

# A L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> class for Draft Screenplays

by  
John Pate

This paper describes class `screenplay`  
version v1.518, released 2006/11/03

## 1 Introduction

Every year *The Academy of Motion Picture Arts and Sciences* runs a competition for screenplay writers, the *Nicholl Fellowships in Screenwriting*. Full-length feature film screenplays are submitted (for a fee). You can win money but, more importantly, placing in the competition will make agents take notice. *The Academy* recommends a particular format which this `class` file attempts to duplicate, using a `.pdf` available from *The Academy* website as the model. This format has various functions, amongst which is a gate-keeping function to show you have at least a first idea. It also gives readers a better chance of assessing how long, in filmic terms, your screenplay would be. Formatted along the lines recommended, you should aim to have your screenplay in the region of 120 pages long. Although they say A4 paper is acceptable, it is widely held that you severely handicap yourself if you don't use American 8.5" by 11" Letter size paper. I have a ream on my lap as I type: 20 pound, 106 bright, acid free. My sister brought it from America for me—it's very expensive to buy outside the US and not easy to find. One might imagine it's at the photocopy stage that dealing with A4 in America becomes a nuisance—one more reason to put your magnum opus to the bottom of the pile.

Of course, if you didn't know this already you likely need to do some serious reading and research *before* you start writing. I guess trying to understand what's going on here is a start.

The format specifications are pretty straightforward. You'll have to believe me when I say it wasn't quite so straightforward for me to start from a minimal understanding of T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> to a usable `.dtx` file.

This is an attempt to turn T<sub>E</sub>X into a typewriter emulator. A crazy idea—which is why it appealed to me, obviously. Fortunately I'm easily entertained.

I hope `screenplay.dtx` proves of use (and instruction) to others as it has to me.

Anyhoo, on with the show...

## 2 Getting started

You should have:

- `README`—plain text file
- `screenplay.dtx`—the source file
- `screenplay.ins`—driver file
- `screenplay.pdf`—the file I assume you’re reading right now
- `test.pdf`—format test .pdf result
- `COPYING`—GNU General Public License

If not, get `screenplay.zip` via,  
<http://dvc.org.uk/sacrific.txt/screenplay.zip>  
Use the command,  
> `latex screenplay.ins`  
to generate from `screenplay.dtx`...

- `screenplay.cls`—class file
- `hardmarg.sty`—optional style package
- `example.tex`—example (very) mini-screenplay
- `test.tex`—format test source file

Once you’ve unpacked the bundle and generated the `class` file do,  
> `latex test`  
to generate the `test.dvi` you can compare with the `test.pdf`.

Make sure everything in the printing chain can deal properly with 8.5ins by 11ins Letter paper. Print out your version of `test` and the provided `test.pdf` (on Letter paper of course!) and compare them. The two printouts should be dimensionally identical. The left margin should measure as *exactly* 1.5ins. If not you have an old version of `dvips` and/or your printer driver is configured incorrectly and you’ll need to fix things before you proceed further.

## 3 Some example markup

Although `screenplay.cls` builds on the standard `TeX` document class `article`, it is not intended you pass it any options (except for the release date specification, see below). The formatting required is minimalist. `screenplay.cls` will issue an error message if you use any options.

`\documentclass{screenplay}[2006/11/03]`  
is your first command, you can give the optional date argument in case of any earlier copies of `screenplay.cls` lying around.

You are now in the *document preamble* where you should initialise the values of some commands.

You must define the title of screenplay...

```
\title{The Title of Your Opus}
```

...it will be uppercased automagically so don't fuss. If it's for the *Nicholl Competition* you're done. However, you may as well also initialise...

```
\author{Your Name}
```

...capitalised as in ordinary writing, it's not munged in the typesetting, and...

```
\address{99 Any Street \\
```

```
Any Town \\ Postcode / Zip \\
```

```
Phone \\
```

```
email@whereami }
```

The preamble is closed, of course, by...

```
\begin{document}
```

The first typesetting command, in the body of the document, should be one to build the title page. For the *Nicholl Competition* this is `\nicholl`, *alternatively* for the style used when submitting to an Agent, etc. use `\coverpage` (which uses your address and name from the preamble).

The first thing you need to see in your screenplay is FADE IN: left justified with a blank line underneath. Scripts always start this way, because they do. You deploy the `\fadein` command for this.

Next would typically be a slugline,

```
\intslug[day]{Johnny's House}
```

which would produce...

```
INT. JOHNNY'S HOUSE    DAY
```

I made the time of day optional because my screenplay had stuff happening in space where there is no time of day. You will likely *always* use either *day* or *night*. (Refer to *The Academy's* information on formatting.) Likewise there's an `\extslug[]{} for EXT.`

After a slugline you would typically have some description. This is simply ordinary text with paragraph breaks as per usual.

When you get to dialogue, you have the `dialogue environment`.

```
\begin{dialogue}{Johnny} After some description you
would typically have some dialogue. \end{dialogue}
```

Directions (short) can be put inside dialogue text either as an optional argument to the dialogue environment, `\begin{dialogue}[shouting]{Johnny}` which will result in parenthesized text below the dialogue name, or the `\paren{}` command can be used to put parentheses into the blocks of dialogue text.

Do not put any blank lines in dialogue text.

So you wander on with your screenplay requiring little in the way of  $\LaTeX$  2 $\epsilon$ 's powers until you reach the end of the final scene. Personally I use *Elvis* with a selection of macros in my `.exrc` to save me having to type the commands I've invented.

The command at the end of the final scene is always `\fadeout` to produce the words FADE OUT: flushright.

Then it's simply a case of `\theend` and `\end{document}`.

The only tricky parts of the formatting are deciding where to put *pagebreaks* and hunting down places where *linebreaks*... `\`... will help `TeX` out with formatting. You may need to insert linebreaks manually even where ‘`Overfull \hbox`’ warnings are absent. You’ll know when it is necessary by spotting extra vertical whitespace in the document. `example.tex` demonstrates this, read through the raw `.tex` file and try modifying it as indicated. It’s likely you will have to put quite a few linebreaks “by hand” into dialogue text due to the restricted margins.

**Do not put in pagebreaks until you have really, really, really finished editing.**

Putting in pagebreaks will likely involve using the `\dialbreak` command, though for description text `\pagebreak` is all that is required.

The `\dialbreak` command needs to know the character name and any optional directions as required.

If you’ve used the `\titleover` environment you may need the `\titbreak` command (more tea, vicar?). However, mostly for TITLE OVER: you’ll be doing a date to accommodate flashbacks or flashforwards, I expect, so use the `\centretitle` command as in `\centretitle{December 21, 2012}`.

Yes, that’s the English (or for that matter French and Canadian) “centre” not “center” in a stunning break from tradition.

For further enlightenment, you should scrutinise the markup of `example.tex` and peruse **The Coding** section of this document.

You can see a completed script marked-up via `LATEX 2ε` and formatted with `screenplay.cls` at <http://dvc.org.uk/sacrific.txt/>. There is also, at time of writing, a “work-in-progress” script that has had manual linebreaks inserted but not manual pagebreaks.

## 4 Hard Margins package

Here we’re emulating a typewriter. On a typewriter, the left margin is a hard stop, whereas the right margin is a bell (those of us who typed in the stone age well remember that “ding”). Even the best typist would run past the right margins by a character or two once in a while, sometimes deliberately. Hence the “fudge factors” interpolated by me, which in days of yore would perhaps have been referred to as the *gutter*.

The provided `hardmarg.sty` shows you what happens with the *fudge factors* eliminated. Place the command `\usepackage{hardmarg}` in the document preamble (i.e. between `\documentclass{...}` and `\begin{document}`) and judge for yourself. Personally I’ll stick with my *fudge*. All the brain-work makes me crave carbs anyway. But if you want, use it to fix your script to tighter margins.

## 5 User customisation

**Don't!** It really is a case of KISS for Draft Screenplay formatting. There are a couple of things you might get away with though...

You may want to change how the time of day is spaced from the rest of the slugline. You can do this without editing the class file by redefining the `\slugspace` command e.g. `\renewcommand*{\slugspace}{ - }` which will change the default 3em space to em hyphen em spacing. Likewise you could `\renewcommand*{\pov}{POV}` to lose the dots. If for some strange reason you don't like *The Academy's* use of uppercase for (MORE) and (CONT'D) you can `\renewcommand*` the `\more` and `\contd` macros.

Should you not want page number 1 to appear on the first page of the body of the script put the `\thispagestyle{empty}` command in your document source file immediately *after* the command to build the title page.

Likely, in the usual L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> fashion, you should develop your own style package depending on what conventions you want to use.

## 6 Specs from *The Academy*

Be sure to read, carefully, and inwardly digest and re-read,

<http://www.oscars.org/nicholl/format.html>

and,

<http://www.oscars.org/nicholl/script.pdf>,

(the model document).

- 20lb Letter paper (8.5"x11.0")  
Courier 12pt, 10 pitch
- **Page**  
Top margin: 1.0"  
Bottom margin: 0.5"-1.5"  
Left margin: 1.5"  
Right margin: 1.0"
- **Dialogue**  
tab NAME to: 4.2"  
Left margin: 3.0"  
Right margin: 2.5"  
*Parenthetical*  
Left margin: 3.7"  
Right margin: 3.3"
- **Numbering**  
top right, Arabic  
Top margin: 0.5"  
Right margin: 0.75"

INT/EXT. SLUG LINE DESCRIPTION DAY/NIGHT  
(with some kind of extra spacing before time of day)

## 7 The Coding

screenplay.cls v1.518 was tested with...

- pdf<sub>ε</sub>TeX 3.141592-1.21a-2.2 (WEB2C 7.5.4)
- dvips(k) 5.95a
- kpathsea version 3.5.4
- article 2004/02/16 v1.4f

IMHO, it's worth reading through the comments, there's some useful stuff for you in there.

We are a L<sup>A</sup>T<sub>ε</sub>X 2<sub>ε</sub> thang...

```
1 \NeedsTeXFormat{LaTeX2e}[2003/12/01]
```

Say it like it is...

```
2 \ProvidesClass{screenplay}[2006/11/03 v1.518
3 LaTeX2e Draft Screenplay document class
4 (C)2006 John Pate
5 released under the GNU General Public License]
```

Whinge if the user is being a bozo...

```
6 \DeclareOption*{%
7     \PackageError{screenplay}{Unwanted option '\CurrentOption'}%
8     {There are no options available for class screenplay v1.518}}
9 \ProcessOptions\relax
```

ifthen needed for testing for optional args...

```
10 \RequirePackage{ifthen}[2001/05/26]
```

Use article class for its base functionality, *must* be American Letter paper *not* A4, screenplay.cls eats any papersize options and ignores them...

```
11 \LoadClass[12pt,letterpaper,onecolumn,oneside]{article}[2004/02/16]
```

geometry package puts marks in to suggest to dvi interpreters to do the right thang for Letter paper...

```
12 \RequirePackage[dvips,letterpaper]{geometry}[2002/07/08]
```

...and make sure your printer driver (which means *Ghostscript* in my case) knows Letter. I say again, you *must* use American 8.5"x11" Letter paper—stationers in the UK may call it "US Quarto" if they've even heard of it, see...  
<http://www.gwydir.demon.co.uk/jo/units/print.htm>

```
13 \setlength{\paperheight}{11in}
14 \setlength{\paperwidth}{8.5in}
```

	<p><i>Must</i> have <code>Courier</code> as font (may cause warnings about font stuff, ignore)...</p> <pre>15 \RequirePackage{courier}[2004/09/15] 16 \renewcommand*{\familydefault}{\ttdefault}</pre> <p>You should <i>not</i> hyphenate or full justify for Nicholl format...</p> <pre>17 \raggedright</pre>
Page Nums	<p>Arabic top right, I'll live with where <math>\mathrm{T}_{\mathrm{E}}\mathrm{X}</math> puts the margin...</p> <pre>18 \pagenumbering{arabic} 19 \pagestyle{myheadings}</pre> <p>(You can force the page numbers out towards the right with <code>\markright{\hspace*{6.2in}}</code> in the document preamble, at the cost of ‘Overfull <code>hbox</code>’ warnings, if you like.)</p>
CR/LF	<p>Line spacing / pitch—it looks to me the standard <math>\mathrm{L}^{\mathrm{A}}\mathrm{T}_{\mathrm{E}}\mathrm{X}2_{\epsilon}</math> <code>courier</code> font is correct pitch, 12pt skip is what's in <code>script.pdf</code>...</p> <pre>20 \setlength{\baselineskip}{12pt plus 0pt minus 0pt}</pre> <p>and this <code>\baselinestretch</code> appears to emulate <code>script.pdf</code> closely and, more importantly, measures up as matching the effect of the <code>\baselineskip</code> value—it matches what <i>Movie Magic Screenwriter</i> does, for instance...</p> <pre>21 \renewcommand*{\baselinestretch}{0.8}</pre> <p><i>vspace</i> adjustment in <code>\minipage</code> and <code>\parbox</code> in dialogue to correct line spacing, should likely be adjusted if <code>\baselinestretch</code> is altered for some (crazy) reason...</p> <pre>22 \newcommand*{\dialfix}{\vspace{2pt}}</pre> <p>skip one line between paras...</p> <pre>23 \setlength{\parskip}{\baselineskip}</pre> <p>no paragraph indent...</p> <pre>24 \setlength{\parindent}{0in}</pre>
<code>\slugspace</code>	<p>Extra inter-word spacing in sluglines...</p> <pre>25 \newcommand*{\slugspace}{\hspace{2.0em plus 0em minus 0em}}</pre> <p>(will result in a total of 3em space).</p>
Margins/tabs	<p>See specs, fudge factors (i.e. a “gutter”) are needed to resemble the “correct” layout—take a ruler to it and <code>script.pdf</code>. 0.1” represents 1 em in <code>courier</code> 12pt 10 pitch.</p>
left margin	<p>1.5”, default 1” +</p> <pre>26 \setlength{\evensidemargin}{0.5in} 27 \setlength{\oddsidemargin}{0.5in}</pre>
top margin	<p>0.5” to number, 1” to body...</p> <pre>28 \setlength{\topmargin}{-0.5in} 29 \setlength{\headsep}{0.4in} 30 \setlength{\topskip}{0in}</pre>

bottom margin 11" - (1" top margin) - (0.5" bottom) =  
31 \setlength{\textheight}{9.5in}  
...bottom can go to 1.5" for page breaking purposes.

right margin 6" width for description text to give 1" right margin, +0.1" fudge...  
32 \setlength{\textwidth}{6.0in}  
33 \newlength{\widthgutter}  
34 \setlength{\widthgutter}{0.1in}  
35 \addtolength{\textwidth}{\widthgutter}  
(later on, we must adjust centring and flush right to the "correct" margin by subtracting \widthgutter).

### Tabbing / spacing dialogues

Width of dialogue text, 3" with 0.2" fudge...  
36 \newlength{\dialwidth}  
37 \setlength{\dialwidth}{3.0in}  
38 \newlength{\dialgutter}  
39 \setlength{\dialgutter}{0.2in}  
40 \addtolength{\dialwidth}{\dialgutter}  
3" from left edge of page (1.5" left margin + 1.5")...  
41 \newcommand\*{\dialtab}{\hspace\*{1.5in}}  
**name**, tab in 4.2" (margin + \dialtab + \dialnametab) from left edge of page...  
42 \newcommand\*{\dialnametab}{\hspace\*{1.2in}}  
parenthetical 1.5" with 0.2" fudge...  
43 \newlength{\parenwidth}  
44 \setlength{\parenwidth}{1.5in}  
45 \addtolength{\parenwidth}{\dialgutter}  
parenthetical, tab in 3.7" (margin + \dialtab + \parentab) from left edge of page...  
46 \newcommand\*{\parentab}{\hspace\*{0.7in}}  
AFAICS floats not involved...  
47 \renewcommand\*{\textfraction}{1}  
48 \renewcommand\*{\topfraction}{0}  
49 \renewcommand\*{\bottomfraction}{0}

**Title Pages** Use one of \nicholl or \coverpage to taste. This must be the first typesetting command after \begin{document}. (\title and \author code is already defined in the article class.)

\nicholl For Nicholl Competition—just the title...  
50 \newcommand{\nicholl}{  
51 \thispagestyle{empty}  
52 \vspace\*{2.25in}  
53 \begin{center}\MakeUppercase{\@title}\end{center}}



```
54 \newpage\setcounter{page}{1}
55 }
```

**\address** Your address details for the title page...

```
56 \newcommand{\@address}{Contact via Agency}
57 \newcommand{\address}[1]{\renewcommand{\@address}{#1}}
...see also...
```

**\agent** Some people like to put copyright and WGA registration details on their submitted scripts...although the received wisdom is that this marks you as an amateur, you can use this macro.

I'm calling this **\agent** to pretend you're putting your Agent's details instead of your own address—hence if you don't define **\address** (and use **\coverpage**) you'll get the boilerplate text “Contact via Agency” under your name in the “author's address details” area. What you would do is omit your own address and only have your Agent's address. Putting this the other side of the page makes it immediately clear what's who. No really, there's no need to use this *unless* you're with an agency but it's your script...

```
58 \newcommand{\@agent}{~}
59 \newcommand{\agent}[1]{\renewcommand{\@agent}{#1}}
```

**\titlesepspace** Determines how far down below the title the extra info on the title page is written. You may have to adjust the **\titlesepspace** if you put in a lot of text for **\address** and/or **\agent** but if you do there's probably too much—it's a rubber length...

```
60 \newlength{\titlesepspace}
61 \setlength{\titlesepspace}{3.75in plus 0in minus 1.5in}
```

**\coverpage** Now the ability to make the the type of title page for Draft Scripts submitted to Agents,

```
62 \newcommand{\coverpage}{
63 \thispagestyle{empty}
64 \vspace*{2.25in}
65 \begin{center}{\MakeUppercase{\@title}}\\
66 \vspace{\baselineskip}
67 by\\
68 \vspace{\baselineskip}
69 \@author
70 \end{center}
71 \vspace{\titlesepspace}}
```

**\agent** details at left margin in a **\parbox**, 2” space then **\author** and **\address** details in a **\parbox**...

```
72 \parbox[t]{2.0in}{\@agent}\hspace{2.0in}
73 \parbox[t]{2.0in}{\@author\\ \@address}
```

Reset page numbers...

```
74 \newpage\setcounter{page}{1}
75 }
```

End title page—author details

`\more \contd` **More and Continued.** You may want to use these macros if you feel it's necessary to follow what appears to be the “older” style where they aren't only used for pagebreaks...

```
76 \newcommand*{\more}{(MORE)}
77 \newcommand*{\contd}{(CONT'D)}
```

**Dialogue blocks...**

`\paren` `\paren{hello world}`  
 parenthetical within dialogue environment...only for *very short* text (directions) within dialogue, the optional argument is for internal use *only*...

```
78 \newcommand{\paren}[2][\]{%
79 #1\parentab\parbox[t]{\parenwidth}{\raggedright(#2)\dialfix}\
80 }
```

`dialogue env` `\begin{dialogue}[directions]{character name} ... \end{dialogue}`  
`\begin{dialogue}{character name} ... \end{dialogue}`  
`\minipage` won't allow pagebreaks, see below for new manual pagebreak command for this environment. Do *not* have blank lines inside this environment. **Name** is outside `\minipage` to mitigate ‘Overfull \hbox’ warnings with long names...

```
81 \newenvironment{dialogue}[2][]{
82
83 \dialtab\dialnametab\hbox{\MakeUppercase{#2}}\*
84 \dialtab\begin{minipage}[b]{\dialwidth}
85 \dialfix
86 \raggedright
87 \ifthenelse{\not\equal{#1}{}}
88 {
89 \paren[] {#1}
90 }{ }
91 }{
92 \end{minipage}
93
94 }
```

`\dialbreak` Forcing pagebreak in dialogue,  
`\dialbreak{Character}`  
`\dialbreak[directions]{Character}`  
 (need to know character name— $\text{\LaTeX 2}_{\epsilon}$  scoping rules don't seem to like global vars, somebody tell me otherwise) *Note:* only do any manual pagebreaks when you absolutely positively have killed every muthafucka in the room!...

```
95 \newcommand{\dialbreak}[2][]{
96 \\\dialnametab\hbox{\more}%
97 \end{dialogue}
98 \pagebreak
```

```

99 \ifthenelse{\not\equal{#1}{}}
100 {
101 \begin{dialogue}[#1]{\MakeUppercase{#2} \contd}
102 }{
103 \begin{dialogue}{\MakeUppercase{#2} \contd}
104 }
105 }

```

### Sluglines

Optional arg for night/day. You probably should not consider time of day optional when you're writing descriptions—unless you're in a spaceship, of course...

```

\intslug INT. SLUGLINE DESCRIPTION DAY
\intslug[time of day]{description text}
\intslug{description text}
106 \newcommand{\intslug}[2][]{
107
108 \MakeUppercase{INT. #2}%
109 \ifthenelse{\not\equal{#1}{}}
110     {%
111 \slugspace\MakeUppercase{#1}\hspace{\fill}
112     }{}
113 \nopagebreak
114
115 }

```

```

\extslug EXT. SLUGLINE DESCRIPTION DAY
\extslug[time of day]{description text}
\extslug{description text}
116 \newcommand{\extslug}[2][]{
117
118 \MakeUppercase{EXT. #2}%
119 \ifthenelse{\not\equal{#1}{}}
120     {%
121 \slugspace\MakeUppercase{#1}\hspace{\fill}
122     }{}
123 \nopagebreak
124
125 }

```

```

\intextslug Apparently people do this...
\intextslug INT./EXT. SLUGLINE DESCRIPTION DAY
\intextslug[time of day]{description text}
\intextslug{description text}
126 \newcommand{\intextslug}[2][]{
127
128 \MakeUppercase{INT./EXT. #2}%
129 \ifthenelse{\not\equal{#1}{}}

```

```

130             {%
131 \slugspace\MakeUppercase{#1}\hspace{\fill}
132             }{}
133 \nopagebreak
134
135 }

\extintslug and/or this...
EXT./INT. SLUGLINE DESCRIPTION DAY
\extintslug[time of day]{description text}
\extinttextslug{description text}
136 \newcommand{\extintslug}[2][]{
137
138 \MakeUppercase{EXT./INT. #2}%
139 \ifthenelse{\not\equal{#1}{}{
140             {%
141 \slugspace\MakeUppercase{#1}\hspace{\fill}
142             }{}
143 \nopagebreak
144
145 }

Title Over

titleover env \begin{titleover} ... \end{titleover}
as pseudo dialogue environment (think “Star Wars”) the optional argument is
for internal use only...
146 \newenvironment{titleover}[1][]{
147
148 \ifthenelse{\not\equal{#1}{}{
149 {
150 TITLE OVER #1:
151 }{
152 TITLE OVER:
153 }%
154 \\*[\baselineskip]
155 \dialtab\begin{minipage}[b]{\dialwidth}
156 \raggedright
157 }{
158 \end{minipage}
159
160 }

\titbreak pagebreak for titleover environment...
161 \newcommand{\titbreak}{
162 \\ \dialnametab\hbox{\more}%
163 \end{titleover}
164 \pagebreak
165 \begin{titleover}[\contd]
166 }

```

```

\centretitle \centretitle {short text}
titleover for short text most often a date, centred with adjustment for gutter...
167 \newcommand{\centretitle}[1]{
168
169 TITLE OVER:\*[ \baselineskip]
170 \hspace*{\fill}{#1}\hspace{\fill}\hspace{\widthgutter}
171
172 }

```

### Other macros

```

\fadein Always the first thing in your script, for some reason...
173 \newcommand{\fadein}{
174
175 FADE IN:
176
177 }

\fadeout Always the last thing in your script for some reason, flush right with adjustment
for gutter...
178 \newcommand{\fadeout}{
179
180 \hspace*{\fill}FADE OUT:\hspace{\widthgutter}
181
182 }

\intercut flush right with adjustment for gutter...
183 \newcommand{\intercut}{
184
185 \hspace*{\fill}INTERCUT WITH:\hspace{\widthgutter}
186
187 }

\pov \revert Switch point of view...
188 \newcommand*{\pov}{P.O.V.}
...switch it back...
189 \newcommand*{\revert}{REVERSE \pov}

\theend But this is what we've all been waiting for...OK, so this is actually the last
thing in the script...
190 \newcommand{\theend}{
191
192 \hspace*{\fill}THE END\hspace{\fill}\hspace{\widthgutter}
193
194 }%THE END

```

## hardmarg.sty **Hard Margins package**

Re-defines margins to exactly as stated in `script.pdf`. Try it out and see if you like it. Also shows you what todo if you want to fiddle with some of the margin settings...

Issue info as to what we do...

```
195 \ProvidesPackage{hardmarg}[2006/11/03 v1.518
196 For use with class screenplay
197 (C)2006 John Pate
198 released under the GNU General Public License]
```

Hard 1" right margin...

```
199 \setlength{\textwidth}{6.0in}
```

Flushright and centring use this, so eliminate it...

```
200 \setlength{\widthgutter}{0in}
```

Set dialogue to 3" width, right margin 2.5"...

```
201 \setlength{\dialwidth}{3.0in}
```

Set parentheticals to 1.5" width, right margin 3.3"...

```
202 \setlength{\parenwidth}{1.5in}
```

## 8 More about paper

Unless you're in America, getting Letter paper will be a problem. Another issue is that the recommended 20lb weight paper is equivalent to 75gsm, a non-standard weight for metric papers. (Well, strictly speaking the Americans have defined their measures using metric standards but let's not get into that.) Americans have a 24lb weight paper that's close to 80gsm. I've got samples of Letter in both 20lb and 80gsm. The same script, with the same card covers (240gsm plain white card) measures 12mm vs 15mm in thickness. The 80gsm stuff is noticeably heavier and thicker (less flexible).

IMHO good quality 20lb is nicer in handling and you can very much tell the difference. Old-style typing paper was usually pretty thin for "normal" use.

Copy Shops will cut down A3 paper to 8.5" by 11" for you but whether they'll get you 75gsm weight is another matter. One told me he could "probably" get 75gsm paper if I really wanted it.

Buying Letter paper mail order in the UK is very expensive and I haven't seen American-style pound bond weights quoted—the "heavy" paper I have was sourced in the UK as XEROX PREMIER 216mm x 279mm 80gsm.

Another issue is that the paper must be three-hole punched. Specs gleaned from the Internet suggest that Letter three-hole paper is a central hole (i.e. 140mm down from the top edge) with the other two holes 108mm up, and down, from that central hole. I'm hoping to get some genuine three-hole from America to check that out, meantime I got a Copy Shop to drill a ream of paper for me (and some card for covers). I'd advise that if you can't buy punched paper, get a Copy Shop to drill it for you. It produces a vastly neater result

than using a hole punch—and a suitable hole punch for “home use” is going to be rather expensive in any case.

And then there’s the paper fasteners...

To cut a long story short, here’s what you want:

- ACCO solid brass #5 brads, 1.25” long, part number 71505;
- ACCO #2 brass washers, part number 71511.

...and a rubber mallet to hammer them in with. The rubber mallet is easy enough to come by but you’ll have to source the paper fasteners from America (it seems to be mainly the screenwriting fraternity that call them “brads” rather than paper fasteners, BTW). The brass-plated ersatz versions available in the UK are clearly *not* up to the job by a long shot. And beware that ACCO make a version that isn’t solid brass and is inferior to these all-brass ones, hence the necessity for part numbers.

The washers are for the side where the prongs are folded over, you use **two brads**, each with a single washer, one in the top and one in the bottom hole leaving the central hole empty.

And while I’m on about paper, 120 pages is really the absolute longest you should go. A script that runs between 100 and 110 pages is a better bet—and that’s with plenty of whitespace in the passages of description.

## 9 Todo

If anybody else uses it maybe they’ll spot stuff that’s wrong but *Johnny* says, “It works for me.”

The test file `test.tex` shows `screenplay` markup applied to “FOR A FEW DAYS MORE” (<http://www.oscars.org/nicholl/script.pdf>) and the `test.pdf` file was generated from that.

A printout of the first page `script.pdf` and `test.pdf` show identical pitch and the inter-line spacing is pretty close. However, I chose a `\baselinestretch` value to complement the `\baselineskip` and be more typographically correct rather than exactly match `script.pdf`.

You can easily tell which is which because `test.pdf` numbers the first page whilst `script.pdf` starts numbering on page 2.

If I get any *useful* feedback I *may* alter things accordingly.

Feel free to help with documentation.

## **10 A note for our Sponsor**

I don't think I'm alone when I say...

Hail Eris!  
All Hail Discordia!



## Change History

1.0.0	General: first <code>screenplay.dtx</code> , 2006/09/23, v1.457 . . . . .	1	1.1.0	General: minor code changes, v1.503 . . . . .	6
1.0.1	General: minor code changes, revision to documentation, v1.461 . . . . .	1	1.1.1	General: minor code changes, v1.504 . . . . .	6
1.0.2	General: minor code changes, revision to documentation, added <code>test</code> file, <code>example</code> file, to <code>.dtx</code> , amended <code>.ins</code> , v1.465 . . . . .	1	1.1.2	General: revision to documentation, v1.507 . . . . .	14
1.0.3	General: minor code changes, revision to documentation, added change history, v1.487 . . . . .	1	1.1.3	<code>CR/LF</code> : minor format fix, v1.508 . . . . .	7
1.0.4	General: enabled hyperlinks in documentation, v1.491 . . . . .	1	1.1.4	<code>Margins/tabs</code> : minor code changes, revision to documentation, v1.509 . . . . .	7
1.0.5	General: minor code changes, revision to documentation, v1.494 . . . . .	1	1.1.5	General: minor code changes, revision to comments, v1.511 . . . . .	6
1.0.6	General: fixed typos, v1.496 . . . . .	1	1.1.6	<code>\more_\contd</code> : minor code changes, revision to documentation, v1.512 . . . . .	10
1.0.7	General: revision to documentation, v1.499 . . . . .	1	1.1.7	<code>\pov_\revert</code> : minor revisions, v1.516 . . . . .	13
1.0.8	General: minor format fix, v1.500 . . . . .	6	1.1.8	<code>Title_Pages</code> : revisions to title page build, documentation, v1.517 . . . . .	8
1.0.9	General: added <code>hardmarg.sty</code> , v1.502 . . . . .	4	1.1.9	<code>Title_Pages</code> : minor revision to title page build, v1.518 . . . . .	8