

# README

xint 1.3d

2019/01/06

<b>Aim</b> . . . . .	<b>p. 1</b>
<b>Usage</b> . . . . .	<b>p. 2</b>
With LaTeX . . . . .	p. 2
With TeX . . . . .	p. 2
<b>Installation</b> . . . . .	<b>p. 2</b>
Method A: using the package manager of your TeX distribution . . . . .	p. 2
Method B: manual installation using <code>xint.tds.zip</code> and <code>unzip</code> . . . . .	p. 3
Method C: manual installation using <code>Makefile</code> and <code>xint.dtx</code> . . . . .	p. 3
Method D: installation starting with only <code>xint.dtx</code> . . . . .	p. 3
<b>License</b> . . . . .	<b>p. 4</b>

Source: `xint.dtx 1.3d 2019/01/06 (doc 2019/01/06)`  
Author: Jean-Francois Burnol  
Info: Expandable operations on big integers, decimals, fractions  
License: LPPL 1.3d

This README is also available as `README.pdf` and `README.html`.

Change log is to be found in `CHANGES.pdf` or `CHANGES.html`.

The user manual is `xint.pdf`, and the commented source code is available as `sourcexint.pdf`.

## Aim

The basic aim is provide *expandable* computations on integers, fractions, and floating point numbers. For example

```
\xinttheexpr reduce(37189719/183618963+11390170/17310720)^17\relax
```

will evaluate exactly the fraction; the result has 462 characters (including the fraction slash.) One can also work with dummy variables:

```
\xinttheexpr mul(add(x(x+1)(x+2), x=y..y+15), y=171286,98762,9296)\relax
```

evaluates to 15979066346135829902328007959448563667099190784.

Float computations are possible at an adjustable precision (default 16).

```
\xintDigits:=48;\xintthefloatexpr 123_456_789^1_000.5\relax  
->3.63692761822782679930738270515740797370813691938e8095
```

(as this example shows the underscore character can be used to separate visually digits, one can also use the space character for that purpose).

Currently, only integer and half-integer exponents are allowed for the power operation in expressions and only the square-root operation is implemented besides the four arithmetic operations. Square-root and the four operations achieve correct rounding in the given arbitrary precision.

## Usage

It is possible to use the package with Plain (using for example `\input xintexpr.sty`) or with LaTeX (`\usepackage{xintexpr}`).

### With LaTeX

```
\usepackage{xint}      % expandable arithmetic with big integers
\usepackage{xintfrac} % decimal numbers, fractions, floats
\usepackage{xinttools} % expandable and non expandable loops
\usepackage{xintexpr} % expressions with infix operators
```

Further packages: `xintbinhex`, `xintgcd`, `xintseries` and `xintcfrac`.

Main dependencies are handled automatically. For example `xintexpr` automatically loads `xinttools` and `xintfrac` (which itself loads `xint`). But the `gcd()/lcm()` functions require extra loading of `xintgcd` and hexadecimal input requires `xintbinhex`.

Package `xintcore` is the subset of `xint` providing only the five operations on big integers: `\xintiiAdd`, `\xintiiMul`, ...

The LaTeX package `bnumexpr` defines a more light-weight parser of arithmetical expressions using big integers, which supports only the four operations, the modulo operation, the power operation, and the factorial. By default it uses the macros from `xintcore` but this can be customized.

### With TeX

One does for example:

```
\input xintexpr.sty
```

The packages may be loaded in any catcode context such that letters, digits, `\` and `%` have their standard catcodes.

`xintcore.sty` and `xinttools.sty` both import `xintkernel.sty` which has the catcode handler and package identifier and defines a few utilities such as `\oodef/\fdef`, `\xint_dothis/\xint_orthat`, or `\xintLength`.

Since 1.3b, `xintkernel.sty` also provides `\xintUniformDeviate` which is a wrapper of the engine `\pdfuniformdeviate` or `\uniformdeviate` done to guarantee more uniformity of the pseudo-random integers. This is used by `xintexpr.sty` for implementing `random()` and `randrange()` functions.

## Installation

### Method A: using the package manager of your TeX distribution

`xint` is included in [TeXLive](#) (hence also [MacTeX](#)) and [MikTeX](#).

There can be a few days of delay between apparition of a new version on [CTAN](#) and availability via the distribution package manager.

## Method B: manual installation using `xint.tds.zip` and `unzip`

Assumes a GNU/Linux-like system (or Mac OS X).

1. obtain `xint.tds.zip` from CTAN: <http://mirror.ctan.org/install/macros/generic/xint.tds.zip>

2. cd to the download repertory and issue:

```
unzip xint.tds.zip -d <TEXMF>
```

where <TEXMF> is a suitable TDS-compliant destination repertory. For example, with TeXLive:

- Linux, standard access rights, hence `sudo` is needed, installation into the “local” tree:

```
sudo unzip xint.tds.zip -d /usr/local/texlive/texmf-local
sudo texhash /usr/local/texlive/texmf-local
```

- Mac OS X, installation into user home folder (no `sudo` needed, and it is recommended to not have a `ls-R` file there, hence no `texhash`):

```
unzip xint.tds.zip -d ~/Library/texmf
```

## Method C: manual installation using `Makefile` and `xint.dtx`

The `Makefile` automatizes rebuilding from `xint.dtx` all documentation files as well as `xint.tds.zip`. It is for GNU/Linux-like (inc. Mac OS X) systems, with a `teTeX` like installation such as TeXLive. The [Latexmk](#) and [Pandoc](#) softwares are required to build all the documentation.

1. obtain `xint.dtx` and `Makefile` from <http://mirror.ctan.org/macros/generic/xint>.
2. put them in an otherwise empty working repertory, run `make` or equivalently `make help` for further instructions.

## Method D: installation starting with only `xint.dtx`

Run `etex xint.dtx` to extract from `xint.dtx` all macro files as well as auxiliary files needed for building the documentation. Among them there is `Makefile.mk`. If you are on a GNU/Linux-type system, rename the file to `Makefile` and execute `make` on command line for further help. If not, you will need to examine the contents of this file to see the commands needed to produce the documentation with `latexmk` (extraction will have created a configuration file `.latexmkrc`) and `pandoc`. If not using `latexmk`, you will need to execute suitable `makeindex -s xint-gind.ist` calls to produce the indices of macros for inclusion into `sourcexint.pdf`. It is also possible to get `xint.pdf` to include the source code. For this, see the instructions in `xint.tex`.

Finishing the installation in a TDS hierarchy:

- move the style files to `TDS:tex/generic/xint/`
- `xint.dtx` goes to `TDS:source/generic/xint/`
- The documentation (`xint.pdf`, `README.md`,...) goes to `TDS:doc/generic/xint/`

Depending on the destination, it may then be necessary to refresh a filename database.

## License

Copyright (C) 2013-2019 by Jean-Francois Burnol

This Work may be distributed and/or modified under the conditions of the LaTeX Project Public License version 1.3c. This version of this license is in

<http://www.latex-project.org/lppl/lppl-1-3c.txt>

and version 1.3 or later is part of all distributions of LaTeX version 2005/12/01 or later.

This Work has the LPPL maintenance status `author-maintained`.

The Author of this Work is Jean-Francois Burnol.

This Work consists of the source file `xint.dtx` and of its derived files: `xintkernel.sty`, `xintcore.sty`, `xint.sty`, `xintfrac.sty`, `xintexpr.sty`, `xintbinhex.sty`, `xintgcd.sty`, `xintseries.sty`, `xintcfrac.sty`, `xinttools.sty`, `xint.ins`, `xint.tex`, `README`, `README.md`, `README.html`, `README.pdf`, `CHANGES.md`, `CHANGES.html`, `CHANGES.pdf`, `pandoctpl.latex`, `doHTMLs.sh`, `doPDFs.sh`, `xint.dvi`, `xint.pdf`, `Makefile.mk`, `xint-gind.ist`, and `.latexmkrc`.