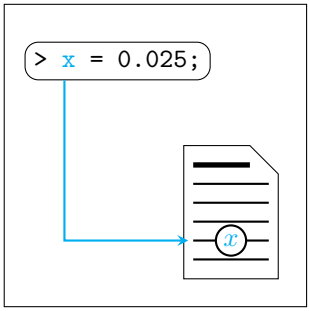
The datax package*

David Gustavsson david.e.gustavsson@gmail.com

November 29, 2020



1 Motivation

datax allows you to export data from your scripts, and import them as literal strings or siunitx commands. If the scripts or the data on which they operate change, the printed data will update as well. This is analogous to how the author uses graphicx to programmatically generate graphics and include in a document.

An alternative is to simply print `\def`s, which scales poorly (but is similar to how this package works under the hood). Or using something like csvread, which seems overkill for individual data points, and can be quite difficult to index into.

datax is intended to work as an extension of your variable name space from scripts to your document: you should simply be able to call your variables by the same name in LaTeX as in your scripts.

2 Installation

If you have a LaTeX package manager, like texlive or miktex, use it to download the package from ctan. Otherwise, download the repository and place `datax.sty` in a place where LaTeX can find it (often `~/texmf/tex/latex/datax/datax.sty`). Run `texhash` if needed.

*This document corresponds to datax v1.1.1, dated 2020/11/29.

Table 1: Outputs from the \datax command

Input	Output
\datax{s}	A literal string
\datax{x}	2.4
\datax{c}	$3 \times 10^8\,\mathrm{m\,s^{-1}}$
\datax{undefined}	??

Table 2: Implemented language plugins

Language	Plugin	Comments
julia	LaTeXDatax.jl	By the present author
Matlab	LaTeXDatax.m	By the present author
Python	LaTeXDatax.py	By the present author

3 Usage

The package is loaded with \usepackage[dataxfile=⟨*data.tex*⟩]{datax}, which reads the file specified as ⟨*dataxfile*⟩. From then on data can be inserted as
\datax \datax{⟨*tag*⟩}. If, for instance, the file data.tex contains references to a string s, a number x and a physical constant c, then the macro produces the output in table 1.

I highly recommend using siunitx (in general, but with datax in particular). It is of course possible to use the literal string function to circumvent siunitx and make datax print numbers without \num and \SI, which is why siunitx is not loaded as a prerequisite, but I have deliberately chosen to use these commands per default when printing numbers and quantities because it looks much better.

4 Interactions

Technically, datax only needs a data file consisting of a number of assignments: \pgfkeyssetvalue{/datax/⟨*tag*⟩}{⟨*value*⟩} but of course the entire point of the package is automation. For this, you need an interaction plugin for your script language. If your language is not listed in table 2, you might need to write this plugin for yourself, or request it.

Installation instructions and usage documentation is available at the respective repositories. In general these plugins are installed using the language's default plugin manager.

Taking julia as an example, the variables in table 1 could have been generated with the script in listing 1.

Listing 1: Example julia script to generate the outputs in table 1

```
#!julia
using LaTeXDatax, Unitful

s = "A literal string";
x = 2.4;
c = 3e8u"m/s";

@datax s x c
```

5 Implementation

Default data file name: `data.tex`

```
\pgfkeys{ /packageoptions/datax file/.initial=data.tex, }
```

Read any given options into family `/packageoptions/`. Then introduce family `/datax/` where all the variables will be stored.

```
\ProcessPgfPackageOptions{/packageoptions}

\pgfkeys{ /datax/.is family, datax, %
        .unknown/.code={ \pgfkeyssetvalue{ %
                         \pgfkeyscurrentpath/\pgfkeyscurrentname %
        }{ #1 } },
}

```

If data file exists, source it (storing all the variables in `/datax/`). Otherwise, throw a warning.

```
\def\dataxfile{./\pgfkeysvalueof{/packageoptions/dataxfile}}
\InputIfFileExists{%
        \dataxfile
        }{}{
        \PackageWarning{datax}{Cannot read file `\dataxfile'}
}
```

`\datax` Include datum. If the supplied tag is unused, print bold question marks (like `\ref`), and throw a warning.

```
\newcommand{\datax}[1]{
        \pgfkeysifdefined{/datax/#1}{ %
                \pgfkeysvalueof{/datax/#1} %
                }{ %
                \PackageWarning{datax}{Data value `#1' undefined}\textbf{??} %
        } %
}
```