

LaTeX 2e/LaTeX 2.09

Subsections

- [LaTeX 2e vs LaTeX 2.09](#)
- [Converting a LaTeX 2.09 document into LaTeX 2e](#)
- [A case study: conversion of a LaTeX 2.09 document into LaTeX 2e with AmS-LaTeX](#)

LaTeX 2e vs LaTeX 2.09

Before *LaTeX 2.09*, fonts were selected via the OFSS (Old Font Selection Scheme). Around the time that *LaTeX 2.09* was released it was realised that a new scheme would need to be devised to manage the vast array of fonts becoming available.

In OFSS, `\it\bf` switched the font to *italic* and then (upright) **boldface**. What if you had wanted a ***bold italic*** font? The idea behind NFSS was to make such influences on the font *orthogonal* so that the sequence `\it\bf` would indeed give a ***bold italic*** font. (That's not quite true ... the effect of such a sequence depends on how it's defined by the particular document class loaded - in many the behaviour is the same as OFSS - *but* `\itfamily\bfseries` and `\bfseries\itfamily` *do* give a ***bold italic*** font.)

Now, *LaTeX 2.09* was released both with OFSS and NFSS ... and along with other influences various incompatible versions of LaTeX were in common use; so users were faced with having files that weren't portable.

LaTeX 2e made NFSS standard and it [*LaTeX 2e*] became the only officially supported version of LaTeX. A **true** *LaTeX 2e* file starts with

```
\documentclass ....
```

but, to provide *compatibility* with *LaTeX 2.09*, when a file starts with

```
\documentstyle ....
```

```
latex emulates LaTeX 2.09.
```

Converting a LaTeX 2.09 document into LaTeX 2e

Most documents written for *LaTeX 2.09* can be converted to *LaTeX 2e* by modifying just the first line of the file. The first line of a *LaTeX 2.09* document is of form:

```
\documentstyle[opt1,...,optm,pkg1,...,pkgn]{docsty}
```

In *LaTeX 2.09* parlance the argument in the curly braces (*docsty*) was called a *document style*. Some possibilities for *docsty* were `article`, `report`, `book`. If say, *docsty* was `article` then *LaTeX 2.09* loaded the file `article.sty`. Between the square brackets were both *options* supported by the *document style*; and *packages* unrelated to the *document style*. Having *pkg1* in the square-bracketed list caused *LaTeX 2.09* to load the file `pkg1.sty`.

The equivalent *LaTeX 2e* commands required to replace the above are:

```
\documentclass[opt1,...,optm] {docsty}
\usepackage{pkg1,...,pkgn}
```

In *LaTeX 2e* parlance the argument in the curly braces is called a *document class*. Like *LaTeX 2.09* some possibilities for the **class** are `article`, `report`, `book`. If the **class** is `article`, say, then *LaTeX 2e* loads the file `article.cls` (so *class* would have been a more appropriate variable name than *docsty*). Between the square brackets go *options* supported by the *document class* and *nothing else*. Any *packages* required are listed between *curly* braces as arguments of `\usepackage`. Having *pkg1* in such a curly-braced list causes *LaTeX 2e* to load the file *pkg1.sty*.

The correct *LaTeX 2e* terminology for a `.sty` file is **package**. Even with *LaTeX 2.09* such files were not *style* files, but as most such files originally created for *LaTeX 2.09* still work with *LaTeX 2e* the *LaTeX* designers decided to retain the `.sty` extension for **package** files.

Sometimes the additional package **latexsym** needs to be included (first) among the list of packages, but the single package **uwamaths** should serve Maths department users for most purposes (it loads the *AmS-LaTeX* package **amssymb** instead of **latexsym**, as well as some other packages).

Some *LaTeX 2.09* documents use old font names like `twlrm` (12 point roman). In such cases, you will need to add

```
\usepackage{rawfonts}
```

to the preamble (along with the changes indicated above) to get a valid *LaTeX 2e* document. See *The rawfonts package* [[rawfonts.dvi](#) | [rawfonts.ps](#)] for more information.

Below, we consider a *LaTeX 2.09* file that was upgraded to *LaTeX 2e* and *AmS-LaTeX*.

You will find some documents cannot be converted to *LaTeX 2e* so simply. Some can be processed in the original form by *LaTeX 2e* ... the `\documentstyle` command signals *LaTeX 2e* to go into *LaTeX 2.09* compatibility mode. Yet other files, though, can only be processed with the old *LaTeX* ... then, you should use the command `latex209` instead of `latex` to process the file.

A case study: conversion of a *LaTeX 2.09* document into *LaTeX 2e* with *AmS-LaTeX*

AmS-LaTeX provides considerable enhancements to basic *LaTeX*. See also [AmS-LaTeX](#).

Preliminary words of caution: You *will* need to do some things differently using *AmS-LaTeX 1.2+* ... but the extra features definitely make it worthwhile. The changes that will cause people the most bother are the plain *TeX* commands:

```
\over, \choose, \atop, \above etc.
```

are **not** allowed. Essentially the *infix* syntax of these commands caused a lot of bother. They are replaced by *prefix* style commands based on the new command `\genfrac` (see *AmS-LaTeX version 1.2 User's Guide* [[amsl.doc.dvi](#) | [amsl.doc.ps](#)] pages 13-15), in particular: `\frac` must be used rather than `\over` and `\binom` must be used rather than `\choose`.

Now we investigate what was needed to convert a typical *LaTeX 2.09* document to *LaTeX 2e* and using *AmS-LaTeX 1.2*. In this case study the user was using *LaTeX 2.09* with **article** style at

11pt and was reading in `bbbold.tex` to get *blackboard bold* fonts (which are **10pt** fonts! ... and so didn't quite look right *particularly* in *headings!!!* ... which tend to be larger than the normal fontsize).

Below are the list of changes that were ``required" to be compatible with the *AmS-LaTeX 1.2+*. The *old* code is always to the left and the *replacement code* to the right (as per the heading). (Some changes you will notice were made to rectify non-ideal use of *LaTeX 2.09*.)

(N)	<i>LaTeX 2.09</i>	<i>AmS-LaTeX 1.2, LaTeX 2e</i>
(0)	<code>\documentstyle[11pt]{article}</code> <code>\input bbbold</code>	<code>\documentclass[11pt]{article}</code> <code>\usepackage{amsmath,amsfonts,amsthm}</code>
	<p><code>bbbold(.tex)</code> is replaced by <code>amsfonts(.sty)</code>. You may choose to use package: amssymb instead ... this loads up package amsfonts as a side-effect. Alternatively you might like to just load the package: uwamaths. Notice several packages can be read in with the one <code>\usepackage</code> command ... so long as the packages without their <code>.sty</code> extensions are separated by commas and there are <i>no</i> blanks, and <i>options</i> if there are any are <i>common</i>. The package: amsmath invokes <i>AmS-LaTeX 1.2+</i>. The package: amsthm is required to make (4) and (5) valid below.</p>	
(1)	<code>\baselineskip=20pt</code>	<code>\renewcommand{\baselinestretch}{1.25}</code>
	<p>Usually put in the <i>preamble</i>, i.e. before <code>\begin{document}</code>. (Redefining <code>\baselinestretch</code> is the <i>LaTeX</i> way of changing interline spacing ... 1.24 to 1.25 corresponds to <i>one-and-a-half</i> spacing.) If using package: uwamaths then you can use the more descriptive: <code>\oneandhalfspacing</code> instead.</p>	
(2)	<code>\pmatrix{.....}</code>	<code>\begin{pmatrix}</code> <code>\end{pmatrix}</code>
	<p>Can use <code>\\</code> in place of <code>\cr</code> if you wish ... but you <i>must not</i> have a <code>\\</code> or <code>\cr</code> just before <code>\end{pmatrix}</code>.</p>	
(3)	<code>\cases{</code>&if \$...\$\cr&if \$...\$\cr <code>}</code>	<code>\begin{cases}</code>&\text{if \$...\$}\\&\text{if \$...\$}\\ <code>\end{cases}</code>
	<p>After the <code>&</code> in <i>LaTeX</i> you are <i>still</i> in math-mode (<i>not</i> text-mode as in <i>plain TeX</i>). So to get text-mode you need to put what comes after <code>&</code> in an <code>\mbox{...}</code> (or <code>\text{...}</code> if using <i>AmS-LaTeX</i>). Note: the <i>AmS-LaTeX</i> command <code>\text</code> is better than <i>LaTeX</i>'s <code>\mbox</code> ... <code>\text</code> is more descriptive and automatically adjusts fontsize, <code>\mbox</code> doesn't! Also note: as with all <i>LaTeX</i> environments, there is <i>no</i> <code>\\</code> or <code>\cr</code> before <code>\end{...}</code>.</p>	
(4)	<code>\proof</code> ... <code>\qed</code>	<code>\begin{proof}</code> ... <code>\end{proof}</code>
	<i>AmS-LaTeX</i> now puts the <code>\qedsymbol</code> flush right ... just as we like it!	
(5)	<code>{\bf Proof of Theorem ...}</code> ... <code>\qed</code>	<code>\begin{proof}[Proof of Theorem ...]</code> ... <code>\end{proof}</code>
	<p>A proof can start with text differing from <code>Proof</code>, by providing the text that you want in square brackets.</p>	

	<code>{ ... \over ... }</code>	<code>\frac{...}{...}</code>
(6)	<p>This is probably the most painful of the changes. Unfortunately, the plain TeX commands <code>\over</code>, <code>\atop</code>, <code>\choose</code>, <code>\above</code> etc. are disabled in <i>AmSLaTeX</i>. All of these can be obtained using <code>\genfrac</code> ... see pages 13-15 of the <i>AmS-LaTeX version 1.2 User's Guide</i> [amsl.doc.dvi amsl.doc.ps] On the plus side, however, you get <code>\dfrac</code> for <code>\displaystyle{\frac{...}{...}}</code> and <code>\tfrac</code> for the corresponding <code>\textstyle</code> version of <code>\frac</code>.</p>	
	<code>{ ... \choose ... }</code>	<code>\binom{...}{...}</code>
(7)	<p>Similar to <code>\over</code>. <i>AmSLaTeX</i> actually uses the <code>\genfrac</code> command to define <code>\binom</code>. Again this is a bit painful, but having <code>\genfrac</code> gives a huge increase in versatility.</p>	
	<code>\def\xxx#1#2#3{...}</code>	<code>\newcommand{\xxx}[3]{...}</code>
(8)	<p>See the <i>LaTeX</i> manual for more examples. It is better (but not essential) to use <code>\newcommand</code> rather than <code>\def</code> because <code>\newcommand</code> tells you if you are redefining something that has already been defined. If you really do want this then you are forced to use <code>\renewcommand</code>; otherwise you then know you should change your macro name to something else.</p>	

[HTML 3.2 Checked!](#)