of `1uu` in the `sp` and `pt` units are irrelevant and even misleading regarding the TeX parsing of `D uu` input. Notice for example that `\the\dimexpr 1bp\relax` gives `1.00374pt` but the actual conversion factor is `1.00375`.

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

Regarding the core units, we always have $\text{phi} > 1$. The increasing sequence $0 \leq \text{trunc}(\text{phi}) \leq \text{trunc}(2\text{phi}) \leq \dots$ is thus *strictly increasing* and, as phi is never astronomically close to 1, **it always has jumps**: not all TeX dimensions can be obtained from an assignment using a core unit distinct from the pt (and sp of course, but we already said it was kept out of the discussion here).

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

The “worst” unit is the largest i.e. the in whose conversion factor is 72.27. The simplest unit to understand is the pc as it corresponds to an integer ratio 12: only dimensions which in scaled points are multiple of 12 are exactly representable in the 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 D which would achieve $1\text{in} = D\text{cm}$: the “step” between attainable dimensions is 72--73 sp for the in and 28--29 sp for the cm , and as 1in differs internally from 2.54 cm by only 12 sp it is impossible to adjust either the in side or the cm side to obtain equality.

In particular $1\text{in} = 2.54\text{cm}$ is **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 $10\text{in} = 254\text{mm}$... It is false though that $1\text{in} = 25.4\text{mm}$!

```
>>> (\dimexpr1in, \dimexpr2.54cm);
@_1      4736286, 4736274
>>> (\dimexpr10in, \dimexpr25.4cm);
@_2      47362867, 47362855
>>> (\dimexpr100in, \dimexpr254cm);
@_3      473628672, 473628672

>>> (\dimexpr1in, \dimexpr25.4mm);
@_4      4736286, 4736285
>>> (\dimexpr10in, \dimexpr254mm);
@_5      47362867, 47362867
```

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

maximal allowed (with 5 places)	the corresponding maximal attainable dim.	minimal TeX dimen denotation causing "Dimension too large"
-----	-----	-----
16383.99999pt	1073741823sp (<code>=\maxdimen</code>)	16383.99999237060546875pt
16322.78954bp	1073741823sp (<code>=\maxdimen</code>)	16322.78954315185546875bp
15355.51532nd	1073741823sp (<code>=\maxdimen</code>)	15355.51532745361328125nd
15312.02584dd	1073741822sp	15312.02584075927734375dd
5758.31742mm	1073741822sp	5758.31742095947265625mm
1365.33333pc	1073741820sp	1365.33333587646484375pc
1279.62627nc	1073741814sp	1279.62627410888671875nc
1276.00215cc	1073741821sp	1276.00215911865234375cc
575.83174cm	1073741822sp	575.83174896240234375cm
226.70540in	1073741768sp	226.70540618896484375in

Perhaps for these various peculiarities with dimensional units, TeX does not provide an output facility for them similar to what `\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 `\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 (or sufficiently close to) `\maxdimen` 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 `\texdimen<uu>up{X}`, if used as `Duu` in a dimension assignment or expression, will (naturally) trigger a “Dimension too large” error.
2. The same potentially happens with `\texdimenwithunit{dimen1}{dimen2}` if `\maxdimen` is not representable exactly by `dimen2` used as a base dimension, (which may happen only if `dimen2 > 1pt`): it is possible that the output `D`, if consequently used as `D\dimexpr dimen2\relax` will trigger “Dimension too large”.
3. For `dd`, `nc` and `in`, and input `X` equal to (or sufficiently close to) `\maxdimen` it turns out that `\texdimen<uu>{X}` produces an output `D` such that `Duu` is the first “virtually attainable” TeX dimension *beyond* `\maxdimen`. Hence

here also Duu will trigger on use “Dimension too large error”.

4. Again for the dd, nc and in units, both the “down” and “up” macros will trigger “Dimension too large” *during their execution* if used with an input equal to (or sufficiently close to) \maxdimen.

`\texdimenpt{<dim. expr.>}`

Does `\the\dimexpr <dim. expr.> \relax` then removes the pt.

`\texdimenbp{<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.

`\texdimenbpdwn{<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.

`\texdimenbpup{<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.

`\texdimennnd{<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.

`\texdimennnddwn{<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.

`\texdimennndup{<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.

`\texdimenddd{<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 “Dimension too large” error. `\maxdimen-1sp` is attainable via `15312.02584dd`.

`\texdimendddown{<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.

`\texdimenddup{<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.

`\texdimenmmm{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that Dmm 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` as input produces on output `5758.31741` and indeed the maximal attainable dimension is `5758.31741mm` (`1073741822sp`).

`\texdimenmmdown{<dim. expr.>}`

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

`\texdimenmmup{<dim. expr.>}`

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

`\texdimenpc{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that Dpc 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).

`\maxdimen` as input produces on output 1365.33333 and indeed the maximal attainable dimension is 1365.33333pc (1073741820sp).

`\texdimenpcdown{<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.

`\texdimenpcup{<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.

`\texdimennnc{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that D_{nc} 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` is 1279.62628 but 1279.62628nc will trigger “Dimension too large” error. `\maxdimen-9sp` is attainable via 1279.62627nc.

`\texdimennncdown{<dim. expr.>}`

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

`\texdimennncup{<dim. expr.>}`

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

`\texdimencc{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that D_{cc} 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` as input produces on output `1276.00215` and indeed the maximal attainable dimension is `1276.00215cc (1073741821sp)`.

`\texdimenccdown{<dim. expr.>}`

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

`\texdimenccup{<dim. expr.>}`

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

`\texdimen cm{<dim. expr.>}`

Produces a decimal (with up to five decimal places) `D` such that `Dcm` 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` as input produces on output `575.83174` and indeed the maximal attainable dimension is `575.83174cm (1073741822sp)`.

`\texdimen cmdown{<dim. expr.>}`

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

`\texdimen cmup{<dim. expr.>}`

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

`\texdimenin{<dim. expr.>}`

Produces a decimal (with up to five decimal places) `D` such that `Din` 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` is `226.70541` but `226.70541in` will trigger “Dimension too large” error. `\maxdimen-55sp` is maximal attainable dimension (via `226.7054in`).

`\texdimenin down{<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.

`\texdimeninup{<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.

`\texdimenbothcmin{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that D_{in} is the largest dimension smaller than the original one and exactly representable both in the `in` and `cm` units.

`\texdimenbothincm{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that D_{cm} is the largest dimension smaller than the original one and exactly representable both in the `in` and `cm` units. It thus represents the same dimension as `\texdimenbothcmin{<dim. expr.>in}`.

`\texdimenbothcminpt{<dim. expr.>}`

Produces a decimal (with up to five decimal places) D such that D_{pt} is the largest dimension smaller than the original one and exactly representable both in the `in` and `cm` units. It thus represents the same dimension as the one provided by `\texdimenbothcmin` and `\texdimenbothincm`.

`\texdimenbothincmpt{<dim. expr.>}`

Same as `\texdimenbothcminpt`.

`\texdimenbothcmisp{<dim. expr.>}`

Produces an integer (explicit digit tokens) N such that N_{sp} is the largest dimension smaller than the original one and exactly representable both in the `in` and `cm` units.

`\texdimenbothincmsp{<dim. expr.>}`

Same as `\texdimenbothcmisp`.

`\texdimenwithunit{<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 `<dim. expr. 1>` if at all possible. If not possible it will be a closest match either from above or below (but one does not know if the other direction is a better or worst match).

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

Extras?

As already stated the “up” and also the “down” macros for the `dd`, `nc` and `in` units will trigger “Dimension too large” if used with inputs equal to or very near `\maxdimen`. “Safe” variants which are guaranteed never to trigger this error but have some extra overhead to filter out inputs very close to `\maxdimen` will *perhaps* be provided if there is some demand for it.

But of course anyhow the output from the “up” macros if used as input with the corresponding unit will be beyond `\maxdimen` if the latter is not attainable, i.e. for all units except `bp`, and `nd` (and `pt` but there is no “up” macro for it).

The dimensions representable with both `in` and `cm` units have the shape $\text{trunc}(3613.5 \cdot k)\text{sp}$ for some integer k . The largest one smaller than a given dimension will thus differ from it by at most about 0.055pt , which is also about 0.02mm .

For example `\texdimenbothincm{1cm}` expands to 0.99994cm which maps internally to 1864566sp which differs from TeX’s 1cm by only -113sp . It can be obtained from 0.39368in or 28.45102pt .

And `\texdimenbothcmin{1in}` expands to 0.99945in , maps internally to 4733685sp which differs from TeX’s 1in by -2601sp . It can be obtained as 2.5386cm or 72.2303pt .

Currently the package does not provide analogous approximations from above. For the 1in for example it would be 4737298sp , i.e. 1.00021in which differs from TeX’s 1in by $+1012\text{sp}$ and is obtained also as 2.54054cm and 72.28543pt .

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 here [issue #2](#) asking to add handling of the `ex`, `em`, and `px` cases. This was done at release 0.99 via the addition of `\texdimenwithunit`.