

# The `fix2col` package\*

David Carlisle†

1998/08/17

## 1 Introduction

This package makes two independent changes to L<sup>A</sup>T<sub>E</sub>X's two column output routine to fix the following two longstanding 'features'.

- If the T<sub>E</sub>X mark system is used (for example using the 'headings' page style in the standard L<sup>A</sup>T<sub>E</sub>X classes) then any marks that originate on the first column are 'lost' as L<sup>A</sup>T<sub>E</sub>X constructs the second column. An example document showing how this can result in incorrect page headings may be found in the latex bug database:  
<http://www.uni-mainz.de/cgi-bin/ltxbugs2html?pr=latex/2613>
- The second feature is documented in the L<sup>A</sup>T<sub>E</sub>X book. By default L<sup>A</sup>T<sub>E</sub>X does not attempt to keep double and single column floats in sequence, so if 'Figure 1' is a double column float produced with `figure*`, then it may float after 'Figure 2' if that is a single column, `figure`, float. Further correspondence about this may also be found in the bug database:  
<http://www.uni-mainz.de/cgi-bin/ltxbugs2html?pr=latex/2346>

## 2 Notes on the Implementation Strategies

### 2.1 Preserving Marks

The standard L<sup>A</sup>T<sub>E</sub>X twocolumn system works internally by making each column a separate 'page' that is passed independently to T<sub>E</sub>X's pagebreaker. (Unlike say the `multicol` package, where all columns are gathered together and then split into columns later, using `\vsplit`.) This means that the primitive T<sub>E</sub>X marks that are normally used for header information, are globally reset after the first column. By default L<sup>A</sup>T<sub>E</sub>X does nothing about this. A good solution is provided by Piet van Oostrum (building on earlier work of Joe Pallas) in his `fixmarks` package.

After the first column box has been collected the mark information for that box is saved, so that any `\firstmark` can be 'artificially' used to set the page-level marks after the second column has been collected. (The second column

---

\*This file has version number v0.03, last revised 1998/08/17.

†Part one is essentially a copy of the `fixmarks` package by Piet van Oostrum, itself based on earlier work by Joe Pallas. Part two is loosely based on the `fixfloats` package, originally by Ed Szynter, with some modifications by Bil Kleb.

`\firstmark` is not normally required.) Unfortunately  $\TeX$  does not provide a direct way of knowing if any marks are in the page, `\firstmark` always has a value from previous pages, even if there is no mark in this page. The solution is to make a copy of the box and then `\vsplit` it so that any marks show up as `\splitfirstmark`.

The use of `\vsplit` does mean that the output routine will globally change the value of `\splitfirstmark` and `\splitbotmark`. The `fixmarks` package goes to some trouble to save and restore these values so that the output routine does *not* change the values. This part of `fixmarks` is not copied here as it is quite costly (having to be run on every page) and there is no reason why anyone writing code using `\vsplit` should allow the output routine to be triggered before the split marks have been accessed.

## 2.2 Preserving Float Order

The standard output routine maintains two lists of floats that have been ‘deferred’ for later consideration. One list for single column floats, and one for double column floats (which are always immediately put onto their deferred list). This mechanism means that  $\LaTeX$  ‘knows’ which type of float is contained in each box by the list that it is processing, but having two lists means that there is no mechanism for preserving the order between the floats in each list.

The solution to this problem consists of two small changes to the output routine.

Firstly, abandon the ‘double column float list’ `\@dbldeferlist` and change every command where it is used so that instead the same `\@deferlist` is used as for single column floats. That one change ensures that double and single column floats stay in the same sequence, but as  $\LaTeX$  no longer ‘knows’ whether a float is double or single column, it will happily insert a double float into a single column, overprinting the other column, or the margin.

The second change is to provide an alternative mechanism for recording the two column floats.  $\LaTeX$  already has a compact mechanism for recording float information, an integer count register assigned to each float records information about the ‘type’ of float ‘figure’, ‘table’ and the position information ‘htp’ etc.

The type information is stored in the ‘high’ bits, one bit position (above ‘32’) allocated to each float type. The ‘low’ bits store information about the allowed positions, one bit each allocated for `h t b p`. In the  $\LaTeX$ 2.09 system, the bit corresponding to ‘16’ formed a ‘boundary’ between these two sets of information, and it was never actually used by the system. Ed Sznyter’s `fixfloats` package not unreasonably used this position to store the double column information, setting the bit for double column floats. Then at each point in the output routine at which a float is committed to a certain region, an additional check must be made to check that the float is (or is not) double column. If it spans the wrong number of columns it is deferred rather than being added.

Unfortunately the bit ‘16’ is not available in  $\LaTeX$ 2 $\epsilon$ . It is used to encode the extra float position possibility ‘!’ that was added in that system. It would be possible to use position ‘32’ and to move the flags for ‘table’, ‘figure’,... up one position, to start at 64, but this would mean that in principle one less float type would be supported, and more importantly is likely to break any other packages that assume anything about the output routine internals. So here I instead use another mechanism for flagging double column floats: By default all floats have

depth 0pt. This package arranges that double column ones have depth 1sp. This information may then be used in the same manner as in the `fixfloats` package, to defer any floats that are not of the correct column spanning type.

Use of the package showed that one also has to change the way  $\LaTeX$  handles star-form floats: if they are immediately deferred (as done normally) certain situations can still result in the float sequence getting out of order. This happens when a floats are placed in the middle of a paragraph. In that case the wide float is deferred immediately while a column wide float early on in the same paragraph might not be handled until the end of the paragraph when it is finally seen by the output routine. Since by that time the wide float is already on the `\@deferlist` the column float will also end up there (which is not only incorrect because it may have fitted onto the page but also because it is then placed at the end of this list). Version v0.03 now fixes this problem.

### 3 Implementation

```
1 (*package)
```

#### 3.1 Preserving Marks

This is just a change to the single command `\@outputdblcol` so that it saves mark information for the first column and restores it in the second column.

```
2 \def\@outputdblcol{%
3   \if@firstcolumn
4     \global\@firstcolumnfalse
```

Save the left column

```
5   \global\setbox\@leftcolumn\copy\@outputbox
```

Remember the marks from the first column

```
6   \splitmaxdepth\maxdimen
7   \vbadness\maxdimen
8   \setbox\@outputbox\vsplit\@outputbox to\maxdimen
```

One minor difference from the current `fixmarks`, pass the marks through a token register to stop any `#` tokens causing an error in a `\def`.

```
9   \toks@\expandafter{\topmark}%
10  \xdef\@firstcoltopmark{\the\toks@}%
11  \toks@\expandafter{\splitfirstmark}%
12  \xdef\@firstcolfirstmark{\the\toks@}%
```

This test does not work if truly empty marks have been inserted, but  $\LaTeX$  marks should always have (at least) two brace groups. (Except before the first mark is used, when the marks are empty, but that is OK here.)

```
13  \ifx\@firstcolfirstmark\@empty
14    \global\let\@setmarks\relax
15  \else
16    \gdef\@setmarks{%
17      \let\firstmark\@firstcolfirstmark
18      \let\topmark\@firstcoltopmark}%
19  \fi
```

End of change

```
20  \else
```

```

21 \global\@firstcolumntrue
22 \setbox\@outputbox\vbox{%
23 \hb@xt@\textwidth{%
24 \hb@xt@\columnwidth{\box\@leftcolumn \hss}%
25 \hfil
26 \vrule \@width\columnseprule
27 \hfil
28 \hb@xt@\columnwidth{\box\@outputbox \hss}}}%
29 \@combinedblfloats

```

Override current first and top with those of first column if necessary

```

30 \setmarks

```

End of change

```

31 \@outputpage
32 \begingroup
33 \@dblfloatplacement
34 \@startdblcolumn
35 \@whiles\if@fcolmade \fi{\@outputpage\@startdblcolumn}%
36 \endgroup
37 \fi}

```

## 3.2 Preserving Float Order

Changes `\@dbldeferlist` to `\@deferlist` are not explicitly noted but are flagged by blank comment lines around the changed line.

```

38 \def\end@dblfloat{%
39 \if@twocolumn
40 \endfloatbox
41 \ifnum\@floatpenalty <\z@
42 \@largefloatcheck

```

Force the depth of two column float boxes.

```

43 \global\dp\@currbox1sp %

```

Next line assumes that first token of `\end@float` is `\endfloatbox` so we gobble that.

```

44 % \@cons\@deferlist\@currbox
45 \expandafter\@gobble\end@float

```

`\@Esphack` is then added by `\endfloat` above.

```

46 \fi
47 % \ifnum \@floatpenalty ==-\@Mii \@Esphack\fi
48 \else
49 \end@float
50 \fi
51 }

```

Test if the float box has the wrong width. (Actually as noted above the test is for a conventional depth setting rather than for the width of the float).

```

52 \def\@testwrongwidth #1{%
53 \ifdim\dp#1=\f@depth
54 \else
55 \global\@testtrue
56 \fi}

```

Normally looking for single column floats, which have zero depth.

```
57 \let\f@depth\z@
```

but when making two column float area, look for floats with 1sp depth.

```
58 \def\@dblfloatplacement{\global\@dbltopnum\c@dbltopnumber
59   \global\@dbltoproom \dbltopfraction\@colht
60   \@textmin \@colht
61   \advance \@textmin -\@dbltoproom
62   \@fpmin \dblfloatpagefraction\textheight
63   \@fptop \@dblftop
64   \@fpsep \@dblfpsep
65   \@fpbot \@dblfpbot
66   \def\f@depth{1sp}}
```

All the remaining changes are replacing the double column defer list or inserting the extra test `\@testwrongwidth{<box>}` at suitable places. That is at places where a box is taken off the deferlist.

```
67 \def \@docclearpage {%
68   \ifvoid\footins
69     \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
70     \setbox\@tempboxa\box\@cclv
71     \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
72     \global \let \@toplist \@empty
73     \global \let \@botlist \@empty
74     \global \@colroom \@colht
75     \ifx \@currlist\@empty
76     \else
77       \@latexerr{Float(s) lost}\@ehb
78       \global \let \@currlist \@empty
79     \fi
80     \@makefcolumn\@deferlist
81     \@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
82     \if@twocolumn
83       \if@firstcolumn
84         \xdef\@deferlist{\@dbltoplist\@deferlist}%
85         \global \let \@dbltoplist \@empty
86         \global \@colht \textheight
87         \begingroup
88           \@dblfloatplacement
89           \@makefcolumn\@deferlist
90           \@whilesw\if@fcolmade \fi{\@outputpage
91             \@makefcolumn\@deferlist}%
92         \endgroup
93       \else
94         \vbox{}\clearpage
95       \fi
96     \fi
```

the next line is needed to avoid losing floats in certain circumstances a single call to the original `\docclearpage` will now no longer output all floats.

```
97   \ifx\@deferlist\@empty \else\clearpage \fi
98   \else
```

```

99     \setbox\@cclv\vbox{\box\@cclv\vfil}%
100    \@makecol\@opcol
101    \clearpage
102    \fi
103 }

104 \def \@startdblcolumn {%
105   \@tryfcolumn \@deferlist
106   \if@fcolmade
107   \else
108     \begingroup
109     \let \reserved@b \@deferlist
110     \global \let \@deferlist \@empty
111     \let \@elt \@sdblcolelt
112     \reserved@b
113   \endgroup
114   \fi
115 }

116 \def \@addtonextcol{%
117   \begingroup
118   \@insertfalse
119   \@setfloattypecounts
120   \ifnum \@fpstype=8
121   \else
122     \ifnum \@fpstype=24
123     \else
124       \@flsettextmin
125       \@reqcolroom \ht\@currbox
126       \advance \@reqcolroom \@textmin
127       \ifdim \@colroom>\@reqcolroom
128         \@flsetnum \@colnum
129         \ifnum \@colnum>\z@
130           \@bitor\@currtype\@deferlist
131           \@testwrongwidth\@currbox
132           \if@test
133           \else
134             \@addtotoporbot
135           \fi
136         \fi
137       \fi
138     \fi
139   \fi
140   \if@insert
141   \else
142     \@cons\@deferlist\@currbox
143   \fi
144   \endgroup
145 }

146 \def \@addtodblcol{%
147   \begingroup
148   \@insertfalse
149   \@setfloattypecounts
150   \@getfpsbit \tw@
151   \ifodd\@tempcnta

```

```

152 \@flsetnum \@dbltopnum
153 \ifnum \@dbltopnum>\z@
154 \@tempswafalse
155 \ifdim \@dbltoproom>\ht\@currbox
156 \@tempswatru
157 \else
158 \ifnum \@fpstype<\sixt@n
159 \advance \@dbltoproom \@textmin
160 \ifdim \@dbltoproom>\ht\@currbox
161 \@tempswatru
162 \fi
163 \advance \@dbltoproom -\@textmin
164 \fi
165 \fi
166 \if@tempswa
167 \@bitor \@currtype \@deferlist

not in fixfloats?
168 \@testwrongwidth\@currbox
169 \if@test
170 \else
171 \@tempdima -\ht\@currbox
172 \advance\@tempdima
173 -\ifx \@dbltoplist\@empty \dbltextfloatsep \else
174 \dblfloatsep \fi
175 \global \advance \@dbltoproom \@tempdima
176 \global \advance \@colht \@tempdima
177 \global \advance \@dbltopnum \m@ne
178 \@cons \@dbltoplist \@currbox
179 \@inserttrue
180 \fi
181 \fi
182 \fi
183 \fi
184 \if@insert
185 \else
186 \@cons\@deferlist\@currbox
187 \fi
188 \endgroup
189 }

190 \def \@addtocurcol {%
191 \@insertfalse
192 \@setfloattypecounts
193 \ifnum \@fpstype=8
194 \else
195 \ifnum \@fpstype=24
196 \else
197 \@flsettextmin
198 \advance \@textmin \@textfloatsheight
199 \@reqcolroom \@pageht
200 \ifdim \@textmin>\@reqcolroom
201 \@reqcolroom \@textmin
202 \fi
203 \advance \@reqcolroom \ht\@currbox

```

```

204     \ifdim \@colroom>\@reqcolroom
205     \@flsetnum \@colnum
206     \ifnum \@colnum>\z@
207     \@bitor\@currtype\@deferlist

We need to defer the float also if its width doesn't fit.

208     \@testwrongwidth\@currbox
209     \if@test
210     \else
211     \@bitor\@currtype\@botlist
212     \if@test
213     \@addtobot
214     \else
215     \ifodd \count\@currbox
216     \advance \@reqcolroom \intextsep
217     \ifdim \@colroom>\@reqcolroom
218     \global \advance \@colnum \m@ne
219     \global \advance \@textfloatsheight \ht\@currbox
220     \global \advance \@textfloatsheight 2\intextsep
221     \@cons \@midlist \@currbox
222     \if@nobreak
223     \nobreak
224     \@nobreakfalse
225     \everypar{}%
226     \else
227     \addpenalty \interlinepenalty
228     \fi
229     \vskip \intextsep
230     \box\@currbox
231     \penalty\interlinepenalty
232     \vskip\intextsep
233     \ifnum\outputpenalty <-\@Mii \vskip -\parskip\fi
234     \outputpenalty \z@
235     \inserttrue
236     \fi
237     \fi
238     \if@insert
239     \else
240     \@addtotoporbot
241     \fi
242     \fi
243     \fi
244     \fi
245     \fi
246     \fi
247     \fi
248     \if@insert
249     \else
250     \@resethfps
251     \@cons\@deferlist\@currbox
252     \fi
253 }

254 \def\@xtryfc #1{%
255     \@next\reserved@a\@trylist{#1}%

```



```

256 \@currtype \count #1%
257 \divide\@currtype\@xxxii
258 \multiply\@currtype\@xxxii
259 \@bitor \@currtype \@failedlist
260 \@testfp #1%

261 \@testwrongwidth #1%

262 \ifdim \ht #1>\@colht
263     \@testtrue
264 \fi
265 \if@test
266     \@cons\@failedlist #1%
267 \else
268     \@ytryfc #1%
269 \fi}

270 \def\@ztryfc #1{%
271     \@tempcnta\count #1%
272     \divide\@tempcnta\@xxxii
273     \multiply\@tempcnta\@xxxii
274     \@bitor \@tempcnta {\@failedlist \@flfail}%
275     \@testfp #1%

    not in fixfloats?

276     \@testwrongwidth #1%

277     \@tempdimb\@tempdima
278     \advance\@tempdimb\ht #1%
279     \advance\@tempdimb\@fpsep
280     \ifdim \@tempdimb >\@colht
281         \@testtrue
282     \fi
283     \if@test
284         \@cons\@flfail #1%
285     \else
286         \@cons\@flsucceed #1%
287         \@tempdima\@tempdimb
288     \fi}

289 </package>

```