jobname-suffix
Compile different content based on the file name

Jason Siefken*

Released September 24, 2022

Contents

Abstract

jobname-suffix allows one to compile a document differently depending on the document's file name (internally called the `jobname` in TEX). This allows one to have one source file and multiple links to this source file (either as symbolic links, or as different files referencing the original via an `\input` command) that are each compile differently. For example, an exam might have an `exam-student.tex` and an `exam-instructor.tex` which both reference the same source code but where the instructor version includes solutions and the student version does not.

1 Introduction

jobname-suffix by default allows you to reference the part of a file name (technically the `\jobname`) that comes after the first "`-`" character and use that information to control what LATEX produces.

Consider the following example: you have three files `exam.tex`, `exam-student.tex`, and `exam-instructor.tex`. The files `exam-student.tex` and `exam-instructor.tex`

---

*Please file an issues/comments to https://github.com/siefkenj/jobname-suffix/issues are either symbolic links to `exam.tex` (if your operating system supports symbolic links) or use `\input{`⟨*exam.tex file path*⟩`}` to include the contents of `exam.tex`.

`exam-student.tex` and *`exam-instructor.tex:`*

```
\input{exam.tex}
```

`exam.tex:`

```
\documentclass{article}
\usepackage{jobname-suffix}

\newcommand{\solution}[1]{#1}
\IfSuffixT[instructor]{
        % Instructors get solutions emboldened
        \renewcommand{\solution}[1]{\textbf{#1}}
}
\IfSuffixT[student]{
        % Students don't see solutions at all
        \renewcommand{\solution}[1]{}
}

\begin{document}
        \begin{enumerate}
                \item First question
                \solution{First answer}

                \item Second question
                \solution{Second answer}
        \end{enumerate}
\end{document}
```

Then, compiling via

```
> lualatex exam-instructor.tex
> lualatex exam-student.tex
```

will produce two pdfs: one with solutions and one without.

2 Usage

2.1 Variables

`\JobnameSuffix`

The computed suffix of the filename (`\jobname`). This is normally the part of the file name after the first “`-`” character and before the file extension. For example, `file-a.tex` has a `\JobnameSuffix` of “`a`”, `file-a-b.tex` has a `\JobnameSuffix` of “`a-b`”, and `file.tex` has a `\JobnameSuffix` of consisting of the empty string.

2.2 Commands

`\IfSuffixTF`
`\IfSuffixT`
`\IfSuffixF`

`\IfSuffixTF[⟨suffixes⟩]{⟨true condition⟩}{⟨false condition⟩}`
`\IfSuffixT[⟨suffixes⟩]{⟨true condition⟩}`
`\IfSuffixF[⟨suffixes⟩]{⟨false condition⟩}`

The argument ⟨*suffixes*⟩ is a comma-separated list of all suffixes you wish to test against.

`\OverrideSuffix`

`\OverrideSuffix{⟨suffix⟩}`

Override `\JobnameSuffix` to be the suffix specified by ⟨*suffix*⟩ instead.

2.3 Environments

`IfSuffix`

The `IfSuffix` environment accepts an optional argument `[`⟨*suffixes*⟩`]` and will display its contents (unmodified) if `\JobnameSuffix` is among the comma-separated list ⟨*suffixes*⟩. Unfortunately this environment cannot contain `verbatim` contents.

3 Compiling

jobname-suffix works by reading the `\jobname` macro that is part of standard TeX. By default, this is set to the file name without the file extension. You can affect the jobname in one of two ways:

1. The name of your file.
2. By supplying the `-jobname` option during compilation. For example, the following two commands would both result in compiling with a `\jobname` of `exam-student`.

```
> lualatex -jobname exam-instructor exam.tex
```

```
> lualatex exam-instructor.tex
```

4 Tips

- For performance reasons, it is best to `\IfSuffixTF` to define/redefine commands in your document preamble rather than to use them in your document body/macro bodies. For example, do

```
\documentclass{article}
\usepackage{jobname-suffix}

\newcommand{\solution}[1]{#1}
\IfSuffixT[instructor]{
        % Instructors get solutions emboldened
        \renewcommand{\solution}[1]{\textbf{#1}}
}

\begin{document}
        \solution{Some Text}
\end{document}
```

instead of

```
\documentclass{article}
\usepackage{jobname-suffix}

\newcommand{\solution}[1]{%
        \IfSuffixTF[instructor]{%
                % Instructors get solutions emboldened
                \textbf{#1}%
        }{%
                #1%
        }
}

\begin{document}
        \solution{Some Text}
\end{document}
```

- To match the case when you have “no suffix” (i.e., when the suffix is the empty string because “`-`” did not appear in the filename), use `\IfSuffix[]` with an empty argument list[1].

5 Installation

For manual installation, the package is available from CTAN.

The package requires LaTeX3 support as provided in the l3kernel and l3packages bundles. Both of these are included in TeX Live and MiKTeX, or are again available in ready-to-install form from CTAN.

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

---

[1] If you omit the optional argument, `\IfSuffixT` will never be true (not even if `\JobnameSuffix` is the empty string).