

**NAME**

**arlatex** – archive a number of ancillary L<sup>A</sup>T<sub>E</sub>X files into a master .tex file

**SYNOPSIS**

**arlatex** [--**outfile**=*filename.tex*] --**document**=*filename.tex filename ...*

**arlatex** --**version**

**arlatex** --**help**

**DESCRIPTION**

**arlatex** is an archiving program like **shar**, **tar**, **zip**, etc. Unlike those other archivers, however, **arlatex** is designed specifically for use with L<sup>A</sup>T<sub>E</sub>X. **arlatex** takes the name of a master .tex file and a number of ancillary files used by that master file (e.g., .tex, .sty, .cls, and .eps files). From these, **arlatex** outputs a single file that, when it's run through **latex**, both regenerates the ancillary files and compiles the document into a .dvi file.

**arlatex** has a few advantages over other archiving programs:

- The .tex files produced by **arlatex** are in a plain-text format. They are therefore perfectly portable and trivial to e-mail to colleagues.
- **arlatex** needs only L<sup>A</sup>T<sub>E</sub>X to run. There is no dependence on any external tools.
- There is no explicit extraction step. As the generated document is run through **latex**, it extracts the ancillary files and builds the document in the same step. The user running **latex** may not even notice that additional files are being produced.

**arlatex** works by writing a number of `\begin{filecontents*} ... \end{filecontents*}` blocks to the output file, followed by the contents of the master file. (In fact, any L<sup>A</sup>T<sub>E</sub>X comments at the beginning of the master file are hoisted to the top of the generated file. This enables the author to draw attention, if so desired, to the fact that ancillary files will be generated.) The `filecontents*` environment, part of standard L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, writes its contents verbatim to a specified file.

**OPTIONS**

--**version**

Output the **arlatex** script's version number.

--**help**

Output brief **arlatex** usage information.

--**document**=*.tex file*

Specify the master document. The output from **arlatex** is this file with all of the other files named on the command line prepended to it. Note that --**document** is a *mandatory* parameter; **arlatex** will abort with an error message if --**document** is not specified.

--**outfile**=*.tex file*

Specify the output file. The output file looks just like the master document, but with a number of `filecontents*` environments preceding the `\documentclass` line. If --**outfile** is not specified, output will be written to the standard output device.

**EXAMPLES**

Suppose you have a paper called *paper.tex* that loads a custom package with `\usepackage{mypackage}`. You want to submit the paper to a conference, but you want to be absolutely certain that *mypackage.sty* doesn't get lost as your paper is shuttled from person to person. Here's how **arlatex** can be of use:

```
arlatex --document=paper.tex mypackage.sty --outfile=paper-submit.tex
```

When *paper-submit.tex* is processed with **latex**, it builds just like the original *paper.tex*, except that it additionally creates a *mypackage.sty* in the current directory:

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
(paper-submit.tex
LaTeX2e <1999/12/01> patch level 1
Babel <v3.6Z> and hyphenation patterns for american, french, german,
ngerman, italian, nohyphenation, loaded.
```

```
LaTeX Warning: Writing file `./mypackage.sty'.
```

```
(/usr/share/texmf/tex/latex/base/article.cls
Document Class: article 1999/09/10 v1.4a Standard LaTeX document class
(/usr/share/texmf/tex/latex/base/size12.clo)) (mypackage.sty)
```

```
.
.
.
```

As another example, here's how you could bundle together all of the files needed to build a large document for longevity:

```
arlatex --document=thesis.tex abstract.tex introduction.tex
background.tex approach.tex experiments.tex relatedwork.tex
conclusions.tex before.eps after.eps fast.eps slow.eps
podunkUthesis.cls --outfile=thesis-all.tex
```

As the number of files to archive together increases it becomes more cumbersome to run **arlatex** manually. Fortunately, using **arlatex** with **bundledoc** is straightforward. **bundledoc** finds all of the files needed to build the document, and **arlatex** combines them into a single file. The following are examples of the bundle: line you might use in a **bundledoc** configuration file:

Unix:

```
bundle: (arlatex --document=$BDBASE.tex $BDINPUTS \
--outfile=$BDBASE-all.tex)
```

Windows:

```
bundle: arlatex --document=%BDBASE%.tex %BDINPUTS% \
--outfile=%BDBASE%-all.tex
```

See the **bundledoc** documentation for more information.

## CAVEATS

**arlatex** makes use of L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>'s `filecontents*` environment. `filecontents*` refuses to overwrite an existing file. However, it also refuses to create a file that exists *anywhere* that L<sup>A</sup>T<sub>E</sub>X can find it. That is, if the user running **latex** on an **arlatex**-generated `.tex` file already has a `/usr/share/texmf/tex/latex/whatever/whatever.sty` file then `filecontents*` will refuse to create a `whatever.sty` file, even in a different directory.

## SEE ALSO

*bundledoc*(1), *latex*(1), *shar*(1), *tar*(1), *zip*(1)

## AUTHOR

Scott Pakin, *scott+bdoc@pakin.org*