

# The Student Handouts Package

James Fennell  
jamespfennell@gmail.com

March 15, 2017

The Student Handouts package, `studenthandouts`, is used to generate a single master document that contains a set of individual student handouts.

The package has two main functions.

First, it provides a simple framework for organizing handout source code, and supplies a set of import management tools for selectively importing a subset of the handouts into the master document. Selective import is convenient when compilation of all of the handouts is unnecessary, for example when working on a new handout.

As a secondary feature, the package defines a basic visual style for handouts. This style can be easily changed.

## Contents

<b>1</b>	<b>Basic usage of the package</b>	<b>2</b>
<b>2</b>	<b>Handout import management</b>	<b>3</b>
<b>3</b>	<b>Options and variables</b>	<b>4</b>
3.1	<i>The blanks option</i>	4
3.2	<i>Variables</i>	5
3.2.1	The handouts subdirectory	5
3.2.2	The handouts label	5
3.2.3	The credit/copyright line	6
<b>4</b>	<b>Auxiliary package features</b>	<b>6</b>
4.1	<i>A table of contents for the compiled document</i>	6
4.2	<i>Setting unit titles</i>	6
<b>5</b>	<b>Modifying the handout style</b>	<b>7</b>
5.1	<i>Editing the page layout</i>	7
5.2	<i>Editing the header and footers</i>	7
5.3	<i>Handout output commands</i>	8
	<b>Index of commands</b>	<b>10</b>

[*Student handouts version 1.0; this documentation compiled March 15, 2017.*]

## 1 Basic usage of the package

This package is used as an aid in managing and compiling student handouts. When using this package, the L<sup>A</sup>T<sub>E</sub>X source for the handouts project as a whole is divided between a single master file and individual handout files, one for each handout. This section describes the details of this structure and the basic method for importing handouts into the master document. All of these features can be seen in the sample code that is distributed with the package.

The handouts master file, into which all of the handouts are imported during compilation, is a standard L<sup>A</sup>T<sub>E</sub>X file that includes this package. Place the usual command in the preamble of this file:

```
\usepackage{studenthandouts}
```

For options that may be passed through `usepackage`, see section 3.1 below.

The source for the handouts themselves is stored in a subdirectory of the directory containing the master document. By default, this subdirectory is `./handouts/`. This directory can be changed; see subsection 3.2.1 below.

The package imposes a one-level organizational structure on the handouts. Each handout is a member of a numbered *unit* and within that unit has a unique handout number. Handout 1.1 is the first handout in unit one. The source for handout 1.1 is, by default, stored in

```
./handouts/handout-1-1.tex
```

Handout 3.4 is the fourth handout in unit three and its source is stored in

```
./handouts/handout-3-4.tex
```

The work ‘unit’ is intentionally generic. When working on a specific project units may represent, for example, individual lessons, or chapters in the textbook the handouts are based on.

The organizational structure is loose. For example, there is no requirement that handouts begin at the beginning and proceed sequentially: the first handout can be handout 10.521 if desired.

The source file for a specific handout begins with a `\sethandouttitle` command and is followed by the L<sup>A</sup>T<sub>E</sub>X code for the handout proper. Place this line at the start:

```
\sethandouttitle{<Title of the handout>}
```

If the handout does not have a title, execute the command with an empty argument:

```
\sethandouttitle{}
```

Once there is a source file for handout  $n.m$ , it is imported into the master document through the `\importhandout` command. Place this command in the master document at the point the handout is to appear:

```
\sethandouttitle
```

```
\importhandout
```

```
\importhandout{n}{m}
```

Again,  $n$  is the unit number and  $m$  is the unique number of the handout within the unit.

## 2 Handout import management

After writing a number of handouts, the master document will have many handout import commands. However at a given time it may not be desirable to compile all of the handouts in the project. For example, if working on a new handout it is faster to only compile the new work when error checking. If collaborating, it may be necessary to compile just a certain subset of the handouts to share.

The import management tools are designed for these situations. The basic mechanism is that when handout  $n.m$  is set to be imported the package first checks, based on previous instructions, whether unit  $n$  is to be imported, and if so whether handout  $n.m$  specifically is to be imported. By default all units and handouts are imported.

The instructions for which handouts are to be imported are given through the commands that follow. The instructions may be changed at any point in the master file, even after some of the handouts have already been imported.

- `\importall` `\importall`  
 Import all handouts. This is the default behavior. The `importall` command is used to reset to the default behavior after other instructions have been given.
- `\importnone` `\importnone`  
 Import none of the handouts.
- `\importonlyunits{<unit numbers>}` `\importonlyunits`  
 Import only those units whose unit numbers appear in `<unit numbers>`. The argument `<unit numbers>` is a comma separated list of numbers with no spaces. Example usage:

```
\importonlyunits{1,2,4}
```
- `\importallunits` `\importallunits`  
 Reverse the last command by removing any unit importing restrictions. The difference between this command and `importall` is that if any specific handout restrictions have been imposed then those restrictions still stand.
- `\importonlyhandouts{<handout numbers>}` `\importonlyhandouts`  
 Import only those handouts whose full handout number  $n.m$  appears in `<unit numbers>`. The argument `<unit numbers>` is a comma separated list of handout numbers with no spaces. Example usage:

```
\importonlyhandouts{5.2,6.5,6.6}
```

- `\importallhandouts`

```
\importallhandouts
```

Reverse the last command by removing any specific handout importing restrictions. The difference between this command and `importall` is that if any specific unit restrictions have been imposed then those restrictions still stand.

An important feature is that the import instructions can be changed on the fly. For example with the following code

```
\importnone
\importhandout{1}{1}
\importhandout{1}{2}
\importonlyhandouts{1.3,2.1}
\importhandout{1}{3}
\importhandout{2}{1}
\importhandout{2}{2}
\importhandout{2}{3}
\importall
\importhandout{2}{4}
```

handouts 1.3, 2.1 and 2.4 will be imported.

In particular, in the import context the `importnone` and `importall` commands work similarly to `\begin{comment}` and `\end{comment}` respectively.

## 3 Options and variables

### 3.1 The blanks option

The package has one option: `blanks` or `noblanks`, with `blanks` as default.

The `blanks` option places a blank page after every handout that has an odd number of pages. That way, the compiled handouts document can be printed double sided, and handouts with an odd number of pages will still be on their own sheet.

To see how the `blanks` options achieves this, see Figure 1. In this example there are three handouts: 1.1 with one page, 1.2 with two pages, and 1.3 with one page. It is required that 1.2 be distributed double-sided.

In the first case in Figure 1, the handouts are compiled one after another with the `noblanks` option. If the resulting document is printed at once double-sided, the back of handout 1.1 will contain the first page of handout 1.2, and the next sheet will contain the second page of handout 1.2 with handout 1.3 on the reverse. In order to print the handouts correctly with this PDF file, separate print calls for each of the three handouts are required.

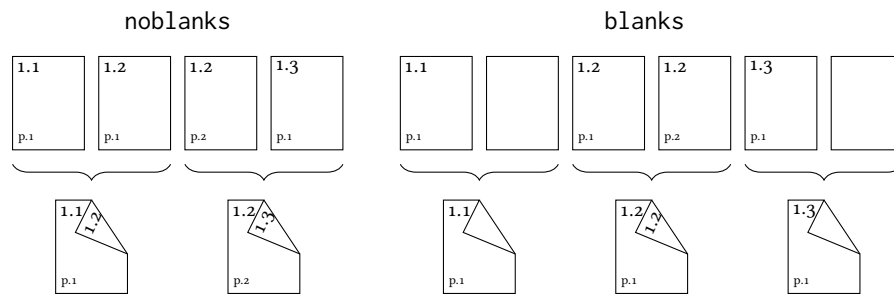


Figure 1: Comparison of the outcome of printing the compiled handouts document double-sided. On the left is the handouts document compiled with the option `noblanks`; on the right is the same document compiled with the option `blanks`.

On the other hand, with the `blanks` option, the resulting PDF document can just be printed double sided at once. The first sheet will contain handout 1.1 on the front, and the blank page after 1.1 on the back. The next sheet will have the two pages of handout 1.2 back-to-back, and the last sheet will have handout 1.3 by itself.

## 3.2 Variables

### 3.2.1 The handouts subdirectory

By default the handouts are stored in the subdirectory `handouts/` of the directory that contains the master document. Change this by redefining `\thehandoutsdirectory`:

```
\renewcommand{\thehandoutsdirectory}{worksheets/}
```

It is necessary to place a forward slash at the end.

If the handouts files are to be stored in the same directory as the master file, set the variable `\thehandoutsdirectory` to be empty:

```
\renewcommand{\thehandoutsdirectory}{}
```

### 3.2.2 The handouts label

By default the package deals with ‘handouts’, and uses that word in the handouts output itself. For example, the default style prints ‘Handout 1.1’ in the header of handout 1.1. A different term like ‘Student worksheet’ might be desired instead. To achieve this, redefine the variable `\thehandoutlabel`:

```
\renewcommand{\thehandoutlabel}{Student worksheet}
```

### 3.2.3 The credit/copyright line

`\thehandoutscredit`

The default handouts style contains a space in left side of the footer where a credit or copyright line can appear. By default it is empty. Set it by redefining the variable `\thehandoutscredit`:

```
\renewcommand{\thehandoutscredit}{NYU Calulus I; Summer 2015}
```

## 4 Auxiliary package features

### 4.1 A table of contents for the compiled document

While each handout has its own page numbering – the first page of a handout is always page one – having a table of contents which references the page numbers of the handouts as they appear in the compiled PDF document can be convenient for determining which pages of the PDF document need to be printed to get certain handouts.

To generate this type of table of contents in the compiled document, place usual command in the master file:

```
\tableofcontents
```

The resulting table of contents will have some custom styling defined by the package.

### 4.2 Setting unit titles

When using this package handouts are organized under generic ‘units’. In a given project these units might represent something concrete; for example, each unit might refer to a specific chapter in a book. In this case individual units might also have titles, like ‘Introduction to Differentiation’, and it may be useful to use these titles in the handouts.

`\setunittitle`

Set the title of a unit with the command `\setunittitle`:

```
\setunittitle{<unit number>}{<unit title>}
```

When changing the handout style, or writing a specific handout, the title of the current unit may be printed with the command `\theunittitle`. If the current unit’s title is not set, `\theunittitle` will do nothing.

In addition, if the title of a unit is set, the title will appear in the compiled document’s table of contents. This will likely make the compiled PDF easier to navigate, and for this reason alone it is worth setting unit titles if they’re available.

## 5 Modifying the handout style

The package defines a basic handouts style. The style is implemented by changing the page margins (using the `geometry` package) and editing the headers and footers of the handouts (using the `fancyhdr` package).

### 5.1 Editing the page layout

When a new handout is loaded the page layout is temporarily changed using the `\thehandoutsgeometry` geometry package. In general handouts do not have large paragraphs of text, so by default the margins are made smaller so that the whole page can be more efficiently used.

When the package needs to change the page layout to that of the handouts it calls the command

```
\thehandoutsgeometry
```

By default, the command `\thehandoutsgeometry` calls the command `newgeometry` (from the `geometry` package) which implements the desired changes. The exact command that the command `\thehandoutsgeometry` executes by default is

```
\newgeometry{top=3cm,left=2cm,right=2cm,bottom=2.5cm}
```

Any desired geometry for the handouts can thus be achieved by setting the command `\thehandoutsgeometry` to execute the `newgeometry` command with appropriate options. The possible options that can be sent to `newgeometry` are extensive and may be discovered in the documentation for the `geometry` package.

For example, to have uniform margins of four centimeters, use the following:

```
\renewcommand{\thehandoutsgeometry}{
  \newgeometry{margins=4cm}
}
```

To make no changes to the geometry – that is, to have the handouts use the same layout as the rest of the document – just set the command to do nothing:

```
\renewcommand{\thehandoutsgeometry}{}
```

### 5.2 Editing the header and footers

The design of the handouts is implemented by editing the headers and footers using the `fancyhdr` package. All of the handouts are within the `studenthandout` fancy page style. This way, the headers and footers of the handouts can be styled without affecting the rest of the document.

To change the default handout style, use the appropriate commands from the `fancyhdr` package. Make the changes to the fancy page style `studenthandout`. For

example, the following code removes the horizontal lines of the headers and footers and removes all of the handout information, except for the handout’s full title which is placed in the center of the header.

```
\fancypagestyle{studenthandout}{
  \renewcommand{\headrulewidth}{0pt}
  \renewcommand{\footrulewidth}{0pt}
  \fancyhf{}
  \fancyhead[C]{\thehandoutfulltitle}
}
```

When changing the handout style the handout output commands in the next section will likely be needed.

### 5.3 Handout output commands

When editing the style there are a number of commands available that print information such as the current handout’s title. None of these commands take an argument. As is commonplace in `LaTeX`, using a command with no argument can result in the whitespace that follows being gobbled up. If this happens use the command with braces after; for example `\thehandoutnumber{}`.

The commands that print information specific to a given handout are as follows.

- `\thehandoutnumber` `\thehandoutnumber`  
 Print the number of the current handout. This is the unique number of the handout within the unit. To print the full handout number, write  

$$\theunitnumber.\thehandoutnumber$$
- `\thehandouttitle` `\thehandouttitle`  
 Print the title of the current handout.
- `\thehandoutfulltitle` `\thehandoutfulltitle`  
 Print the full title of the current handout. This will read something like ‘Handout 3.4: Handout Title’, or, if the handout has no title, simply ‘3.4’.
- `\thehandoutpage` `\thehandoutpage`  
 Print the current handout page number. This is distinct from the document page number as each handout begins at Page 1. To print the document page number use the usual command `\thepage`.



The commands that print information about the current unit or the handouts as a whole are as follows.

- `\theunitnumber` `\theunitnumber`  
Print the number of the current unit.
- `\theunittitle` `\theunittitle`  
Print the title of the current unit, as set through the `setunittitle` command. If the current unit's title has not been set, this command will do nothing.
- `\theunitfulltitle` `\theunitfulltitle`  
Print the full title of the current unit. This will read something like '1: Unit Title'. If the unit's title has not been set the full title will just read '1'.
- `\thehandoutslabel` `\thehandoutslabel`  
Print the handouts label – that is, the word or phrase that says what the handouts are. By default this is 'Handout' but it can be changed; see section 3.2.2 above.
- `\thehandoutscredit` `\thehandoutscredit`  
Print the handouts credit. This is by default blank but may be set to include information like the teacher's name or the institution's name; see section 3.2.3 above.

Finally, the package provides a command `\fullhandoutinfo`

`\fullhandoutinfo`

which outputs a table containing all nine pieces of information from above, as well as the output of the command `thepage`. This command is intended for debugging and is not designed to be used in the final document.

## **Index of commands**

fullhandoutinfo, 9  
importallhandouts, 4  
importallunits, 3  
importall, 3  
importhandout, 2  
importnone, 3  
importonlyhandouts, 3  
importonlyunits, 3  
sethandouttitle, 2  
setunittitle, 6  
thehandoutfulltitle, 8  
thehandoutnumber, 8  
thehandoutpage, 8  
thehandoutscredit, 6, 9  
thehandoutsdirectory, 5  
thehandoutsgeometry, 7  
thehandoutslabel, 5, 9  
thehandouttitle, 8  
theunitfulltitle, 9  
theunitnumber, 9  
theunittitle, 9