

The **drama** LaTeX package drama package (Frankenstein's heart)

Matt Swift <swift@alum.mit.edu>

Version: 0.2 Date: 1996/05/05

Documentation revision: 1996/05/05

Abstract

Defines macros for typesetting a basic production-style stage script.
This package is useful but may have problems and is unsupported.

Contents

I	Discussion	2
II	Implementation	3
1	Version control	3
2	characters	4
	2.0.1 sectioning	4
	2.0.2 speakers and stage directions	6

Part I

Discussion

I am close to handling the insertion of a “continued” message at the top of the page when a speech is broken across the page. But it’s not there yet.

`script` This is the only environment so far, for a basic production-style stage script. It is implemented as a bitstyle: see the *bits* package for details of use. I think you only need to issue an `\author {author’s name}` and `\date {date}` command. Also, create an empty file called `bits.cfg` on your system (or put the `\author` declaration in it).

Options come in pairs (boldface is default):

`oneact/multiact` Multiact scripts use act numbers; oneacts just have scenes.

`\character` `\character {full description}{name}{csname}` defines a new character. *full description* is used in the dramatis personae listing. *name* is the name to use to refer to the character in stage directions, and to introduce their speeches. `\character` defines a command `\(csname)` which prints *name* in the font given in `\DirectStyle`. Use this in stage directions. It is an abbreviation macro of category `Character`; following spacing is handled automatically for you. See the *abbrevs* package for details. Also defined is the command `\(csname)says`, which is used to introduce the character’s speeches. The suffix “says” can be changed by redefining `\SpeakAppend` before using `\character`. You can also define it before loading the *drama* package.

`\SpeakAppend`

`\scene` Use `\scene` and `\act` to start acts and scenes.

`\act` `\direct {stage directions}`. Use `\setstage {stage directions}` at the beginning of scenes and acts.

`\direct`

`\setstage`

Part II

Implementation

1 Version control

```
\fileinfo These definitions must be the first ones in the file.
\DoXUsepackage 1 \def\fileinfo{drama package (Frankenstein's heart)}
\HaveECitationS 2 \def\DoXPackageS {}
\fileversion 3 \def\initelyHaveECitationS {}
\filedate 4 \def\fileversion{v0.2}
\docdate 5 \def\filedate{1996/05/05}
\PPOptArg 6 \def\docdate{1996/05/05}
7 \edef\PPOptArg {%
8 \filedate\space \fileversion\space \fileinfo
9 }
```

If we're loading this file from a `\ProcessDTXFile` command (see the *compsci* package), then `\JustLoadInformation` will be defined; otherwise we assume it is not (that's why the FunkY Name).

If we're loading from `\ProcessDTXFile`, we want to load the packages listed in `\DoXPackageS` (needed to typeset the documentation for this file) and then bail out. Otherwise, we're using this file in a normal way as a package, so do nothing. `\DoXPackageS`, if there are any, are declared in the `dtx` file, and, if you're reading the typeset documentation of this package, would appear just above. (It's OK to call `\usepackage` with an empty argument or `\relax`, by the way.)

```
10 \makeatletter% A special comment to help create bst files. Don't change!
11 \ifundefined{JustLoadInformation} {%
12 }{% ELSE (we know the compsci package is already loaded, too)
13 \UndefineCS\JustLoadInformation
14 \SaveDoXVarS
15 \eExpand\csname DoXPackageS\endcsname\In {%use \csname in case it's undefined
16 \usepackage{#1}%
17 }%
18 \RestoreDoXVarS
19 \makeatother
20 \endinput
21 }% A special comment to help create bst files. Don't change!
```

Now we check for $\text{\LaTeX}2\text{e}$ and declare the LaTeX package.

```
22 \NeedsTeXFormat{LaTeX2e}
23 \ProvidesPackage{drama}[\PPOptArg]
24 \RequirePackage{abbrevs,bits,dialogue,moredefs,relsize}
25 %\RequirePackage{xmark}
26 \let\sc@mark\Gobble
27 \VerboseErrors
28
29 \DeclareBooleanOptions{oneact}{multiact}
30 \ProcessOptions
```

2 characters

```
\SpeakAppend
\sc@cast 31 \providecommand\SpeakAppend {says}
\character 32 \ReserveCS\sc@cast
33 \NewUserAbbrevDefiner\NewCharacterAbbrev{Character}
34 \let\TMFontCharacter\scshape
35 \newcommand*\character [3] {% args: dramper name csname
36 \expandafter\NewCharacterAbbrev\csname #3\endcsname{#2}%
37 \NewName* {#3\SpeakAppend} {} {%
38 \speak{#2}%
39 }%
40 \addto@macro\sc@cast{,#1}%
41 }
```

The `\@cdr` gets rid of the first silly comma.

```
42 \newcommand\DramPer {%
43 \ifx\sc@cast\LongEmpty\else
44 {\larger\scshape Cast}%
45 \relax\begin{verse}
46 \edef\sc@t@a{\E@cdr\sc@cast\@nil}%
47 \@for\sc@t@b:=\sc@t@a \do {%
48 \sc@t@b\%
49 }%
50 \relax\end{verse}%
```

Prepare for another set of characters.

```
51 \InitCS\sc@cast
52 \bigbreak
53 \fi
54 }
```

2.0.1 sectioning

`\sc@toks@saved@output` We pull some fancy stuff with the output routine here. The algorithm is a simpler version of Knuth's (Knuth 1986, 260ff).

```
55 \newtokens\sc@toks@saved@output
56 \sc@toks@saved@output=\output
```

The `\sc@botmark`, if not empty, will contain text to be inserted at the top of the next page. It must contain vertical not horizontal material.

We actually need to communicate the boolean `sc@directing` via the mark, because only the information in the marks is guaranteed to reflect the state at the end of the page, and not several lines later.

```
57 \providelength\sc@len
58
59 \newcommand\sc@outputwrapper {%
60 \setlength{\sc@len}{\dp255}%
61 \the\sc@toks@saved@output
62 \def\sc@t@b {%
63 \sc@botmark
64 \addtolength{\sc@len}{-\prevdepth}%
65 \kern\sc@len
```

```

66 }%
67 \InitCS\sc@t@a
68 \ifx\sc@botmark\ShortEmpty
69 \let\sc@t@a\sc@t@b
70 \else
71 \ifsc@directing
72 \let\sc@t@a\sc@t@b
73 \fi
74 \fi
75 \sc@t@a
76 }
77 \newcommand\script {%
78 \MakeDirectTakeArg
79 \relax\begin{Script}%
80 }
81 \NewName{endscript} {} {%
82 \relax\end{Script}%
83 }
84 \NewName{script*} {} {%
85 \MakeDirectTakeNoArg
86 \relax\begin{Script}%
87 }
88 \NewName{endscript*} {} {%
89 \relax\end{Script}%
90 }
91 \newcommand\sc@make@begin@bit@D [2] {% args: bitstyle copystyle
92 \DefName{sc@begin@bit@#1@#2}{##1##2} {%
93 \def\BitStyle{#1}%
94 \sc@begin@bit@common{##1}{##2}%
95 \PreBitAll
96 %
97 % \ClaimXMarkTwo
98 % \FrankenInfo{drama}{The script environment is changing \protect\output.}%
99 % \output={\sc@outputwrapper}%
100 % \let\maxdepth\maxdimen
101 % \let\@maxdepth\maxdimen
102 \if@oneact@
103 \def\thepage {%
104 \arabic{bitbitbit}\hspace{\@ne em}\arabic{page}%
105 }%
106 \else
107 \def\thepage {%
108 \arabic{bitbit}--\arabic{bitbitbit}\hspace{\@ne em}\arabic{page}%
109 }%
110 \fi
111 \DramPer
112 %
113 \@nameuse{PreBit#1}%
114 }%
115 }
116 \newcommand\sc@make@end@bit@D [2] {% args: bitstyle copystyle
117 \DefName{sc@end@bit@#1@#2}{##1} {%
118 \sc@possiblyfinishdirecting
119 \TheEnd

```

```

120   \FrankenInfo{drama}{\protect\output is back to normal now.}%
121   }%
122 }
123 \NewBitstyle{Script}
124 \PromulgateComponentAcrossCopystyles{end@bit}{D}{Script}
125 \PromulgateComponentAcrossCopystyles{begin@bit}{D}{Script}
126
127 \newcommand\act {%
128   \sc@possiblyfinishdirecting
129   \sc@mark{\ShortEmpty}%
130   \stepcounter{bitbit}%
131   \pagebreak[3]%
132   \bigskip
133   \centerline{\relsize{2}Act \thebitbit}%
134   \par
135   \nobreak
136   \smallskip
137 }
138 \newcommand\scene {%
139   \sc@possiblyfinishdirecting
140   \sc@mark{\ShortEmpty}%
141   \stepcounter{bitbitbit}%
142   \pagebreak[3]%
143   \bigskip
144   \centerline{\relsize{1}Scene \thebitbitbit}%
145   \par
146   \nobreak
147   \smallskip
148 }

```

2.0.2 speakers and stage directions

```

149 \newlength{\halfacross}
150 \setlength{\halfacross}{\textwidth}
151 \divide\halfacross by 2
152 \newlength{\quarteracross}
153 \setlength{\quarteracross}{\textwidth}
154 \divide\quarteracross by 4
155
156 \newenvironment{DramaTab} [1] {%
157   \relax\begin{list}{} {%
158     \setlength{\leftmargin}{#1}%
159     \setlength{\listparindent}{\z@}%
160     \sloppy
161     }%
162   \item\relax
163 }{%
164 \relax\end{list}%
165 }
166 \newcommand\setstage [1] {%
167 \relax\begin{DramaTab}{\halfacross}%
168   \DirectStyle
169   \DirectOpen
170   \ignorespaces

```

```

171 #1%
172 \unskip
173 \DirectClose
174 \relax\end{DramaTab}%
175 }

```

It is a question of style whether to announce the speaker's name a second time, if his speech continues after a stage direction in a paragraph by itself. If the page breaks immediately before the stage direction, or during it, no continuation message will be given. If it breaks after the stage direction, and before the next speaker is announced with the `\speak` command, a continuation message appropriate to the speaker *before* the stage direction is given. Of course, sectioning commands will cancel any continuation messages.

Communicating via the mark is giving me a problem. I need to solve this problem and replace `@sc@bool` with mark information.

Also, when using the no-argument form on a source with arguments, you cannot swallow the spaces with `\ignorespaces`, because the open brace shields them.

There are problems with certain combinations of taking arguments and reannouncing speakers after a stage direction. I should draw a chart, and see how I can handle things.

```

\DirectOpen
\DirectClose 176 \newcommand\DirectOpen {\textrm{}}
177 \newcommand\DirectClose {\textrm{}}

178 \newboolean{sc@directing}
179
180 \newcommand*\DirectNoArg {%
181 \sc@noargs@finishdirecting
182 \sc@direct
183 }

184 \newcommand\DirectWithArg [1] {%
185 \sc@cargs@finishdirecting
186 \sc@direct
187 #1%
188 \sc@really@finishdirecting
189 }

190 \newcommand\sc@direct {%
191 \global\let\sc@lastmark\sc@themark
192 \sc@mark{\ShortEmpty}%
193 \sc@directingtrue
194 \pagebreak[0]% mild encouragement
195 \relax\begin{DramaTab}{\quarteracross}%
196 \DirectStyle
197 \DirectOpen
198 \ignorespaces
199 }

200 \newcommand\sc@really@finishdirecting {%
201 \unskip
202 \DirectClose
203 \relax\end{DramaTab}
204 \sc@directingfalse
205 % \sc@mark\expandafter{\sc@lastmark}% FIXME:
206 }

```

Looks like args version is null.

```
207 \ReserveCS\sc@args@finishdirecting
208 \newcommand\sc@noargs@finishdirecting {%
209   \ifsc@directing
210     \sc@really@finishdirecting
211   \fi
212 }
213 \ReserveCS\sc@possiblyfinishdirecting
214 \newcommand\MakeDirectTakeArg {%
215   \let\sc@possiblyfinishdirecting\sc@args@finishdirecting
216   \let\direct\DirectWithArg
217 }
218 \newcommand\MakeDirectTakeNoArg {%
219   \let\sc@possiblyfinishdirecting\sc@noargs@finishdirecting
220   \let\direct\DirectNoArg
221 }
```

To do: Do penalties better?

`\GiveSpeaker`

```
222 \newcommand\GiveSpeaker [1] {%
223   \par
224   \centerline{\textsc{#1}}%
225 }
```

`\GiveSpeakerContinued` The space is explicit for clarity.

```
226 \newcommand\GiveSpeakerContinued [1] {%
227   \GiveSpeaker{#1\space\rlap{\textsl{(continued)}}}%
228 }
229 \newcommand\speak [1] {%
230   \sc@possiblyfinishdirecting
231   \sc@mark{\ShortEmpty}%
232   \pagebreak[2]%
233   \smallskip
234   \sc@mark{\GiveSpeakerContinued{#1}}%
235   \GiveSpeaker{#1}%
236   \par
237   \noprogrambreak
238   \noindent
239 }
```


References

Knuth, Donald Ervin. 1986. *The T_EXbook*. 16th printing, revised. Volume A of *Computers & Typesetting*. Illustrated by Duane Bibby. Reading MA: Addison Wesley. First published in 1984.

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.

Symbols	E	L
<code>\@for</code> 47	<code>\E@cdr</code> 46	<code>\larger</code> 44
<code>\@ifundefined</code> 11	<code>\edef</code> 7, 46	<code>\leftmargin</code> 158
<code>\@maxdepth</code> 101	<code>\eExpand</code> 15	<code>\let</code> .. 26, 34, 69, 72, 100, 101, 191, 215, 216, 219, 220
<code>\@nameuse</code> 113	<code>\else</code> 43, 70, 106	<code>\listparindent</code> 159
<code>\@ne</code> 104, 108	<code>\end</code> 50, 82, 89, 164, 174, 203	<code>\LongEmpty</code> 43
<code>\@nil</code> 46	<code>\endcsname</code> 15, 36	
<code>\@</code> 48	<code>\endinput</code> 20	
	environments:	
A	<code>script</code> 2	M
<code>\act</code> 2, 127	<code>\expandafter</code> ... 36, 205	<code>\makeatletter</code> 10
<code>\addto@macro</code> 40		<code>\makeatother</code> 19
<code>\addtolength</code> 64	F	<code>\MakeDirectTakeArg</code> 78, 214
<code>\arabic</code> 104, 108	<code>\fi</code> . 53, 73, 74, 110, 211	<code>\MakeDirectTakeNoArg</code> 85, 218
	<code>\filedate</code> <u>1</u>	<code>\maxdepth</code> 100
B	<code>\fileinfo</code> <u>1</u>	<code>\maxdimen</code> 100, 101
<code>\begin</code> 45, 79, 86, 157, 167, 195	<code>\fileversion</code> <u>1</u>	
<code>\bigbreak</code> 52	<code>\FrankenInfo</code> ... 98, 120	N
<code>\bigskip</code> 132, 143	G	<code>\NeedsTeXFormat</code> ... 22
<code>\BitStyle</code> 93	<code>\GiveSpeaker</code> <u>222</u> , 227, 235	<code>\NewBitstyle</code> 123
C	<code>\GiveSpeakerContinued</code> <u>226</u>	<code>\newboolean</code> 178
<code>\centerline</code> 133, 144, 224	<code>\global</code> 191	<code>\NewCharacterAbbrev</code> 33, 36
<code>\character</code> 2, <u>31</u>	<code>\Gobble</code> 26	<code>\newcommand</code> 35, 42, 59, 77, 91, 116, 127, 138, 166, 176, 177, 180, 184, 190, 200, 208, 214, 218, 222, 226, 229
<code>\ClaimXMarkTwo</code> 97	H	<code>\newenvironment</code> ... 156
<code>\csname</code> 15, 36	<code>\halfacross</code> 149–151, 167	<code>\newlength</code> ... 149, 152
D	<code>\HaveECitationS</code> <u>1</u>	<code>\NewName</code> . 37, 81, 84, 88
<code>\DeclareBooleanOptions</code> 29	<code>\hspace</code> 104, 108	<code>\newtokens</code> 55
<code>\def</code> 1–6, 62, 93, 103, 107	I	<code>\NewUserAbbrevDefiner</code> 33
<code>\DefName</code> 92, 117	<code>\if@oneact@</code> 102	<code>\nobreak</code> 135, 146
<code>\direct</code> 2, 216, 220	<code>\ifsc@directing</code> 71, 209	<code>\noindent</code> 238
<code>\DirectClose</code> 173, <u>176</u> , 202	<code>\ifx</code> 43, 68	<code>\nopagebreak</code> 237
<code>\DirectNoArg</code> .. 180, 220	<code>\ignorespaces</code> . 170, 198	O
<code>\DirectOpen</code> 169, <u>176</u> , 197	<code>\In</code> 15	<code>\output</code> .. 56, 98, 99, 120
<code>\DirectStyle</code> .. 168, 196	<code>\InitCS</code> 51, 67	P
<code>\DirectWithArg</code> 184, 216	<code>\initelyHavECitationS</code> 3	<code>\pagebreak</code> 131, 142, 194, 232
<code>\divide</code> 151, 154	<code>\item</code> 162	
<code>\do</code> 47	J	
<code>\docdate</code> <u>1</u>	<code>\JusTLoaDInformatioN</code> 13	
<code>\DoXPackageS</code> 2	K	
<code>\DoXUsepackagE</code> <u>1</u>	<code>\kern</code> 65	
<code>\dp</code> 60		
<code>\DramPer</code> 42, 111		

<code>\par</code> .. 134, 145, 223, 236	<code>\sc@botmark</code> 63, 68	<code>\setstage</code> 2, 166
<code>\PPOptArg</code> 1, 23	<code>\sc@cast</code> . 31, 43, 46, 51	<code>\ShortEmpty</code> 68,
<code>\PreBitAll</code> 95	<code>\sc@direct</code> 182, 186, 190	129, 140, 192, 231
<code>\prevdepth</code> 64	<code>\sc@directingfalse</code> . 204	<code>\sloppy</code> 160
<code>\ProcessOptions</code> ... 30	<code>\sc@directingtrue</code> . 193	<code>\smallskip</code> 136, 147, 233
<code>\PromulgateComponentAcrossCores</code> .. 124, 125	<code>\sc@directingtrue</code> . 193	<code>\space</code> 8, 227
..... 124, 125	<code>\sc@len</code> .. 57, 60, 64, 65	<code>\speak</code> 38, 229
<code>\protect</code> 98, 120	<code>\sc@make@begin@bit@D</code> 91	<code>\SpeakAppend</code> 2, 31
<code>\providecommand</code> ... 31	<code>\sc@make@end@bit@D</code> . 116	<code>\stepcounter</code> .. 130, 141
<code>\providelength</code> 57	<code>\sc@mark</code> 26, 129, 140,	
<code>\ProvidesPackage</code> .. 23	192, 205, 231, 234	T
	<code>\sc@noargs@finishdirecting</code>	<code>\textrm</code> 176, 177
Q 181, 208, 219	<code>\textsc</code> 224
<code>\quarteracross</code>	<code>\sc@outputwrapper</code> 59, 99	<code>\textsl</code> 227
..... 152–154, 195	<code>\sc@possiblyfinishdirecting</code>	<code>\textwidth</code> ... 150, 153
	. 118, 128, 139,	<code>\the</code> 61
R	213, 215, 219, 230	<code>\thebitbit</code> 133
<code>\relax</code> 45, 50,	<code>\sc@really@finishdirecting</code>	<code>\thebitbitbit</code> 144
79, 82, 86, 89, 188, 200, 210	<code>\TheEnd</code> 119
157, 162, 164,	<code>\sc@t@a</code> 46,	<code>\thepage</code> 103, 107
167, 174, 195, 203	47, 67, 69, 72, 75	<code>\TMFontCharacter</code> .. 34
<code>\relsize</code> 133, 144	<code>\sc@t@b</code> 47, 48, 62, 69, 72	U
<code>\RequirePackage</code> . 24, 25	<code>\sc@themark</code> 191	<code>\UndefineCS</code> 13
<code>\ReserveCS</code> 32, 207, 213	<code>\sc@toks@saves@output</code>	<code>\unskip</code> 172, 201
<code>\RestoreDoXVarS</code> ... 18 55, 61	<code>\usepackage</code> 16
<code>\rlap</code> 227	<code>\scene</code> 2, 138	
	<code>\script</code> 77	V
S	<code>script (environment)</code> . 2	<code>\VerboseErrors</code> 27
<code>\SaveDoXVarS</code> 14	<code>\scshape</code> 34, 44	
<code>\sc@args@finishdirecting</code>	<code>\setlength</code> 60,	Z
..... 185, 207, 215	150, 153, 158, 159	<code>\z@</code> 159
<code>\sc@begin@bit@common</code> 94		