

# LaTeX2<sub>ε</sub> Classes for the Journal of Machine Learning Research

Nicola L. C. Talbot

<http://theoval.cmp.uea.ac.uk/~nlct/>

2012-05-16 (version 1.16)

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Required Packages</b>	<b>2</b>
<b>3</b>	<b>Guidelines for Article Authors</b>	<b>3</b>
3.1	Title Information	3
3.2	Font Changing Commands	6
3.3	Structure	6
3.4	Citations and Bibliography	7
3.5	Figures and Tables	8
3.5.1	Sub-Figures and Sub-Tables	9
3.6	Algorithms	10
3.7	Description Lists	11
3.8	Theorems, Lemmas etc	12
3.9	Cross-Referencing	12
3.10	Mathematics	14
3.11	Color vs Grayscale	15
3.12	Where To Go For Help	16
<b>4</b>	<b>Guidelines for Production Editors</b>	<b>17</b>
4.1	jmlrbook Class Options	17
4.2	The Preamble	18
4.3	Main Book Commands	19
4.3.1	Two Column Articles in a One Column Book	23
4.3.2	Cross-Referencing	23
4.4	Altering the Layout of the Main Title Page	24
4.5	Potential Pitfalls	25
4.6	Creating the Book Using makejmlrbook	26
	<b>Index</b>	<b>28</b>

# 1 Introduction

The `jmlr` class is for articles that need to be formatted according to the Journal of Machine Learning Research style. This class is based on the `jmlr2e` and `jmlrwcp2e` packages but has been adapted to enable it to work better with the `combine` class to collate the articles into a book. Section 3 describes how to use the `jmlr` class.

The `jmlrbook` class is for combining JMLR articles into a book. This class uses `combine` and `hyperref`, which are troublesome enough on their own but together are quite fragile. The `jmlrbook` class redefines some internals to get `combine` and `hyperref` to work together but some packages (e.g. `subfig` and `pdfpages`) are likely to mess everything up and cause errors. This is why the guidelines to authors are fairly stringent and why `jmlr` will give an error message if certain packages are loaded.<sup>1</sup> The `jmlrbook` class works best with PDF $\LaTeX$  so authors should ensure that their articles can compile with PDF $\LaTeX$ . Section 4 describes how to use the `jmlrbook` class.

Note that the `jmlr` (and therefore `jmlrbook`) class automatically loads the `hyperref` package, but some packages need to be loaded before `hyperref`.

Anything that needs to be done before `hyperref` is loaded can be specified by defining the command

`\jmlrprehyperref`

`\jmlrprehyperref`

*before* the class is loaded. For example, to load the packages `foo` and `bar` before `hyperref`, you can do:

```
\newcommand{\jmlrprehyperref}{\usepackage{foo,bar}}
\documentclass{jmlr}
```

The `makejmlrbook` Perl script can be used to make a book that uses the `jmlrbook` class. In addition to creating the print and online versions of the book, it will compile the individual articles, running Bib $\TeX$  where necessary, and create a set of HTML files containing a list of all the articles imported into the book along with links to the abstracts and PDFs of the individual articles. Section 4.6 describes how to use the `makejmlrbook` application.

There is also a Java application that can be used as an alternative to `makejmlrbook`, which includes diagnostic tools. See <http://www.dickimaw-books.com/apps/makejmlrbookgui/> for further information.

## 2 Required Packages

The `jmlr` class is based on the `scrartcl` class and loads the following packages: `amsmath`, `amssymb`, `natbib`, `url`, `graphicx` and `algorithm2e`, `hyperref`, `nameref` and `xkeyval`. Note

<sup>1</sup>Currently `jmlr` will check if `subfig`, `pdfpages`, `geometry`, `psfig`, `epsfig` and `theorem` are loaded and will throw an error. If other packages are found to be a problem, they will be added to the list.

that unlike the `jmlr2e` and `jmlrwcp2e` packages, this class file does not load the obsolete `epsfig` package.

The `jmlrbook` class additionally loads the `combine` class and the following packages: `combnat`, `setspace` and `fink`.

The `makejmlrbook` script requires Perl,  $\text{T}_{\text{E}}\text{X}$  and  $\text{T}_{\text{E}}\text{X}4\text{ht}$ .

### 3 Guidelines for Article Authors

Article authors should use the `jmlr` class. This class comes with example files `jmlr-sample.tex` and `jmlrwcp-sample.tex`, which can be used as templates.

The following class options are available:

**nowcp** The article is for the Journal of Machine Learning Research (default).

**wcp** The article is for JMLR Workshop and Conference Proceedings.

**twocolumn** Use two-column style.

**onecolumn** Use one-column style (default).

**color** Color version (see Section 3.11).

**gray** Grayscale version (see Section 3.11).

**tablecaption=top** in a table environment, `\floatconts` puts the caption at the top.

**tablecaption=bottom** in a table environment, `\floatconts` puts the caption at the bottom.

#### 3.1 Title Information

The `jmlr` class uses different syntax from `jmlr2e` and `jmlrwcp2e` to specify the title information. In particular, it doesn't define `\jmlrheading` and `\ShortHeading`. Instead, the following commands should be used:

`\jmlrvolume`

`\jmlrvolume{\<number>}`

This specifies the volume number. For example:

```
\jmlrvolume{2}
```

`\jmlryear`

`\jmlryear{\<year>}`

This specifies the year. For example:

```
\jmlryear{2010}
```

<code>\jmlrsubmitted</code>	<code>\jmlrsubmitted{&lt;date&gt;}</code>
This specifies the submission date.	
<code>\jmlrpublished</code>	<code>\jmlrpublished{&lt;date&gt;}</code>
This specifies the publication date.	
<code>\jmlrworkshop</code>	<code>\jmlrworkshop{&lt;title&gt;}</code>
This specifies the workshop title (for use with the wcp class option). The title information is specified using the commands described below. These commands should typically go in the preamble. As with most class files, The title itself is produced using	
<code>\maketitle</code>	<code>\maketitle</code>
This command should go after <code>\begin{document}</code> . For example: <code>\begin{document}</code> <code>\maketitle</code> Before <code>\maketitle</code> , you must specify the title information using the following commands:	
<code>\title</code>	<code>\title[&lt;short title&gt;]{&lt;title&gt;}</code>
This specifies the article's title. A short title for the page header can be supplied via the optional argument <i>&lt;short title&gt;</i> . If you want to force a line break in the title, use	
<code>\titlebreak</code>	<code>\titlebreak</code>
instead of <code>\newline</code> or <code>\\</code> as this will ensure that the line break doesn't also end up in the table of contents or bookmarks when the article is included in a book.	
<code>\editor</code>	<code>\editor{&lt;name&gt;}</code>
This specifies the editor's name. If there is more than one editor, use:	
<code>\editors</code>	<code>\editors{&lt;names&gt;}</code>
<code>\author</code>	<code>\author{&lt;author specs&gt;}</code>
This specifies the author. The specifications <i>&lt;author specs&gt;</i> are a bit different to <code>jmlr2e</code> and <code>jmlrwcp2e</code> . Use	

`\Name` `\Name [<abbreviated name>] {<author's name>}`

to specify the author's name. Note that if the surname contains a space it must be grouped (enclosed in braces {}). Similarly if the initial letter of each forename is a diacritic it must be grouped. If the abbreviation of the name doesn't get parsed properly you can override the default using the optional argument. (See below for examples.)

`\Email` `\Email{<author's email>}`

This specifies the author's email address. It should only be used within the argument to `\author`.

`\and` `\and`

This should be used to separate two authors with the same address.

`\AND` `\AND`

This should be used to separate authors with different addresses.

`\\` `\\`

This should be used before an author's address or between authors with the same address where there are more than two authors.

`\addr` `\addr`

This should be used at the start of the address.

**Example 1** Two authors with the same address:

```
\author{\Name{Jane Doe} \Email{abc@sample.com}\and
\Name{John {Basey Fisher}} \Email{xyz@sample.com}\\
\addr Address}
```

In this example, the second author has a space in his surname so the surname needs to be grouped.

**Example 2** Three authors with the same address:

```
\author{\Name{Fred Arnold {de la Cour}} \Email{an1@sample.com}\\
\Name{Jack Jones} \Email{an3@sample.com}\\
\Name{{\'}E}louiise {\'}E}abhla Finchley} \Email{an2@sample.com}\\
\addr Address}
```

In this example, the third author has an accent on her forename initials so grouping is required.

**Example 3** Authors with a different address:

```
\author{\Name{John Smith} \Email{abc@sample.com}}\
\addr Address 1
\AND
\Name{May Brown} \Email{xyz@sample.com}}\
\addr Address 2
}
```

**Example 4** The author is actually a company so there's no first name and surname:

```
\author{\Name[Some Company, Ltd]{Some Company, Ltd}\Email{xyz:some.com}}\
\addr Address
}
```

### 3.2 Font Changing Commands

Use the  $\TeX$ 2 $\epsilon$  font changing commands, such as `\bfseries` or `\textbf{<text>}`, rather than the obsolete  $\TeX$ 2.09 commands, such as `\bf`.

`\url` `\url{<address>}`

This will typeset `<address>` in a typewriter font. Special characters, such as `~`, are correctly displayed. Example:

```
\url{http://theoval.cmp.uea.ac.uk/~nlct/}
```

`\mailto` `\mailto{<email address>}`

This will typeset the given email address in a typewriter font. Note that this is not the same as `\Email`, which should only be used in the argument of `\author`.

### 3.3 Structure

`abstract` 

```
\begin{abstract}
<text>
\end{abstract}
```

The abstract text should be displayed using the abstract environment.

`keywords` `\begin{keywords}<keyword list>\end{keywords}`

The keywords should be displayed using the keywords environment.

`\acks` `\acks{<text>}`

This displays the acknowledgements.

`\section`      `\section{<title>}`

Section titles are created using `\section`. The heading is automatically numbered and can be cross-referenced using `\label` and `\ref`. Unnumbered sections can be produced using:

`\section*`      `\section*{<title>}`

`\subsection`      `\subsection{<title>}`

Sub-section titles are created using `\subsection`. Unnumbered sub-sections can be produced using:

`\subsection*`      `\subsection*{<title>}`

`\subsubsection`      `\subsubsection{<title>}`

Sub-sub-section titles are created using `\subsubsection`. Unnumbered sub-sub-sections can be produced using:

`\subsubsection*`      `\subsubsection*{<title>}`

Further sectioning levels can be obtained using `\paragraph` and `\subparagraph`, but these are unnumbered with running heads.

`\appendix`      `\appendix`

Use `\appendix` to switch to the appendices. This changes `\section` to produce an appendix. Example:

```
\appendix
\section{Proof of Theorems}
```

### 3.4 Citations and Bibliography

The `jmlr` class automatically loads `natbib` and sets the bibliography style to `plainnat`. References should be stored in a `.bib` file.

`\bibliography`      `\bibliography{<bib file>}`

This displays the bibliography.

`\citep` `\citep[<pre note>][<post note>]{<label>}`

Use `\citep` for a parenthetical citation.

`\citet` `\citet[<note>]{<label>}`

Use `\citet` for a textual citation.

See the natbib documentation<sup>2</sup> for further details.

### 3.5 Figures and Tables

Floats, such as figures, tables and algorithms, are moving objects and are supposed to float to the nearest convenient location. Please don't force them to go in a particular place. In general it's best to use the `htbp` specifier and don't put the float in the middle of a paragraph (that is, make sure there's a paragraph break above and below the float). Floats are supposed to have a little extra space above and below them to make them stand out from the rest of the text. This extra space is put in automatically and shouldn't need modifying.

To ensure consistency, please *don't* try changing the format of the caption by doing something like:

```
\caption{\textit{A Sample Caption.}}
```

or

```
\caption{\em A Sample Caption.}
```

You can, of course, change the font for individual words or phrases. For example:

```
\caption{A Sample Caption With Some \emph{Emphasized Words}.}
```

The `jmlr` class provides the following command for displaying the contents of a figure or table:

`\floatconts` `\floatconts{<label>}{<caption command>}{<contents>}`

This ensures that the caption is correctly positioned and that the contents are centered. For example:

```
\begin{table}[htbp]
\floatconts
  {tab:example}% label
  {\caption{An Example Table}}% caption command
  {%
    \begin{tabular}{ll}
      \bfseries Dataset & \bfseries Result\\
      Data1 & 0.123456
    \end{tabular}
  }
```

---

<sup>2</sup><http://ctan.org/pkg/natbib>

```

\end{tabular}
}
\end{table}

```

The jmlr class automatically loads graphicx which defines:

`\includegraphics` `\includegraphics[<options>]{<file name>}`

where *<options>* is a comma-separated list of options.

For example, suppose you have an image called `mypic.png` in a subdirectory called `images`:

```

\begin{figure}[htbp]
\floatconts
{fig:example}% label
{\caption{An Example Figure}}% caption command
{\includegraphics[width=0.5\textwidth]{images/mypic}}
\end{figure}

```

Note that you shouldn't specify the file extension when including the image. It's helpful if you can also provide a grayscale version of color images. This should be labeled as the color image but with `-gray` immediately before the extension. (The extension need not be the same as that of the color image.) For example, if you have an image called `mypic.pdf`, the grayscale can be called `mypic-gray.pdf`, `mypic-gray.png` or `mypic-gray.jpg`. See Section 3.11 for further details.

`\includeteximage` `\includeteximage[<options>]{<file name>}`

If your image file is made up of  $\text{\LaTeX}$  code (e.g. `tikz` commands) the file can be included using `\includeteximage`. The optional argument is a key=value comma-separated list where the keys are a subset of those provided by `\includegraphics`. The main keys are: `width`, `height`, `scale` and `angle`.

### 3.5.1 Sub-Figures and Sub-Tables

The `subfig` package causes a problem for `jmlrbook` so the `jmlr` class will give an error if it is used. Therefore the `jmlr` class provides its own commands for including sub-figures and sub-tables.

`\subfigure` `\subfigure[<title>][<valign>]{<contents>}`

This makes a sub-figure where *<contents>* denotes the contents of the sub-figure. This should also include the `\label`. The first optional argument *<title>* indicates a caption for the sub-figure. By default, the sub-figures are aligned at the base. This can be changed with the second optional argument *<valign>*, which may be one of: `t` (top), `c` (centred) or `b` (base).

For example, suppose there are two images files, `mypic1.png` and `mypic2.png`, in the subdirectory `images`. Then they can be included as sub-figures as follows:

```
\begin{figure}[htbp]
\floatconts
{fig:example2}% label for whole figure
{\caption{An Example Figure.}}% caption for whole figure
{%
  \subfigure{%
    \label{fig:pic1}% label for this sub-figure
    \includegraphics{images/mypic1}
  }\quad % space out the images a bit
  \subfigure{%
    \label{fig:pic2}% label for this sub-figure
    \includegraphics{images/mypic2}
  }
}
\end{figure}
```

`\subtable` `\subtable[ $\langle title \rangle$ ][ $\langle valign \rangle$ ]{ $\langle contents \rangle$ }`

This is an analogous command for sub-tables. The default value for  $\langle valign \rangle$  is `t`.

### 3.6 Algorithms

`algorithm`

```
\begin{algorithm}
 $\langle contents \rangle$ 
\end{algorithm}
```

Enumerated textual algorithms can be displayed using the `algorithm` environment. Within this environment, use `\caption` to set the caption (and `\label` to cross-reference it). Within the body of the environment you can use the `enumerate` environment.

`enumerate*`

```
\begin{enumerate*}
\item  $\langle text \rangle$ 
...
\end{enumerate*}
```

If you want to have nested `enumerate` environments but you want to keep the same numbering throughout the algorithm, you can use the `enumerate*` environment, provided by the `jmlr` class. For example:

```
\begin{enumerate*}
\item Set the label of vertex  $s$  to 0
\item Set  $i=0$ 
```

```

\begin{enumerate*}
  \item \label{step:locate}Locate all unlabelled vertices
    adjacent to a vertex labelled  $i$  and label them  $i+1$ 
  \item If vertex  $t$  has been labelled,
    \begin{enumerate*}
      \item[] the shortest path can be found by backtracking, and
        the length is given by the label of  $t$ .
    \end{enumerate*}
  \end{enumerate*}
  otherwise
  \begin{enumerate*}
    \item[] increment  $i$  and return to step~\ref{step:locate}
  \end{enumerate*}
\end{enumerate*}
\end{algorithm}

```

algorithm2e

```

\begin{algorithm2e}
\contents
\end{algorithm2e}

```

Pseudo code can be displayed using the algorithm2e environment, provided by the algorithm2e package, which is automatically loaded. For example:

```

\begin{algorithm2e}
\caption{Computing Net Activation}
\label{alg:net}
\dontprintsemicolon
\linesnumbered
\KwIn{$x_1, \ldots, x_n, w_1, \ldots, w_n$}
\KwOut{$y$, the net activation}
 $y \leftarrow 0$ ;
\For{$i \leftarrow 1$ \KwTo $n$}{
   $y \leftarrow y + w_i x_i$ ;
}
\end{algorithm2e}

```

See the algorithm2e documentation<sup>3</sup> for more details.

### 3.7 Description Lists

altdescription

```

\begin{altdescription}{\langle widest label \rangle}
\item[\langle label \rangle] \langle item text \rangle
\end{altdescription}

```

<sup>3</sup><http://ctan.org/pkg/algorithm2e>

In addition to the standard description environment, the `jmlr` class also provides the `altdescription` environment. This has an argument that should be the widest label used in the list. For example:

```
\begin{altdescription}{differentiate}
\item[add] A method that adds two variables.
\item[differentiate] A method that differentiates a function.
\end{altdescription}
```

### 3.8 Theorems, Lemmas etc

The `jmlr` class provides the following theorem-like environments: `theorem`, `example`, `lemma`, `proposition`, `remark`, `corollary`, `definition`, `conjecture` and `axiom`. Within the body of those environments, you can use the `proof` environment to display the proof if need be. The theorem-like environments all take an optional argument, which gives the environment a title. For example:

```
\begin{theorem}[An Example Theorem]
\label{thm:example}
This is the theorem.
\begin{proof}
This is the proof.
\end{proof}
\end{theorem}
```

### 3.9 Cross-Referencing

Always use `\label` when cross-referencing, rather than writing the number explicitly. The `jmlr` class provides some convenience commands to assist referencing. These commands, described below, can all take a comma-separated list of labels.

`\sectionref`

`\sectionref{<label list>}`

Used to refer to a section or sections. For example, if you defined a section as follows:

```
\section{Results}\label{sec:results}
```

you can refer to it as follows:

```
The results are detailed in \sectionref{sec:results}.
```

This command may also be used for sub-sections and sub-sub-sections.

`\appendixref`

`\appendixref{<label list>}`

Used to refer to an appendix or multiple appendices.

`\equationref`

`\equationref{<label list>}`

Used to refer to an equation or multiple equations.

`\tableref`     `\tableref{\langle label list \rangle}`

Used to refer to a table or multiple tables. This can also be used for sub-tables where the main table number is also required.

`\subtabref`     `\subtabref{\langle label list \rangle}`

Used to refer to sub-tables without the main table number, e.g. (a) or (b).

`\figureref`     `\figureref{\langle label list \rangle}`

Used to refer to a figure or multiple figures. This can also be used for sub-figures where the main figure number is also required, e.g. 2(a) or 4(b).

`\subfigref`     `\subfigref{\langle label list \rangle}`

Used to refer to sub-figures without the main figure number, e.g. (a) or (b).

`\algorithmref`     `\algorithmref{\langle label list \rangle}`

Used to refer to an algorithm or multiple algorithms.

`\theoremref`     `\theoremref{\langle label list \rangle}`

Used to refer to a theorem or multiple theorems.

`\lemmaref`     `\lemmaref{\langle label list \rangle}`

Used to refer to a lemma or multiple lemmas.

`\remarkref`     `\remarkref{\langle label list \rangle}`

Used to refer to a remark or multiple remarks.

`\corollaryref`     `\corollaryref{\langle label list \rangle}`

Used to refer to a corollary or multiple corollaries.

`\definitionref`     `\definitionref{\langle label list \rangle}`

Used to refer to a definition or multiple definitions.

`\conjectureref`

`\conjectureref{<label list>}`

Used to refer to a conjecture or multiple conjectures.

`\axiomref`

`\axiomref{<label list>}`

Used to refer to an axiom or multiple axioms.

`\exempleref`

`\exempleref{<label list>}`

Used to refer to an example or multiple examples.

### 3.10 Mathematics

The `jmlr` class loads the `amsmath` package so you can use any of the commands and environments defined in that package. A brief summary of some of the more common commands and environments is provided here. See the `amsmath` documentation<sup>4</sup> for further details.

`\set`

`\set{<text>}`

In addition to the commands provided by `amsmath`, the `jmlr` class also provides the `\set` command which can be used to typeset a set. For example:

The universal set is denoted  $\set{U}$

Unnumbered single-line equations should be displayed using `\[` and `\]`. For example:

$$[E = mc^2]$$

Numbered single-line equations should be displayed using the `equation` environment. For example:

```
\begin{equation}\label{eq:trigrule}
\cos^2\theta + \sin^2\theta \equiv 1
\end{equation}
```

Multi-lined numbered equations should be displayed using the `align` environment. For example:

```
\begin{align}
f(x) &= x^2 + x \label{eq:f} \\
f'(x) &= 2x + 1 \label{eq:df}
\end{align}
```

---

<sup>4</sup><http://ctan.org/pkg/amsmath>

Unnumbered multi-lined equations should be displayed using the `align*` environment. For example:

```
\begin{align*}
f(x) &= (x+1)(x-1)\\
&= x^2 - 1
\end{align*}
```

If you want to mix numbered with unnumbered lines use the `align` environment and suppress unwanted line numbers with `\nonumber`. For example:

```
\begin{align}
y &= x^2 + 3x - 2x + 1\nonumber\\
&= x^2 + x + 1\label{eq:y}
\end{align}
```

An equation that is too long to fit on a single line can be displayed using the `split` environment.

Text can be embedded in an equation using `\text{<text>}` or you can use `\intertext{<text>}` to interrupt a multi-line environment such as `align`.

Predefined operator names are listed in [table 1](#). For additional operators, either use

`\operatorname`

```
\operatorname{<name>}
```

for example

```
If $X$ and $Y$ are independent,
$\operatorname{var}(X+Y) =
\operatorname{var}(X) + \operatorname{var}(Y)$
```

or declare it with

`\DeclareMathOperator`

```
\DeclareMathOperator{<command>}{<name>}
```

for example

```
\DeclareMathOperator{var}{var}
```

and then use this new command:

```
If $X$ and $Y$ are independent,
$\var(X+Y) = \var(X)+\var(Y)$
```

If you want limits that go above and below the operator (like `\sum`) use the starred versions (`\operatorname*` or `\DeclareMathOperator*`).

### 3.11 Color vs Grayscale

It's helpful if authors supply grayscale versions of their articles in the event that the article is to be incorporated into a black and white printed book. With external PDF, PNG

Table 1: Predefined Operator Names (taken from amsmath documentation)

<code>\arccos</code>	<code>arccos</code>	<code>\deg</code>	<code>deg</code>	<code>\lg</code>	<code>lg</code>	<code>\projlim</code>	<code>projlim</code>
<code>\arcsin</code>	<code>arcsin</code>	<code>\det</code>	<code>det</code>	<code>\lim</code>	<code>lim</code>	<code>\sec</code>	<code>sec</code>
<code>\arctan</code>	<code>arctan</code>	<code>\dim</code>	<code>dim</code>	<code>\liminf</code>	<code>lim inf</code>	<code>\sin</code>	<code>sin</code>
<code>\arg</code>	<code>arg</code>	<code>\exp</code>	<code>exp</code>	<code>\limsup</code>	<code>lim sup</code>	<code>\sinh</code>	<code>sinh</code>
<code>\cos</code>	<code>cos</code>	<code>\gcd</code>	<code>gcd</code>	<code>\ln</code>	<code>ln</code>	<code>\sup</code>	<code>sup</code>
<code>\cosh</code>	<code>cosh</code>	<code>\hom</code>	<code>hom</code>	<code>\log</code>	<code>log</code>	<code>\tan</code>	<code>tan</code>
<code>\cot</code>	<code>cot</code>	<code>\inf</code>	<code>inf</code>	<code>\max</code>	<code>max</code>	<code>\tanh</code>	<code>tanh</code>
<code>\coth</code>	<code>coth</code>	<code>\injlim</code>	<code>injlim</code>	<code>\min</code>	<code>min</code>		
<code>\csc</code>	<code>csc</code>	<code>\ker</code>	<code>ker</code>	<code>\Pr</code>	<code>Pr</code>		
		<code>\varlimsup</code>	$\varlimsup$	<code>\varinjlim</code>	$\varinjlim$		
		<code>\varliminf</code>	$\varliminf$	<code>\varprojlim</code>	$\varprojlim$		

or JPG graphic files, you just need to supply a grayscale version of the file. For example, if the file is called `myimage.png`, then the gray version should be `myimage-gray.png` or `myimage-gray.pdf` or `myimage-gray.jpg`. You don't need to modify your code. The `jmlr` class checks for the existence of the grayscale version if it is print mode (provided you have used `\includegraphics` and haven't specified the file extension).

`\ifprint` `\ifprint{<true part>}{<false part>}`

You can use `\ifprint` to determine which mode you are in. For example:

```
in \figureref{fig:nodes}, the
\ifprint{dark gray}{purple}
ellipse represents an input and the
\ifprint{light gray}{yellow} ellipse
represents an output.
```

Another example:

```
{\ifprint{\bfseries}{\color{red}}important text!}
```

You can use the class option `gray` to see how the document will appear in gray scale mode.

The `xcolor` class is loaded with the `x11names` option, so you can use any of the `x11` predefined colors (listed in the `xcolor` documentation<sup>5</sup>).

### 3.12 Where To Go For Help

If you have a  $\text{\LaTeX}$  query, the first place to go to is the UK TUG FAQ<sup>6</sup>.

<sup>5</sup><http://ctan.org/pkg/xcolor>

<sup>6</sup><http://www.tex.ac.uk/faq>

If you are unfamiliar or just getting started with  $\text{\LaTeX}$ , there's a list of on-line introductions to  $\text{\LaTeX}$  at: <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=man-latex>

There are also forums, mailing lists and newsgroups. For example, the  $\text{\LaTeX}$  Community (<http://www.latex-community.org/>), the texhax mailing list (<http://tug.org/mailman/listinfo/texhax>) and comp.text.tex (archives available at <http://groups.google.com/group/comp.text.tex/>).

Documentation for packages or classes can be found using the texdoc application. For example:

```
texdoc natbib
```

Alternatively, you can go to <http://www.ctan.org/pkg/<name>> where *<name>* is the name of the package. For example: <http://www.ctan.org/pkg/natbib>

For a general guide to preparing papers (regardless of whether you are using  $\text{\LaTeX}$  or a word processor), see Kate L. Turabian, "A manual for writers of term papers, theses, and dissertations", The University of Chicago Press, 1996.

## 4 Guidelines for Production Editors

The jmlrbook class can be used to combine articles that use the jmlr document class into a book. The following sample files are provided: paper1/paper1.tex, paper2/paper2.tex, paper3/paper3.tex, jmlr-sample.tex, jmlrwcp-sample.tex, jmlrbook-sample.tex and proceedings-sample.tex. All but the last two are articles using the jmlr class. The last two (jmlrbook-sample.tex and proceedings-sample.tex) uses the jmlrbook class file to combine the articles into a book. Note that no modifications are needed to the files using the jmlr class when they are imported into the book. They can either be compiled as stand-alone articles or with the entire book.

Before you compile the book, make sure that all the articles compile as stand-alone documents (and run Bib $\text{\TeX}$  where necessary). You can use the makejmlrbook Perl script to compile the book and create associated HTML files. See Section 4.6 for details.

### 4.1 jmlrbook Class Options

**nowcp** The imported pre-published articles were published in the Journal of Machine Learning Research (default).

**wcp** The imported pre-published articles were published in the JMLR Workshop and Conference Proceedings.

If the book has a mixture of JMLR and JMLR WCP articles, you can switch between them using

`\jmlrwcp`

`\jmlrwcp`

and

`\jmlrnowcp`

`\jmlrnowcp`

Alternatively, you can set the name of the journal or conference proceedings using:

`\jmlrproceedings`

`\jmlrproceedings{<short title>}{<long title>}`

**color** Color version (see Section 3.11). Use this option for the on-line version with hyperlinks enabled (default).

**gray** Grayscale version (see Section 3.11). Use this option for the print version without hyperlinks.

**tablecaption=top** in a table environment, `\floatconts` puts the caption at the top.

**tablecaption=bottom** in a table environment, `\floatconts` puts the caption at the bottom.

**letterpaper** Set the paper size to letter (default).

**7x10** Set the paper size to 7 × 10 inches.

**10pt** Use 10pt as the normal text size.

**11pt** Use 11pt as the normal text size (default).

**12pt** Use 12pt as the normal text size.

## 4.2 The Preamble

Any packages that the imported articles load (which aren't automatically loaded by `jmlr`) must be loaded in the book's preamble. For example, if one or more of the articles load the `siunitx` package, this package must be loaded in the book.

Commands that are defined in the imported articles will be local to that article unless they have been globally defined using `\gdef` or `\global`. Since most authors use `\newcommand` and `\newenvironment` (or `\renewcommand` and `\renewenvironment`) this shouldn't cause a conflict if more than one article has defined the same command or environment. For example, in the sample files supplied, both `paper1/paper1.tex` and `paper2/paper2.tex` have defined the command `\samplecommand` using `\newcommand`. As long as this command isn't also defined in the book, there won't be a conflict.

`\title`

`\title[<PDF title>]{<book title>}`

In the book preamble, `\title` sets the book title and the optional argument is used for the PDF title, which will be displayed when the reader views the PDF file's properties in their PDF viewer. (Note that in the imported articles, `\title` sets the article's title and the optional argument sets the short title for the page header and table of contents.)

`\author` `\author[<PDF author(s)>]{<book author(s)>}`

In the book preamble, `\author` sets the book's author (or editor) and the optional argument is used for the PDF author, which will be displayed when the reader views the PDF file's properties in their PDF viewer. (Note that in the imported articles, `\author` sets the article's author and the optional argument sets the short author list for the page header.)

`\volume` `\volume{<number>}`

This command sets the book's volume number. Omit if the book has no volume number.

`\subtitle` `\subtitle{<sub-title>}`

This command sets the book's subtitle. Omit if the book has no sub-title.

`\logo` `\logo{<image command>}`

This sets the book's title image. Use `\includegraphics` and omit the file extension. If you provide a grayscale version as well as a color version, the grayscale version will be used for the print version of the book. (See Section 3.11 for further details.)

`\team` `\team{<team title>}`

This can be used to set the name of the editorial team. This command may be omitted if not required.

`\productioneditor` `\productioneditor{<name>}`

This command may be used to name the production editor. The command may be omitted if not required.

See Section 4.4 for details on how to modify the layout of the title page.

### 4.3 Main Book Commands

All commands that are provided by the `jmlr` class are also available with the `jmlrbook` class, but some commands might behave differently depending on whether they are in the main part of the book or within the imported articles.

In the main part of the book you can use the following commands:

`\maketitle` `\maketitle`

This displays the book's title page. Note that `\maketitle` has a different effect when used in imported articles.

<code>\frontmatter</code>	<div><code>\frontmatter</code></div> <p>Use this command at the start of the front matter (e.g. before the foreword or preface). This will make chapters unnumbered even if you use <code>\chapter</code> instead of <code>\chapter*</code>. It also sets the page style and sets the page numbering to lower case Roman numerals.</p>
<code>authorsignoff</code>	<div> <code>\begin{authorsignoff}</code>  <code>&lt;author list&gt;</code>  <code>\end{authorsignoff}</code> </div> <p>This environment may be used by the author signing off at the end of a chapter such as the foreword. Within the environment use:</p>
<code>\Author</code>	<div><code>\Author{&lt;details&gt;}</code></div> <p>for the author's details. More than one <code>\Author</code> should be used if there is more than one author. Example:</p> <pre> \begin{authorsignoff} \Author{Nicola Talbot\\ University of East Anglia} \Author{Anne Author\\ University of No Where} \end{authorsignoff} </pre>
<code>preface</code>	<div><code>\begin{preface}[&lt;filename&gt;]</code></div> <p>This environment may be used to typeset the preface. This starts a new chapter using <code>\chapter{&lt;prefacename&gt;}</code></p>
<code>\prefacename</code>	<p>where <code>\prefacename</code> defaults to “Preface”. This environment should typically go in the front matter and is provided to allow <code>makejmlrbook</code> create a standalone document for the preface. The optional argument is the filename (without any extension or path) that will be used by <code>makejmlrbook</code>. This defaults to <code>preface</code> but, to conform with JMLR guidelines, should be changed to the surname of the first author (editor) followed by the final two digits of the year. See the JMLR website for further details of the guidelines.</p>
<code>signoff</code>	<div> <code>\begin{signoff}[&lt;team name&gt;]{&lt;date&gt;}</code>  <code>&lt;editor list&gt;</code>  <code>\end{signoff}</code> </div> <p>This environment may be used by the editorial team when signing off a chapter such as the preface. If the optional argument is omitted, “The Editorial Team” is used. If you</p>

are using the preface environment described above, the signoff environment must go inside the preface environment.

Within the signoff environment use:

`\Editor` `\Editor{<details>}`

for each editor. Example:

```
\begin{signoff}{March 2010}
% First editor:
\Editor{Nicola Talbot\\
University of East Anglia\\
\mailto{N.Talbot@uea.ac.uk}}
% Second editor:
\Editor{Anne Editor\\
University of Nowhere\\
\mailto{ae@sample.com}}
\end{signoff}
```

`\tableofcontents` `\tableofcontents`

This command displays the book's table of contents. Note that it has a different effect if used in an imported article.

`\mainmatter` `\mainmatter`

Use this command to switch to the book's main matter. This will switch the chapter numbering back on, reset the page numbering to Arabic and set up the main page style.

`\part` `\part [<short title>]{<title>}`

If used in the main part of the book, this command will start a new part and issue a clear double page. Note that this command has a different effect if used in an imported article.

`\addtocpart` `\addtocpart{<title>}`

This adds <title> to the table of contents, issues a clear double page, but doesn't display any text or affect the part numbering.

`\chapter` `\chapter [<short title>]{<title>}`

This command may be used in the main body of the book but will cause an error if used within an imported article.

<code>\section</code>	<code>\section[<i>&lt;short title&gt;</i>]{<i>&lt;title&gt;</i>}</code>
<code>\subsection</code>	<code>\subsection[<i>&lt;short title&gt;</i>]{<i>&lt;title&gt;</i>}</code>
<code>\subsubsection</code>	<code>\subsubsection[<i>&lt;short title&gt;</i>]{<i>&lt;title&gt;</i>}</code>
<code>\paragraph</code>	<code>\paragraph[<i>&lt;short title&gt;</i>]{<i>&lt;title&gt;</i>}</code>
<code>\subparagraph</code>	<code>\subparagraph[<i>&lt;short title&gt;</i>]{<i>&lt;title&gt;</i>}</code>
<p>These commands may be used in the main body of the book or within imported articles. In the main body of the book they need to be within a chapter and will be numbered according to the chapter.</p>	
<code>\appendix</code>	<code>\appendix</code>
<p>If used in the main body of the book, this will switch to the book appendices. Subsequent <code>\chapter</code> commands will produce the appendices. If used within an imported article, it will switch to the article appendices and won't affect the main part of the book.</p>	
<code>jmlrpapers</code>	<code>\begin{jmlrpapers}</code> <i>&lt;imported papers&gt;</i> <code>\end{jmlrpapers}</code>
<p>This environment must be used when importing articles. Within this environment, use the following commands to import articles:</p>	
<code>\importpubpaper</code>	<code>\importpubpaper[<i>&lt;label&gt;</i>]{<i>&lt;directory&gt;</i>}{<i>&lt;file&gt;</i>}{<i>&lt;pages&gt;</i>}</code>
<p>This imports an article that has already been published elsewhere. The <i>&lt;pages&gt;</i> argument should be the page range from the <i>previously published</i> version of this article. This may not necessarily be the same as the page range of the article in the book. The directory the imported file is contained in is given by <i>&lt;directory&gt;</i>. If the file is in the same directory as the book, use a dot. The file name is given by <i>&lt;file&gt;</i>. The article is also given a label, specified by the optional argument. This is <i>&lt;directory&gt;/&lt;file&gt;</i> by default. The label is used as a prefix to labels in the imported articles which ensures that cross-references are unique. You can also use this label to reference the article elsewhere in the book (see Section 4.3.2).</p>	

`\importpaper` `\importpaper[⟨label⟩]{⟨directory⟩}{⟨file⟩}`

Imports an article that is being published in the book. The arguments are the same as above except that there is no page range (the page range is computed automatically).

`\importarticle` `\importarticle[⟨label⟩]{⟨directory⟩}{⟨file⟩}`

This imports an article that hasn't been published elsewhere. There is no page range, but the other arguments are the same as those describe above for `\importpubpaper`.

Example: to import a previously published paper `paper1/paper1.tex` and an unpublished paper `paper2/paper2.tex`:

```
\begin{jmlrpapers}
\importpubpaper{paper1}{paper1}{23--45}
\importarticle{paper2}{paper2}
\end{jmlrpapers}
```

#### 4.3.1 Two Column Articles in a One Column Book

The `jmlrbook` class column style will override the column style of the imported articles. You can use the `twocolumn` class option to `jmlrbook`, but this will make the whole book with two columns. If you only want the imported articles to be in two columns, then put `\twocolumn` in the `jmlrpapers` environment to switch on two column formatting. The effect will be localised to the end of the environment.

#### 4.3.2 Cross-Referencing

You can cross-reference other parts of the book using the standard `\label/\ref` mechanism, but if you want to reference something within an imported article, you must prefix the label with the label given when importing the article (that is, the optional argument to `\importpubpaper`, `\importpaper` or `\importarticle`). For example, if you want to reference a section labeled `sec:results` in the imported paper `paper1/paper1.tex`, you would need to do:

```
see Section~\ref{paper1/paper1sec:results}
```

or

```
see \sectionref{paper1/paper1sec:results}
```

In addition to the commands described in Section 3.9, the `jmlrbook` class also provides the following cross-referencing commands:

`\chapterref` `\chapterref{⟨label list⟩}`

Reference a chapter or chapters. The argument is a comma-separated list of labels.

`\articlepageref` `\articlepageref{⟨label⟩}`

This displays the starting page number of the article whose label is given by  $\langle label \rangle$ . Note that this must be a single label, not a list. For example:

An interesting article starts on page~\articlepageref{paper1/paper1}

\articlepagesref

\articlepagesref{ $\langle label \rangle$ }

This displays the page range of the article whose label is given by  $\langle label \rangle$ . Again, this must be a single label, not a list. This page range is unrelated to the  $\langle pages \rangle$  argument of \importpubarticle.

\articletitleref

\articletitleref{ $\langle label \rangle$ }

This displays the short title for the article whose label is given by  $\langle label \rangle$ . Again, this must be a single label, not a list.

\articleauthorref

\articleauthorref{ $\langle label \rangle$ }

This displays the author list for the article whose label is given by  $\langle label \rangle$ . Again, this must be a single label, not a list.

#### 4.4 Altering the Layout of the Main Title Page

\titlebody

\titlebody

The main body of the book's title page is given by the command \titlebody. Within the definition of this command, you can use:

\SetTitleElement

\SetTitleElement{ $\langle element \rangle$ }{ $\langle pre \rangle$ }{ $\langle post \rangle$ }

where  $\langle element \rangle$  can be: title, volume, issue<sup>7</sup>, subtitle, logo, team, author, date, productioneditor. The  $\langle pre \rangle$  and  $\langle post \rangle$  arguments specify what to do before and after the element. Note that \SetTitleElement does nothing if that element hasn't been set. For example, if \volume has been omitted or \volume{} is used, then

\SetTitleElement{volume}{\mainvolume font}{\postmainvolume}

will do nothing (so you don't end up with **Volume** :).

\IfTitleElement

\IfTitleElement{ $\langle element \rangle$ }{ $\langle true part \rangle$ }{ $\langle false part \rangle$ }

This does  $\langle true part \rangle$  if  $\langle element \rangle$  has been set otherwise it does  $\langle false part \rangle$ . For example, \postmainvolume is defined as:

<sup>7</sup>The default title page layout doesn't use issue, but if required it can be set with \issue{ $\langle number \rangle$ }

```

\newcommand{\postmainvolume}{%
  \IfTitleElement{subtitle}{:}{:}\par\relax
}

```

This means that it will only print a colon after the volume number if the subtitle has been set.

The default definition of `\titlebody` is:

```

\newcommand{\titlebody}{%
  \SetTitleElement{title}{\maintitlefont}{\postmaintitle}%
  \SetTitleElement{volume}{\mainvolume}{\postmainvolume}%
  \SetTitleElement{subtitle}{\mainsubtitlefont}{\postmainsubtitle}%
  \SetTitleElement{logo}{\mainlogofont}{\postmainlogo}%
  \SetTitleElement{team}{\mainteamfont}{\postmainteam}%
  \SetTitleElement{author}{\mainauthorfont}{\postmainauthor}%
  \SetTitleElement{productioneditor}{\mainproductioneditorfont}{%
    {\postmainproductioneditor}%
  }
}

```

## 4.5 Potential Pitfalls

The combine class and hyperref package are individually both easily broken by packages that change certain internals and they don't ordinarily work together. The jmlr-book class applies patches to the internal referencing mechanism to make them work together, but it's a fairly fragile alliance. Some packages are known to break it, for example subfig, pdfpages and geometry. This is why the jmlr class checks for known problem packages and generates an error message to dissuade authors from using them. It's likely that there are other packages that may cause a problem and, as they are found, they will be added to the check list. Also, it's possible for an author to disable the package checking mechanism if they are determined to use a particular package.

In the event that an article has loaded a problem package, the editors will have to decide whether to ask the author to change the article so that it doesn't cause a problem or to make the changes themselves or to find a way of fudging things to get it to work. It depends on the level of  $\text{\LaTeX}$  expertise amongst the editors and the time available.

Another problem that can arise is when different articles use packages that conflict. For example, one article uses package foo and another uses package bar. Each article compiles okay as a stand-alone article, but when combined foo and bar conflict. Another problem may occur when articles load the same package but with conflicting package options. To reduce the chance of this occurring, the jmlr class loads some commonly used packages. For example, it loads the algorithm2e package with the algo2e and ruled options and provides the algorithm environment in addition to algorithm2e's algorithm2e environment. Different versions of the same package can also be a problem. To help counteract the problem caused by different papers using different versions of the algorithm2e package, jmlrbook defines most of the old style commands if they don't exist.

Articles that use different input encodings can also cause a problem. For example, if one article uses utf8 and another uses latin1. If the authors have directly entered a diacritic or ligature, such as é or æ, instead of using a  $\text{\LaTeX}$  command, such as `\'e`

or `\ae`, then this will cause an error on compiling the book.<sup>8</sup> The choice then is to either change all non-keyboard characters with the appropriate  $\TeX$  commands or to use the `\inputencoding` command, supplied by the `inputenc` package, to switch the encoding at the start of each article.

Authors who use `\nonumber` within an equation environment can mess up the hyperlinks. Remove `\nonumber` and change the equation environment to `\[ ... \]` (or just make it a numbered equation).

If the article changes the graphics path using `\graphicspath`, `jmlrbook` won't find the graphics if the imported articles aren't in the same directory as the book.

## 4.6 Creating the Book Using `makejmlrbook`

The `makejmlrbook` Perl script is designed to make it easier to produce the print and online versions of the book, as well as producing an HTML index of all the imported articles with links to the abstracts and PDFs of individual articles. Note that for it to work properly, the articles must be imported using `\importarticle`, `\importpaper` or `\importpubpaper`, and the imported articles must use the `jmlr` class. Note that I have only tested `makejmlrbook` on Linux.

(If you don't like using a command line script or you want to use some diagnostic tools, you may prefer to use `MakeJmlrBookGUI`, which can be downloaded from <http://www.dickimaw-books.com/apps/makejmlrbookgui/>)

On UNIX style systems, the script can be invoked from a terminal using:

```
makejmlrbook [<options>] <filename>
```

If that doesn't work, or you aren't using a UNIX style operating system, the script can be invoked from a terminal or command prompt using:

```
perl makejmlrbook [<options>] <filename>
```

The mandatory argument *<filename>* is the name of the master  $\TeX$  file containing the book. It must use the `jmlrbook` class. You may omit the `.tex` extension. For example, if the file is called `proceedings.tex`, you can call `makejmlrbook` as follows:

```
perl makejmlrbook proceedings
```

This will create the files `proceedings-print.pdf` (the print version) and `proceedings-online.pdf` (the online version). It will also create a directory (folder) called `html` in which the HTML files and individual article PDFs will be placed.

The options to `makejmlrbook` are as follows:

`--online` Generate the color on-line version (default).

`--noonline` Don't generate the color on-line version.

`--print` Generate the grayscale print version (default).

---

<sup>8</sup>and may also cause a problem for the editor's text editor.

- `--noprint` Don't generate the grayscale print version.
- `--html` Generate the HTML files and the individual article PDFs (default).
- `--nohtml` Don't generate the HTML files and the individual article PDFs.
- `--logourl` *<url>* Make the logo on the HTML index page link to *<url>*.
- `--extractpreface` Extract the preface as a standalone document with links in the HTML index. (Only has an effect if combined with `--html` option.) This will only work if the preface has been put inside the preface environment with the signoff environment that each editor with `\Editor`.
- `--noextractpreface` Don't try extracting the preface. (Default.)
- `--batchtex` Run  $\TeX$  in batch mode.
- `--nobatchtex` Don't run  $\TeX$  in batch mode (default).
- `--quieter` Reduce chatter to STDOUT (doesn't eliminate all messages). This also runs  $\TeX$  in batch mode.
- `--noquieter` Don't reduce messages to STDOUT (default).
- `--version` Display the version number and exit.
- `--help` List all available options.

There are also some more advanced options, but these haven't been fully tested:

- `--latexapp` *<name>* Application used to call  $\LaTeX$ . Defaults to "pdflatex".
- `--latexopt` *<string>* Options to pass to  $\LaTeX$ .
- `--format` *<string>* Output format (defaults to "pdf"). This may need to be changed if you change the  $\LaTeX$  application.
- `--bibtexapp` *<name>* Application use to process the bibliography. Defaults to "bibtex".
- `--bibtexopt` *<string>* Options to pass to Bib $\TeX$ .

## Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

<b>Symbols</b>		
\\	..... 5	\citet ..... 8
<b>A</b>		class options:
abstract (environment)	. 7	10pt ..... 18
\acks	..... 7	11pt ..... 18
\addr	..... 6	12pt ..... 18
\addtocpart	..... 22	7x10 ..... 18
algorithm (environment)	..... 11, 26	color ..... 3, 18
algorithm2e (environment)	..... 11, 26	gray ..... 3, 17, 18
algorithm2e package	..... 3, 11, 12, 26	letterpaper ..... 18
\algorithmref	..... 13	nowcp ..... 3, 18
align (environment)	... 15	onecolumn ..... 3
align* (environment)	.. 15	tablecaption
altdescription (environment)	..... 12	bottom ..... 3, 18
amsmath package	. 3, 14–16	top ..... 3, 18
amssymb package	..... 3	twocolumn ..... 3, 23
\AND	..... 5	wcp ..... 3, 4, 18
\and	..... 5	combine class ..... 2, 3, 25
\appendix	..... 8, 22	combnat package ..... 3
\appendixref	..... 13	conjecture (environment)
\articleauthorref	... 24	..... 12
\articlepageref	..... 24	\conjectureref ..... 14
\articlepagesref	... 24	corollary (environment)
\articletitleref	... 24	..... 12
\Author	..... 20	\corollaryref ..... 14
\author	..... 5, 19	<b>D</b>
authorsignoff (environment)	..... 20	\DeclareMathOperator
axiom (environment)	... 12	..... 16
\axiomref	..... 14	\DeclareMathOperator*
<b>B</b>		..... 16
\bibliography	..... 8	definition (environment)
<b>C</b>		..... 12
\caption	..... 11	\definitionref ..... 14
\chapter	..... 22	description (environment)
\chapterref	..... 24	..... 12
\citep	..... 8	<b>E</b>
		\Editor ..... 21, 27
		\editor ..... 5
		\editors ..... 5
		\Email ..... 5
		enumerate (environment)
		..... 11
		enumerate* (environment)
		..... 11
		environments:
		abstract ..... 7
		algorithm ..... 11, 26
		algorithm2e ... 11, 26
		align ..... 15
		align* ..... 15
		altdescription ... 12
		authorsignoff .... 20
		axiom ..... 12
		conjecture ..... 12
		corollary ..... 12
		definition ..... 12
		description ..... 12
		enumerate ..... 11
		enumerate* ..... 11
		equation ..... 15, 26
		example ..... 12
		jmlrpapers .... 23, 23
		keywords ..... 7
		lemma ..... 12
		preface .... 21, 21, 27
		proof ..... 12
		proposition ..... 12
		remark ..... 12
		signoff .... 21, 21, 27
		split ..... 15
		table ..... 3, 18
		theorem ..... 12
		epsfig package ..... 2, 3
		equation (environment)
		..... 15, 26
		\equationref ..... 13
		example (environment) . 12
		\exempleref ..... 14
		<b>F</b>
		\figureref ..... 13
		fink package ..... 3
		\floatconts .... 3, 9, 18
		\frontmatter ..... 20

<b>G</b>		<code>\lemmaref</code> ..... 14	<code>\set</code> ..... 14	
<code>\gdef</code> ..... 19	<code>\logo</code> ..... 19	<b>M</b>		
geometry package .... 2, 26		<code>\mailto</code> ..... 7	setspace package ..... 3	
<code>\global</code> ..... 19		<code>\mainmatter</code> ..... 21	<code>\SetTitleElement</code> .... 25	
graphicx package .... 3, 9		makejmlrbook ..... 3, 17, 21, 26, 27	signoff (environment) ..... 21, 21, 27	
<b>H</b>		<code>\maketitle</code> ..... 4, 20	siunitx package ..... 19	
hyperref package ... 2, 3, 25			split (environment) ... 15	
<b>I</b>		<b>N</b>		
<code>\ifprint</code> ..... 16		<code>\Name</code> ..... 5	subfig package ... 2, 10, 26	
<code>\IfTitleElement</code> ..... 25		nameref package ..... 3	<code>\subfigref</code> ..... 13	
<code>\importarticle</code> ..... 23		natbib package ..... 3, 8	<code>\subfigure</code> ..... 10	
<code>\importpaper</code> ..... 23, 24		<code>\newcommand</code> ..... 19	<code>\subparagraph</code> ..... 22	
<code>\importpubarticle</code> ... 24		<code>\newenvironment</code> ..... 19	<code>\subsection</code> ..... 7, 22	
<code>\importpubpaper</code> .. 23, 24			<code>\subsection*</code> ..... 7	
<code>\includegraphics</code> ... 9, 16, 19		<b>O</b>		
<code>\includeteximage</code> .... 10		<code>\operatorname</code> ..... 15	<code>\subsubsection</code> .... 7, 22	
inputenc package ..... 26		<code>\operatorname*</code> ..... 16	<code>\subsubsection*</code> ..... 8	
<code>\intertext</code> ..... 15			<code>\subtable</code> ..... 10	
<code>\issue</code> ..... 25			<code>\subtabref</code> ..... 13	
<b>J</b>		<b>P</b>		
jmlr2e package ..... 2-5		<code>\paragraph</code> ..... 22	table (environment) . 3, 18	
jmlrbook package ..... 26		<code>\part</code> ..... 22	<code>\tableofcontents</code> .... 21	
<code>\jmlrnowcp</code> ..... 18		pdfpages package .... 2, 26	<code>\tableref</code> ..... 13	
jmlrpapers (environ- 23, 23		preface (environment) ..... 21, 21, 27	<code>\team</code> ..... 19	
ment) ..... 23, 23		<code>\prefacename</code> ..... 21	<code>\text</code> ..... 15	
<code>\jmlrprehyperref</code> .... 2		<code>\productioneditor</code> ... 20	theorem (environment) . 12	
<code>\jmlrproceedings</code> .... 18		proof (environment) ... 12	theorem package ..... 2	
<code>\jmlrpublished</code> ..... 4		proposition (environ- 12	<code>\theoremref</code> ..... 14	
<code>\jmlrsubmitted</code> ..... 4		ment) ..... 12	tikz package ..... 10	
<code>\jmlrvolume</code> ..... 4		psfig package ..... 2	<code>\title</code> ..... 4, 19	
<code>\jmlrwcp</code> ..... 18			<code>\titlebody</code> ..... 25	
jmlrwcp2e package .... 2-5		<b>R</b>		
<code>\jmlrworkshop</code> ..... 4		remark (environment) .. 12	<code>\titlebreak</code> ..... 5	
<code>\jmlryear</code> ..... 4		<code>\remarkref</code> ..... 14	<code>\twocolumn</code> ..... 23	
		<code>\renewcommand</code> ..... 19	<b>U</b>	
		<code>\renewenvironment</code> ... 19	<code>\url</code> ..... 6	
			url package ..... 3	
<b>K</b>		<b>S</b>		
keywords (environment) . 7		scrartcl class ..... 3	<b>V</b>	
<b>L</b>		<code>\section</code> ..... 7, 22	<code>\volume</code> ..... 19	
<code>\label</code> ..... 11, 13		<code>\section*</code> ..... 7	<b>X</b>	
lemma (environment) ... 12		<code>\sectionref</code> ..... 13	xcolor package ..... 17	
			xkeyval package ..... 3	