

# eqexam: An Exam Construction Package

D. P. Story  
Email: [dpstory@uakron.edu](mailto:dpstory@uakron.edu)

Processed January 7, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Package options and Process Options</b>	<b>4</b>
2.1	Early inclusion of packages . . . . .	4
2.2	Options New to eqexam . . . . .	4
2.3	Options from and for web . . . . .	9
2.4	Options from and for exerquiz . . . . .	9
2.4.1	Title page options . . . . .	10
2.5	Drivers Recognized . . . . .	11
2.6	Bring in Config Files . . . . .	12
2.7	Process Options . . . . .	12
2.8	Save Switch Values . . . . .	15
<b>3</b>	<b>Required Packages</b>	<b>15</b>
<b>4</b>	<b>Page Layout</b>	<b>18</b>
<b>5</b>	<b>Counters, Lengths and Tokens</b>	<b>19</b>
<b>6</b>	<b>Some Macros to Support the Options</b>	<b>20</b>
<b>7</b>	<b>Colors</b>	<b>21</b>
<b>8</b>	<b>Version Control</b>	<b>22</b>
<b>9</b>	<b>Title Definitions from Web</b>	<b>30</b>
<b>10</b>	<b>Identification Information</b>	<b>31</b>
<b>11</b>	<b>Stand alone Code</b>	<b>39</b>
<b>12</b>	<b>Switching proofing symbols</b>	<b>39</b>

<b>13 The Main Code</b>	<b>41</b>
13.1 Running Heads and Feet . . . . .	44
13.2 <code>\maketitle</code> definitions . . . . .	46
13.3 The cover page definitions . . . . .	47
13.4 Insert Points in Margins and Compute Page Totals . . . . .	52
13.5 Computing Number of Points within a Segment of the Exam . . . . .	58
13.6 Useful Commands to Write Problems . . . . .	61
13.6.1 The <code>\placeAtxy</code> command . . . . .	61
13.6.2 The <code>workarea</code> environment . . . . .	61
13.6.3 The <code>splitsolution</code> environment . . . . .	62
13.7 The <code>exam</code> Environment . . . . .	71
13.8 <code>problem</code> Environments . . . . .	77
<b>14 Vertical Space Filling Options</b>	<b>97</b>
14.1 General filler lines commands and controls . . . . .	97
14.2 Utility commands used with the <code>flexextended</code> option . . . . .	111
14.3 Filler lines with <code>answerkey</code> option . . . . .	114
14.4 Filler lines with <code>nosolutions</code> option . . . . .	116
<b>15 Concerning the <code>fortextbook</code> option</b>	<b>118</b>
15.1 Setting options with <code>\textbookOpts</code> . . . . .	119
15.2 Macros to display answers/shortsolns . . . . .	121
15.3 Marginal Matter. . . . .	129
15.4 In support of solutions at end of document and chapter . . . . .	137
15.5 Modifying and restoring the Layout . . . . .	138
15.6 We shipout in support of <code>fortextbook</code> . . . . .	140
15.7 Modify <code>equequestions</code> environment . . . . .	140
15.8 Modifications for solutions page . . . . .	141
15.9 Some Convenience/Formatting Commands . . . . .	143
15.10The <code>probset</code> and <code>example</code> environments . . . . .	144
15.11Commands in support of Solution Manuals . . . . .	145
<b>16 <code>fortextbook</code> Style File</b>	<b>147</b>
<b>17 <code>xkeyval</code> Extensions</b>	<b>148</b>
<b>18 Index</b>	<b>160</b>

## 1 Introduction

This package is my attempt at writing a set of macros for creating exams. The package can be used with or without `web` or `exerquiz`. When used without, what is produced is a document suitable for printing. When `exerquiz` is used, the same document is produced, but with hypertext links to solutions. This may be useful for publishing solutions on the web, or publishing pretests with/without solutions.

The package must be as flexible as possible vis-à-vis PDF. (1) **web** only does not add much, it does input hyperref and test info such as `\title`, `\subject`, etc are placed in the Document Info fields of the PDF. (2) If **exerquiz** is also input (prior to **eqexam**), then hyperlinks will be created to the solutions to the test, if solutions are included at the end of the document. (3) If **web** and **exerquiz** are input, and the **online** option is taken, then the checkboxes will be come interactive, the space left to work the problems will be multi-line text boxes, fill-ins limited to True/False and simple text fill-in the blank will also become text boxes.

When in **online** mode, the student can take the test in a computer lab, the completed exam can be printed and handed in, or perhaps submitted server-side script.

(2011/05/13) The version of **eqexam** is a departure from previous versions. Previously, the list of problems were not in list, they were left-justified, with the problem number extending out into a little area determined by `\oddsidemargin`. This makes it hard to reformat a list of problems to fit into a custom book format. This new version defines a new environment, **equestions**, that makes each problem into a list. The list environment allows for an easy redesign of the formatting of the problems. The purpose of this new scheme, is to open up **eqexam** as a format package that can be used by author for writing a textbook.

The **fortextbook** option supplies support for authors writing a textbook. The **exam** environment is re-cast into the **probset** environment, it can be used to write problem sets within the text. there is a version for the instructor and the student. The instructors version writes answers to the problem sets to the margins (or inline). Two solutions are offered, short and long. The short solutions appear in the back of the book (odd-numbered ones for the student edition. The long solutions are used to build the stand-alone solutions manuals for both the student and the instructor. More details are found in ‘**Concerning the fortextbook option**’ on page 118, see also ‘**fortextbook Style File**’ on page 147.

## 2 Package options and Process Options

Let us catalog the options of this package.

### 2.1 Early inclusion of packages

We include these packages early in the process of loading.

```
1 <*package>
2 \RequirePackage{ifpdf}[2006/02/20]
3 \RequirePackage{ifxetex}[2006/08/21]
```

Very useful package for defining key-values.

```
4 \usepackage{xkeyval}
```

### 2.2 Options New to eqexam

`usecustomdesign` Here are some options unique to this package. Use this option to avoid `eqexam` from setting up the “standard” page layout.

```
5 \DeclareOptionX{usecustomdesign}{\eqcustomdesigntrue}
6 \newif\ifeqcustomdesign \eqcustomdesignfalse
7 \DeclareOptionX{nocustomdesign}{\let\eqe@nocustomdesign=1}
8 \let\eqe@nocustomdesign=0
```

`fortextbook`

An option to extend the application of `eqexam` to provide support (exercises, providing solutions, short solutions, answers, and hints) for authors writing textbooks. See ‘[Concerning the `fortextbook` option](#)’ on page 118.

```
9 \DeclareOptionX{fortextbook}{\eqfortextbooktrue}
10 \newif\ifeqfortextbook \eqfortextbookfalse
```

`forinstr`

`forstudent`

These two options simply set a switch to signal the intention of the document author.

```
11 \DeclareOptionX{forinstr}{\eqforinstrtrue}
12 \DeclareOptionX{forstudent}{\eqforinstrfalse}
13 \newif\ifeqforinstr \eqforinstrfalse
```

`nomarginwrite`

The switch `\ifeqwritetomargins` is used by the `fortextbook` option. It is normally `true`, but if set to `false`, the `\AddToShipoutPicture` is not generated at the beginning of the document. Here is the code taken from below:

```
\ifeqfortextbook\ifeqwritetomargins
\AtBeginDocument{\chkmarginboxwidth
\AddToShipoutPicture{\eqe@tb@shipout}}
\fi\fi
```

Using this option, the check for the margin width is not done, and writing to the margins is turned off. (`\marginpar` still works)

```
14 \DeclareOptionX{nomarginwrite}{\eqwritetomarginsfalse}
15 \newif\ifeqwritetomargins \eqwritetomarginstrue
```

**Configuration Files.** This section contains options for the configuration files.

**cfg** The `cfg` option is used to specify a named configuration file, extension must be `.cfg`; usage `cfg=hwdoc`.

```

16 \define@key{eqexam.sty}{cfg}[]{%
17   \def\arg@i{#1}\ifx\arg@i\@empty
18     \PackageWarning{eqexam}{No value for 'cfg' specified}\else
19     \def\ifeqexamCFG{true}\def\eqexamCFG{#1.cfg}%
20     \AtEndOfPackage{\InputIfFileExists{#1.cfg}
21       {\typeout{Inputting #1.cfg}}{\PackageWarning{eqexam}{%
22         Cannot find configuration file #1.cfg}}}\fi
23   }
24 \def\ifeqexamCFG{false}
25 \let\eqexamCFG\@empty

```

**myconfig**

**myconfigi...myconfigvi**

We offer seven sets of configuration files, that should be enough, especially light of the new `cfg` option, defined above.

```

26 \@for\eqe@tmp@i:={,i,ii,iii,iv,v,vi}\do{%
27   \edef\eqe@tmp@exp{%
28     \noexpand\DeclareOptionX{myconfig\eqe@tmp@i}%
29     {\noexpand\AtEndOfPackage{\expandafter\noexpand
30       \csname eqemyconfig\eqe@tmp@i\endcsname}}}%
31   }\eqe@tmp@exp
32 }

```

**Point options.** Options relating to points, points on left, right, both, no points, totals on left and right. We offer options for points and totals.

```

pointsonleft 23 \DeclareOptionX{pointsonleft}{\AtEndOfPackage{\PointsOnLeft}}
pointsonright 34 \DeclareOptionX{pointsonright}{\AtEndOfPackage{\PointsOnRight}}
pointsonboth 35 \DeclareOptionX{pointsonboth}{\AtEndOfPackage{\PointsOnBothSides}}
nopoints 36 \DeclareOptionX{nopoints}{\AtEndOfPackage{\NoPoints}}
totalsonleft 37 \DeclareOptionX{totalsonleft}{\AtEndOfPackage{\TotalsOnLeft}}
totalsonright 38 \DeclareOptionX{totalsonright}{\AtEndOfPackage{\TotalsOnRight}}
nozerototals 39 \DeclareOptionX{nozerototals}{\AtEndOfPackage{\noZeroTotals}}

```

**nototals**

**noparttotals**

**parttotalsonright**

**parttotalsonleft**

**noseparationrule**

**nosummarytotals**

**Totals options.** Options relating to totals

```

40 \let\eqe@YES=y \let\eqe@NO=n
41 \let\eqe@One=1 \let\eqe@Zero=0
42 \let\eqe@Two=2 \let\eqe@Three=3 \let\eqe@Four=4
43 \def\eqe@h{h}
44 \DeclareOptionX{nototals}{\AtEndOfPackage{\NoTotals}}
45 \DeclareOptionX{noparttotals}{%
46   \AtEndOfPackage{\let\eqe@parttotals\eqe@NO}}
47 \DeclareOptionX{parttotalsonright}{%
48   \def\eqeomarginbox{\eqeomarginboxright}}
49 \DeclareOptionX{parttotalsonleft}{%
50   \def\eqeomarginbox{\eqeomarginboxleft}}
51 \def\eqeomarginbox{\eqeomarginboxright}
52 \DeclareOptionX{noseparationrule}{%

```

```

53 \AtEndOfPackage{\separationruleOff}}
54 % \AtEndOfPackage{\let\separationrule\relax}}
55 \DeclareOptionX{nosummarytotals}{\let\eq@nosummarytotals\eqe@YES}

```

**cover page options.** There are two such options, `coverpage` and `coverpagesumry`. If this option is taken, a cover page is generate.

`coverpage`

```

56 \DeclareOptionX{coverpage}{\def\eqex@coverpage{\eqexcoverpage}%
57 \setcounter{page}{0}}

```

`coverpagesumry`

If this option is taken, an **Exam Record** is generated on the cover page, provided the `coverpage` option is taken. Possible values aer `byparts`, `bypages`, or `none`.

```

58 \define@choicekey+{eqexam.sty}{coverpagesumry}[\val\nr]%
59 {byparts,bypages,none}{%
60 \ifcase\nr\relax
61 \def\sumryAnnots{\cpSumrybyparts}\or
62 \def\sumryAnnots{\cpSumrybypages}\or
63 \let\sumryAnnots\relax
64 \fi
65 }{\PackageWarning{aeb}{Bad choice for coverpagesumry, permissible values
66 are byparts, bypages, and none. Try again}}
67 \let\sumryAnnots\relax

```

`nospacetowork`

**Options related to how the document is built.** The vertical space defined by the solution environment is removed.

```

68 \DeclareOptionX{nospacetowork}{%
69 \AtEndOfPackage{\let\eq@insertverticalspace\eqe@NO}}

```

`answerkey`

Equivalent to `solutionsafter` and `proofing`.

```

70 \@ifundefined{ifanswerkey}{\newif\ifanswerkey\answerkeyfalse}{}
71 \@ifundefined{ifsolutionsAtEnd}{}{}
72 {\newif\ifsolutionsAtEnd\solutionsAtEndtrue}{}

```

A more intelligent processing of certain options is deployed: Now, we allow only one of the options `answerkey`, `nosolutions`, `vspacewithsolns`, `solutionsafter`, and `solutionsonly` at a time.

```

73 \let\thisOpt@OK \eqe@YES
74 \def\w@rningBadOpts{\PackageWarningNoLine{eqexam}{You've chosen two
75 conflicting options,\MessageBreak \l@stPO\space and \CurrentOption.
76 We will recognize\MessageBreak the \l@stPO\space option.
77 Please correct\MessageBreak this if my guess is wrong}}
78 \def\eqe@optiont@kenMsg{\PackageInfo{eqexam}
79 {Option \CurrentOption\space taken}}
80 \DeclareOptionX{answerkey}{\ifx\thisOpt@OK\eqe@YES
81 \def\l@stPO{answerkey}\let\thisOpt@OK\eqe@NO
82 \eqe@optiont@kenMsg\expandafter\AnswerKey\else
83 \w@rningBadOpts\fi}
84 \def\AnswerKey{\solutionsAtEndfalse\answerkeytrue
85 \eq@proofingtrue\eq@solutionsaftertrue\displayworkareafalse}

```

**vspacewithsolns** When `vspacewithsolns` is used, vertical space is created by the solutions environment, and the solutions are written to the end of the file. Added `ftbsolns` as an alias for `vspacewithsolns`. This is implemented through a Boolean switch `\ifvspacewithsolns`, which I'm sorry now I've made so long. To make up for that bad decision, I also define `\fkeyalt` to be `\ifvspacewithsolns` with making conditional decisions, of course, one cannot say `\keyalttrue`.

```

86 \newif\ifvspacewithsolns\vspacewithsolnsfalse
87 \def\ifkeyalt{\csname ifvspacewithsolns\endcsname}
88 \let\keyalttrue\vspacewithsolnstrue
89 \let\keyaltfalse\vspacewithsolnsfalse
90 %\def\ifkeyalt{\csname ifvspacewithsolns\endcsname}
91 \def\ifkeyOrkeyalt{\ifanswerkey
92   \def\eqe@next{\csname iftrue\endcsname}\else
93   \ifvspacewithsolns\def\eqe@next{\csname iftrue\endcsname}\else
94   \def\eqe@next{\csname iffalse\endcsname}\fi\fi\eqe@next}
95 \DeclareOptionX{vspacewithsolns}{\ifx\thisOpt@OK\eqe@YES
96   \def\lstPO{vspacewithsolns}\let\thisOpt@OK\eqe@NO
97   \eqe@option@kenMsg
98   \vspacewithsolnstrue\expandafter\displayworkareatru
99   \else\warningBadOpts\fi}
100 \DeclareOptionX{ftbsolns}{\ExecuteOptionsX{vspacewithsolns}}

```

**flexextended** The `flexextended` is an (experimental) option to allow filler lines (fl) to be superimposed under the solutions, when the `answerkey` option is taken.

```

101 \DeclareOptionX{flexextended}{\AtEndOfPackage{\flexextendedInput}}
102 \def\flexextendedInput{\eqe@flexextendedtrue
103   \InputIfFileExists{flexextended.def}
104     {\PackageInfo{eqexam}{Inputting flexextended.def for
105       flexextended\MessageBreak option}}
106     {\eqe@flexextendedfalse\PackageWarning{eqexam}
107       {Cannot find flexextended.def for flexextended\MessageBreak
108       option. You should rebuild the package. Removing\MessageBreak
109       flexextended for now}}}
110 \@ifundefined{ifeqe@flexextended}
111   {\newif\ifeqe@flexextended \eqe@flexextendedfalse}{}

```

**useforms** Use forms (if online option is taken); otherwise draw rectangles for multiple choice/multiple selection questions.

```

112 \DeclareOptionX{useforms}{\AtEndOfPackage{\def\sqstar{*}}}

```

**allowcirc4mc** This option uses `lcircle10` to draw circles around multiple choice questions.

```

113 \newif\ifallowcircmc \allowcircmcfalse
114 \DeclareOptionX{allowcirc4mc}{\allowcircmctrue}

```

**online PDF Options** The various options to go beyond paper!

**pdf** Options related to the interactive capability of `eqexam`. For each of the PDF

**links** options, we auto-check for `pdftex` and `xetex`.

```

115 \def\eqe@auto@chk@drivers{\ifpdf\ExecuteOptionsX{pdftex}\else

```

**email**

```

116 \ifxetex\ExecuteOptionsX{xetex}\fi\fi}
117 \newif\ifeqonline \eqonlinefalse
118 \DeclareOptionX{online}{\eq@auto@chk@drivers
119 \let\eq@online\eq@YES\eqonline>true\ExecuteOptionsX{links}}
120 \DeclareOptionX{pdf}{\eq@auto@chk@drivers\let\load@web\eq@YES}
121 \DeclareOptionX{links}{\eq@auto@chk@drivers
122 \let\load@web\eq@YES\let\load@exerquiz\eq@YES}
123 \DeclareOptionX{email}{\eq@auto@chk@drivers
124 \let\use@email\eq@YES\ExecuteOptionsX{online}}

```

**obeylocalversions** This option is used for multiple versions of a document.

```

125 \newif\ifeqobeylocalversion \eqobeylocalversionfalse
126 \DeclareOptionX{obeylocalversions}{\eqobeylocalversiontrue}

```

**usexkv** Causes the xkeyval package to be input, this option extends the option list of \fillIn.

```

127 \DeclareOptionX{usexkv}{\let\eq@usexkeys\eq@YES}

```

2014/12/19 Now, the usexkeys is on by default.

```

128 \let\eq@usexkeys\eq@YES

```

**max Renditions** Options relating to renditions. The max and rendition option can be used instead of the \numVersions{2} and \forVersion{a}, respectively. These options allow you to set the version information through a package option. This allows us, for example, to use a cfg file such as rendition.cfg to dynamically set the version. This feature is used primarily by AeB Exam Builder.

```

129 \let\eq@renditionOptions\@empty
130 \let\eq@max@selected\@empty \let\eq@ren@selected\@empty
131 \DeclareOptionX{max}{\def\eq@max@selected{#1}%
132 \g@addto@macro\eq@renditionOptions{\numVersions{#1}}}
133 \DeclareOptionX{rendition}{\def\eq@ren@selected{#1}%
134 \g@addto@macro\eq@renditionOptions{\forVersion{#1}}}

```

**allowrandomize Randomization** Options relating to randomization. Use this option to randomize the choices of a multiple choice question.

```

135 \DeclareOptionX{allowrandomize}{\AtEndOfPackage{\inputRandomizeChoices}}
136 \def\inputRandomizeChoices{\InputIfFileExists{aebrandom.def}
137 {\typeout{inputting aebrandom.def}}{\cannot find aebrandom.def}}

```

**Set Misc. Defaults and Helper Macros.** We set some defaults, and define macros for use by the document author.

```

138 \let\eq@online\eq@NO
139 \def\ifisOnline{\ifx\eq@online\eq@YES}
140 \let\use@email\eq@NO
141 \let\load@web\eq@NO
142 \let\load@exerquiz\eq@NO
143 \def\sqLinks{\def\sqstar{}}\sqLinks
144 \def\sqForms{\def\sqstar{*}}

```



```

145 \def\NoSpaceToWork{\let\eq@insertverticalspace\eqe@NO}
146 \def\SpaceToWork{\let\eq@insertverticalspace\eqe@YES}
147 \let\eq@nototals\eqe@NO
148 \let\eq@nosummarytotals\eqe@NO
149 \let\eq@parttotals\eqe@YES
150 \let\eqx@separationrule\eqe@YES
151 \let\eq@insertverticalspace\eqe@YES
152 \let\eqex@coverpage\relax
153 \def\@reportpoints{0}
154 \let\marginpoints\@empty
155 \let\totalsbox=\hfil

```

## 2.3 Options from and for web

forpaper	
forcolorpaper	Here is the list of options of exerquiz we plan to recognize.
forcolorpaper*	<pre> 156 \@ifundefined{ifequusecolor}{\newif\ifequusecolor\equusecolorfalse}{ 157 \DeclareOptionX{forpaper}{\equusecolorfalse\eqforpapertrue 158 \PassOptionsToPackage{monochrome}{\eq@ColorPackage}} 159 \DeclareOptionX{forcolorpaper}{\equusecolortrue\eqforpapertrue} 160 \newif\ifForceNoColor\ForceNoColorfalse 161 \DeclareOptionX{forcolorpaper*}{\equusecolortrue\eqforpapertrue 162 \ForceNoColortrue} % for print </pre>
preview	Preview shows outlines for form fields.
webOpts	<pre> 163 \DeclareOptionX{preview}{\previewtrue} </pre> <p>Pass options to web, when pdf or higher is used.</p> <pre> 164 \define@key{eqexam.sty}{webOpts}[]{\def\eqe@webOpts{#1,\eqDriverName}} 165 \def\eqe@webOpts{\eqDriverName} </pre>

## 2.4 Options from and for exerquiz

exerquizOpts	Options from exerquiz that are useful for this package. Pass options to exerquiz, when links or higher is used.
	<pre> 166 \define@key{eqexam.sty}{exerquizOpts}[]{\def\eqe@eqOpts{#1,nodljs}} 167 \def\eqe@eqOpts{nodljs} </pre>
nosolutions	Solutions related options
nohiddensolutions	<pre> 168 \DeclareOptionX{nosolutions}{\ifx\thisOpt@OK\eqe@YES </pre>
solutionsafter	<pre> 169 \def\l@stPO{nosolutions}\let\thisOpt@OK\eqe@NO </pre>
solutionsonly	<pre> 170 \eqe@optiont@kenMsg\eq@nolinktrue\eq@nosolutionstrue 171 \expandafter\displayworkareatru 172 \else\w@rningBadOpts\fi} 173 \DeclareOptionX{nohiddensolutions}{\eq@globalshowsolutionstrue} 174 \DeclareOptionX{noHiddensolutions}{% 175 {\eq@globalshowsolutionstrue\AtBeginDocument{\def\Hidesymbol{h}}} 176 \DeclareOptionX{solutionsafter}{\ifx\thisOpt@OK\eqe@YES 177 \def\l@stPO{solutionsafter}\let\thisOpt@OK\eqe@NO 178 \eqe@optiont@kenMsg 179 \eq@solutionaftertrue\expandafter </pre>

```

180 \displayworkareafalse\else\warningBadOpts\fi}
181 \DeclareOptionX{solutionsonly}{\ifx\thisOpt@OK\eqe@YES
182 \def\l@stPO{solutionsafter}\let\thisOpt@OK\eqe@NO
183 \eqe@option@kenMsg\AtEndOfPackage{\therearesolutionstrue
184 \let\exerSolnsHeadnToc\relax}
185 \solutionsonlytrue\answerkeytrue\expandafter
proofing 186 \displayworkareafalse\else\warningBadOpts\fi}

```

The `proofing` option sets a switch that controls whether the checkbox for multiple choice questions appears, and whether the answer for the `\fillin` command appears. `\ifeq@proofing` is set to true when the `answerkey` option is taken.

```

187 \DeclareOptionX{proofing}{\eqe@proofingtrue}

```

We provide two helper commands for turning on or off proofing. These are `\showproofing` for turning on proofing and `\hideproofing` for turning off proofing. There was some reason for defining these two, but can't remember now.

```

188 \newcommand{\showproofing}{\eqe@proofingtrue}
189 \newcommand{\hideproofing}{\eqe@proofingfalse}
showgrayletters 189 \newcommand{\hideproofing}{\eqe@proofingfalse}

```

When this option is in effect, capital letters in gray appear under the multiple choice question boxes.

```

190 \newif\ifaebshowgrayletters\aebshowgraylettersfalse
191 \DeclareOptionX{showgrayletters}{%
192 {\AtEndOfPackage{\aebshowgrayletterstrue}}

```

Define `\ifdisplayworkarea` and user commands to turn off and on.

```

\ifdisplayworkareaOn 193 \@ifundefined{ifdisplayworkarea}{\newif\ifdisplayworkarea
\ifdisplayworkareaOff 194 \displayworkareafalse}{%
195 \providecommand\displayworkareaOn{\displayworkareatrue}
196 \providecommand\displayworkareaOff{\displayworkareafalse}

```

**Color packages** We set the color package, `xcolor` preferred.

```

197 \IfFileExists{xcolor.sty}%
198 {\def\eqe@ColorPackage{xcolor}\def\eqe@color@opt{table}%
199 \PassOptionsToPackage{xcolor}{table}}
200 {\def\eqe@ColorPackage{color}\let\eqe@color@opt\@empty}

```

`noxcolor` `noxcolor` forces the use of the color package.

```

201 \DeclareOptionX{noxcolor}{\def\eqe@ColorPackage{color}}

```

#### 2.4.1 Title page options

`useclassmaketitle` To support the use of `eqexam` outside a straight exam document, we allow the user to bypass the redefinition of `\maketitle`. This is automatic with the `fortextbook` option.

```

202 \newif\ifeqeuseclassmaketitle \eqeuseclassmaketitlefalse
203 \DeclareOptionX{useclassmaketitle}{\eqeuseclassmaketitletrue}
204 \newif\if@bypasseqexamheading \@bypasseqexamheadingfalse
205 \DeclareOptionX{bypasspkgpagestyle}{\@bypasseqexamheadingtrue}
206 \ExecuteOptionsX{nototals}}

```

## 2.5 Drivers Recognized

These drivers are only relevant when a PDF option is taken (`pdf`, `links`, `online`, `email`). For ordinary paper documents, it is not necessary to specify the driver. If you put the assignment/homework/test (solns) on the web, suggested option is `pdf`, this inputs `hyperref`, and the document info dialog is filled in.

```
dvipsone 207 \DeclareOptionX{dvipsone}{%
  dvips   \def\eqe@drivernum{0}\def\eqDriverName{dvipsone}%
  pdftex 208   \PassOptionsToPackage{\eq@ColorPackage}{dvipsone}}
dvipdfm 209   \PassOptionsToPackage{\eq@ColorPackage}{dvipsone}}
dvipdfmx 210 \DeclareOptionX{dvips}{\def\eqe@drivernum{0}\def\eqDriverName{dvips}%
  xetex   211   \PassOptionsToPackage{\eq@ColorPackage}{dvips}}
textures 212 \DeclareOptionX{pdftex}{\def\eqe@drivernum{1}\def\eqDriverName{pdftex}%
  213   \PassOptionsToPackage{\eq@ColorPackage}{pdftex}}
  214 \DeclareOptionX{dvipdfm}{%
  215   \def\eqe@drivernum{2}\def\eqDriverName{dvipdfm}%
  216   \PassOptionsToPackage{\eq@ColorPackage}{dvipdfm}}
  217 \DeclareOptionX{dvipdfmx}{\def\eqe@drivernum{2}%
  218   \def\eqDriverName{dvipdfmx}%
  219   \PassOptionsToPackage{\eq@ColorPackage}{dvipdfmx}}
  220 \DeclareOptionX{xetex}{\def\eqe@drivernum{2}\def\eqDriverName{xetex}%
  221   \PassOptionsToPackage{\eq@ColorPackage}{xetex}}
  222 \DeclareOptionX{textures}{%
  223   \def\eqe@drivernum{3}\def\eqDriverName{textures}%
  224   \PassOptionsToPackage{\eq@ColorPackage}{textures}}
  225 \DeclareOptionX*{%
  226   \PassOptionsToPackage{\CurrentOption}{\eq@ColorPackage}}
  227 \def\eqe@drivernum{5}
  228 \let\eqDriverName\empty
```

If `exerquiz` is not loaded, when we need to define some of the switches that were defined in `exerquiz`.

The following switches are used in the options above, and are also defined in `web`, `exerquiz`, or `eforms`.

```
229 \newif\ifeq@solutionsafter \eq@solutionsafterfalse
230 \def\ifsolutionsafter{\csname ifeq@solutionsafter\endcsname}
231 \let\solutionsaftertrue\eq@solutionsaftertrue
232 \let\solutionsafterfalse\eq@solutionsafterfalse
233 \newif\ifsolutionsonly\solutionsonlyfalse
234 \newif\ifeq@hidesolution \eq@hidesolutionfalse
235 \newif\ifeq@globalshowsolutions \eq@globalshowsolutionsfalse
236 \newif\ifeq@nosolutions \eq@nosolutionsfalse
237 \newif\ifeq@proofing \eq@proofingfalse
238 \newif\ifeq@nolink \eq@nolinkfalse
239 \@ifundefined{ifpreview}{\newif\ifpreview \previewfalse}{}
240 \newif\ifeqforpaper \eqforpaperfalse
```

We define the commands for inputting the CFG files.

```
241 \def\eqe@csarg#1#2{\expandafter#1\csname#2\endcsname}
242 \@for\eqe@tmp@i:={,i,ii,iii,iv,v,vi\do{\eqe@csarg
243   \edef\eqemyconfig\eqe@tmp@i}{\noexpand
```

```

244 \InputIfFileExists{eqexam\eqe@tmp@i.cfg}{-}{-}%
245 \eqe@tmp@exp
246 }

```

## 2.6 Bring in Config Files

First read `web.cfg`, to possibly get the driver, then input `eqecus.opt`, which is used to create convenient custom options.

Here is an example of usage for defining your own custom options, must be based on current options, this code would be in the file `eqecus.opt`.

```

\DeclareOptionX{atbdbopts}
{%
  \ExecuteOptionsX{online}
  \ExecuteOptionsX{forcolorpaper}
  \ExecuteOptionsX{nosolutions}
  \ExecuteOptionsX{nopoints}
  \ExecuteOptionsX{nototals}
  \ExecuteOptionsX{nospacetowork}
  \ExecuteOptionsX{obeylocalversions}
  \ExecuteOptionsX{myconfig}
}

```

The following config files are input prior to `\ProcessOptionsX`, and can, therefore, contain declaration of options. `web.cfg` usually only specifies the default driver. `eqecus.opt` is used by `@EASE`, but can be used locally.

```

247 \let\bWebCustomize\endinput
248 \let\eWebCustomize\relax
249 \InputIfFileExists{web.cfg}{-}{-}
250 \InputIfFileExists{eqecus.opt}{-}{-}

```

These two are used by the rendition package and the exam builder utility.

```

251 \InputIfFileExists{rendition.cfg}{-}{-}
252 \InputIfFileExists{exambuilder.cfg}{-}{-}

```

## 2.7 Process Options

Now process the options.

```

253 \ProcessOptionsX

```

When the online or email option is taken, as well as answerkey, we cancel the `flexextended` option.

```

254 \ifeqonline
255   \ifanswerkey
256   \def\flexextendedInput{\let\turnfl@nskeyMsg\@empty
257   \PackageWarningNoLine{eqexam}
258   {You've chosen the online or email option\MessageBreak
259   with the flexextended option. This is not\MessageBreak
260   supported, removing the flexextended option}}\fi
261 \fi

```

The `\selectedMC` command contains the choice for the styling for the region multiple choice questions: rectangles or circles. The default is rectangles.

262 `\AtEndOfPackage{\selectedMC}`

(2015/07/12) Move the inclusion of the color package prior to loading many of the other required packages.

263 `\edef\eq@tmpexp{\noexpand`

264 `\RequirePackage{\eq@ColorPackage}}\eq@tmpexp`

We require a minimal version for xcolor.

265 `\@ifpackageloaded{xcolor}{\AtEndOfPackage{\let\CT@cell@color\relax`

266 `\let\CT@arc@\relax}%`

267 `\@ifpackagelater{xcolor}{2004/07/04}{\PackageError{eqexam}{%`

268 `*****\MessageBreak`

269 `* Your Version of 'xcolor.sty' is too old!\MessageBreak`

270 `* You need the version from 2004/07/04 or newer\MessageBreak`

271 `* or use: \string\usepackage[noxcolor]{eqexam}\MessageBreak`

272 `* or \string\documentclass[noxcolor]{article}\MessageBreak`

273 `*****}\}%`

274 `}%`

275 `{}}`

If `nocustomdesign` option is taken, we set the switch `\eqcustomdesignfalse`.

276 `\if\eq@nocustomdesign1\eqcustomdesignfalse\fi`

Define a `\immediate\write` helper macro.

277 `\long\def\eq@IW0#1{\immediate\write#1}`

### Early definitions for the `fortextbook` option.

`\showAllAnsAtEnd` If the user has chosen the `vspacewithsolns` option, we must turn off all other solution options, namely `answerkey`. This command is used internally.

(2016/10/02) Include `\eqTopOfSolnPage` from `exerquiz` to support copying question to solution page.

278 `\newcommand{\showAllAnsAtEnd}{%`

279 `\makeAnsEnvForSolnsAtEnd\eqTopOfSolnPage`

280 `\answerkeytrue\eq@proofingtrue`

281 `\eq@solutionsaftertrue\vspacewithsolnstrue`

282 `\displayworkareafalse`

283 `}`

`\makeAnsEnvForSolnsAtEnd` One user wanted to be able to use the `answers` environment in the solutions section at the end of the document (when the `vspacewithsolns` is used). Here it is. This definition is added to the definition of `\showAllAnsAtEnd`.

284 `\newcommand{\makeAnsEnvForSolnsAtEnd}{%`

285 `% \proofingsymbol{\ding{52}}%`

286 `\let\answers\answers@sq`

287 `\let\endanswers\endanswers@sq`

288 `\let\mananswers\mananswers@sq`

```

289 \let\endanswers\endanswers@sq
290 }

291 \newcommand{\solAtEndFormatting}[1]{\def\eqeAEFormatting{#1}}
292 \let\eqeAEFormatting@empty

\writeAllAnsAtEnd writes the \showAllAnsAtEnd command to the solutions file.
293 \def\writeAllAnsAtEnd{\ifsolutionsonly\else
294     \let\quiz@solns\ex@solns
295     \eqe@IWO\quiz@solns{\string\showAllAnsAtEnd}%
296     \ifx\eqeAEFormatting@empty\else
297         \eqe@IWO\quiz@solns{\string\eqeAEFormatting}\fi
298     \fi
299 }
300 \def\writeWithSolDocTrue{\writeT@SolnFile{\protect\withinsoldoctrue}}

```

**\setSolnMargins** Sets the value of `\eqemargin` in the context of the solution file, this command is redefined later.

```

301 \newcommand{\setSolnMargins}[1]{\setlength\eqemargin{#1}\ignorespaces}

```

(2011/05/08) In the new version of `eqexam`, the one that makes the problems within an `exam` environment, into a list, the solutions file that appears at the end of the document also needs to be put into a list. Here, we define the command that writes the beginning of the `eqequestions` environment to the beginning of the `\jobname.sol` file. We hard-wire write `\setSolnMargins{\the\eqemargin}` to the solution file, but leave a back door open to write an alternate string.

**\altSetSolnMargins** (2014/03/20) `\altSetSolnMargins` allows you, through its argument to pass an alternate string.

```

302 \def\altSetSolnMargins#1{\def\alt@SetSolnMargins{#1}}
303 \let\alt@SetSolnMargins@empty
304 \def\writeBeginEqeQuestions{\ifsolutionsonly\else
305     \ifOKToWriteExamData
306         \let\quiz@solns\ex@solns
307         \ifx\alt@SetSolnMargins@empty
308             \writeT@SolnFile{\string\setSolnMargins{\the\eqemargin}}\else
309             \writeT@SolnFile{\alt@SetSolnMargins}\fi
310         \writeT@SolnFile{\protect\eqgriii\string\noindent
311             \string\begin{eqequestions}}}%
312     \fi
313 \fi
314 }

```

(2011/05/08) We define the command that writes the end of the `eqequestions` environment to the beginning of the `\jobname.sol` file.

```

315 \newcommand{\setBtwnExamSkip}[1]{\gdef\btwnExamSkipAmt{#1}}
316 \def\btwnExamSkip{\ifdim#1=Opt\else
317     \vskip#1\relax\fi}}
318 \setBtwnExamSkip{6pt}
319 \def\writeEndEqeQuestions{%
320     \ifsolutionsonly\else

```

```

321 \ifOKToWriteExamData
322 \let\quiz@solns\ex@solns
323 \eqe@IWO\quiz@solns{\string\eqgrii
324 \string\end{eqequestions}^^J}%
325 \writeT@SolnFile{\string\btwnExamSkip^^J}%
326 \fi\fi
327 }

If \ifvspacewithsolns we set the switches need to simulate nosolutions.
328 \def\csarg#1#2{\expandafter#1\csname#2\endcsname}
329 \def\saveIFEQE#1{\def\ARG{#1ifSave}%
330 \expandafter\csarg\expandafter
331 \let\expandafter\ARG\csname#1\if#1true\else false\fi\endcsname}
332 \saveIFEQE{vspacewithsolns}\saveIFEQE{answerkey}
333 \saveIFEQE{eq@proofing}\saveIFEQE{eq@solutionsafter}
334 \saveIFEQE{eq@nolink}\saveIFEQE{eq@nosolutions}
335 \saveIFEQE{displayworkarea}
336 \def\vpwsSimulateNoSolns{% dps28
337 \ifvspacewithsolns
338 \answerkeyfalse\eq@proofingfalse\eq@solutionsafterfalse
339 \eq@nolinkfalse\eq@nosolutionsfalse\displayworkareatrue
340 \fi}
341 \def\reverseVSWs{\vspacewithsolnsfalse
342 \answerkeyifSave\eq@proofingifSave
343 \eq@solutionsafterifSave\eq@nolinkifSave
344 \eq@nosolutionsifSave\displayworkareaifSave
345 }
346 \vpwsSimulateNoSolns

```

## 2.8 Save Switch Values

Now, save the current state of the switches defined above. When, and if, the packages web, exerquiz and eforms are loaded, they will overwrite the choices set by the author, so we save them.

```

347 \let\savedeq@online\eq@online
348 \let\savedifeq@solutionsafter\ifeq@solutionsafter
349 \let\savedifeq@hidesolution\ifeq@hidesolution
350 \let\savedifeq@globalshowsolutions\ifeq@globalshowsolutions
351 \let\savedifeq@nosolutions\ifeq@nosolutions
352 \let\savedifeq@proofing\ifeq@proofing
353 \let\savedifeq@nolink\ifeq@nolink
354 \let\savedifpreview\ifpreview
355 \let\savedifeqforpaper\ifeqforpaper
356 \let\ifnosolutions\ifeq@nosolutions

```

## 3 Required Packages

The following are the required packages for eqexam.

```

357 \RequirePackage{amstext,amssymb}

358 \@ifundefined{if@fleqn}{\let\fleqnOn\relax\let\fleqnOff\relax}
359   {\def\fleqnOn{\@fleqntrue}\def\fleqnOff{\@fleqnfalse}}

```

Bring the comment package in early, before verbatim, these two clash a bit.

```

360 \RequirePackage{aeb-comment}

```

If `\BeforeIncludedComment` is defined, the old version of `aeb-comment` is used; otherwise, the new version of `aeb-comment` is being used. The new version supports utf-8.

```

361 \@ifundefined{BeforeIncludedComment}{\let\aebc@end\endgroup}
362   {\let\aebc@end\relax}
363 \def\eqe@commentChkMsg{\@ifpackageloaded{comment}
364   {\PackageWarningNoLine{eqexam}
365     {The comment package is incompatible with the\MessageBreak
366       aeb-comment package, do not use the comment package}}{}}
367 \AtBeginDocument{\eqe@commentChkMsg}

```

The macro `\includeexersolutions` is defined in `eqexam.def`. We execute the command `\include@solutions` before the web package is loaded. The web package has a `\AtEndDocument` as well, and inserts a new page that we don't want.

```

368 \AtEndDocument{\includeexersolutions}

```

If user has specified one of the pdf options (pdf, links, online, email), we bring in the web package.

```

369 \@ifpackageloaded{web}{\let\load@web\eqe@YES}{%
370   \ifx\load@web\eqe@YES\ifnum\eqe@drivernum=5
371     \PackageInfo{eqexam}{You have not selected a driver %
372       for eqexam. Perhaps the \MessageBreak
373       driver is introduced through web.cfg}\fi
374     \expandafter\RequirePackage\expandafter[\eqe@webOpts]{web}%
375     \edef\@pdfcreator{\@pdfcreator, The eqexam Package}\fi
376 }

```

If user has specified links, online or email, we bring in the `exerquiz` package.

```

377 \@ifpackageloaded{exerquiz}{\let\load@exerquiz\eqe@YES}{%
378   \let\symbolchoice\@gobble
379   \ifx\load@exerquiz\eqe@YES\expandafter
380     \RequirePackage\expandafter[\eqe@eqOpts]{exerquiz}[2011/08/30]

```

We input `exerquiz` with the `nodljs`, we don't need all the JavaScript to process interactive shortquizzes or quizzes, but we do want the option of adding in document JavaScript, so after we input `exerquiz`, we set the switches to allow these features.

```

381   \let\importdljs\eqe@YES\let\execjs\eqe@YES
382   \fi
383 }
384 \@ifpackageloaded{eforms}{\let\symbolchoice\@gobble}{%

```

If `eforms` is not loaded, we define `\symbolchoice`, which is defined in `eforms` to gobble up its argument, we don't need it.



```

385 \let\symbolchoice\@gobble
386 }

```

Here is a fix to a problem I've been having previewing in dviwindo. I've traced the problem down to \pdfviewparams. Redefining \pdfviewparams as follows.

```

387 \def\eqDvipsone{dvipsone}
388 \ifpackageloaded{hyperref}
389 {\ifx\eqDriverName\eqDvipsone
390 \renewcommand\pdfviewparams{ null null null}\fi
391 }\let\textorpdfstring\@firstoftwo}

```

Now that we have possibly input web or exerquiz, we need to restore the authors options.

```

392 \let\eq@online\savedeq@online
393 \let\ifeq@solutionsafter\savedifeq@solutionsafter
394 \def\ifsolutionsafter{\ifeq@solutionsafter} % user interface
395 \let\ifeq@hidesolution\savedifeq@hidesolution
396 \let\ifeq@globalshowssolutions\savedifeq@globalshowssolutions
397 \let\ifeq@nosolutions\savedifeq@nosolutions
398 \let\ifeq@proofing\savedifeq@proofing
399 \let\ifeq@nolink\savedifeq@nolink
400 \let\ifpreview\savedifpreview
401 \let\ifeqforpaper\savedifeqforpaper

```

Other packages of interest.

```

402 \RequirePackage{calc}
403 \RequirePackage{pifont}
404 \RequirePackage{array}

```

Here, I input the verbatim package after the comment package.

```

405 \RequirePackage{verbatim}
406 \ifundefined{dlcomment}{\typeout{!! defining dlcomment}}%
407 \let\dlcomment\comment
408 \let\enddlcomment\endcomment
409 \let\eqSavedComment\dlcomment
410 \let\endeqSavedComment\enddlcomment
411 }{}

```

When constructing paper tests, I often use a multi-column format for some of the questions, so let's require this package

```

412 \RequirePackage{multicol}
413 \setlength\columnseprule{.4pt}
414 \raggedcolumns\multicolsep=3pt
415 \newcommand{\setmulticolprob}{%
416 \setlength{\linewidth}{\linewidth+\eqemargin}}

```

For the fortetbook option, we require eso-pic.

```

417 \edef\eqe@reqPack{\ifeqfortetbook\noexpand\RequirePackage{eso-pic}\else
418 \relax\fi}
419 \eqe@reqPack

```

We've processed the options, I need to detect whether the document author has not chosen any solutions options.

```
420 \ifvspacewithsolns\solutionsAtEndtrue\fi
421 \ifanswerkey\solutionsAtEndfalse\fi
422 \ifeq@nosolutions\solutionsAtEndfalse\fi
423 \ifeq@solutionsafter\solutionsAtEndfalse\fi
```

## 4 Page Layout

(2011/05/08) The revised version of `eqexam` allows the document author to more easily design the size of the page; the new version makes all content inside the `exam` environment into a list, this gives us better control over the margins and spacing.

`\eqexammargin` (2011/05/08) Use this command to set the margin for the `exam` environment.

```
424 \@ifundefined{eqemargin}{\newlength{eqemargin}}{}
425 \providecommand{\prbDecPt}[1]{\def\eqe@decPointPrb{#1}}
426 \def\eqe@decPointPrb{.}\def\eqe@dpsepPrb{\ }
427 \providecommand{\prbPrtsep}[1]{\def\eqe@prtsepPrb{#1}}
428 \def\eqe@prtsepPrb{\ }
429 \providecommand{\prbNumPrtsep}[1]{\def\eqe@hspannerPrb{#1}}
430 \def\eqe@hspannerPrb{\ }
431 \newcommand{\eqexammargin}[2][\normalsize\normalfont\bfseries]{%
432   \settowidth{eqemargin}{#1#2\eqe@decPointPrb\eqe@hspannerPrb}}
(2011/05/08) The default margin for the eqexam environment, two digits and a
space.
433 \eqexammargin{00}

434 \def\eqe@hspannerSoln{\ } % space after prob number
435 \providecommand{\solNumPrtsep}[1]{\def\eqe@hspannerSoln{#1}}
```

`\eqeSetExamPageParams` (2011/05/08) The default spacing maximizes the amount of space on the page.

```
436 \newcommand{\eqeSetExamPageParams}{%
437   \setlength{\headheight}{12pt}
438   \setlength{\topmargin}{-.5in}
439   \setlength{\headsep}{20pt}
440   \setlength{\oddsidemargin}{0pt}
441   \setlength{\evensidemargin}{0pt}
442   \setlength{\marginparsep}{11pt}
443   \setlength{\marginparwidth}{35pt}
444   \setlength{\footskip}{11pt}
445 }
```

`\eqExamPageLayout` Set the basic parameters of this exam page package

```
446 \newcommand{\eqExamPageLayout}{%
447   \setlength\textwidth\paperwidth
448   \addtolength{\textwidth}{-2in}
449   \addtolength{\textwidth}{-\oddsidemargin}
```

```

450 \setlength\textheight{\paperheight}
451 \addtolength\textheight{-2in}
452 \addtolength\textheight{-\headheight}
453 \addtolength\textheight{-\headsep}
454 \addtolength\textheight{-\topmargin}
455 \addtolength\textheight{-\footskip}
456 }

```

(2011/05/08) If usecustomdesign is used it is expected that \eqeSetExamPageParams and \eqeSetExamPageParams are redefined in the preamble, otherwise, we set up the standard parameters; otherwise

```

457 \ifeqcustomdesign\else
458 \eqeSetExamPageParams
459 \eqExamPageLayout
460 \fi

```

A simple page layout scheme for this exam.

```

461 \newcommand{\ps@eqExamheadings}
462 {%
463 \renewcommand{\@oddhead}{%
464 {\normalfont\normalsize\ifnum\value{page}<2
465 \hfil\else\eqExamRunHead\fi}}%
466 \renewcommand{\@evenhead}{\@oddhead}
467 \renewcommand{\@oddfoot}{\settotsbox\runExamFooter}
468 \renewcommand{\@evenfoot}{\@oddfoot}
469 }
470 \raggedbottom

```

## 5 Counters, Lengths and Tokens

<p>eqpointsofar</p> <p>eqpointsthispage</p> <p>eq@numparts</p>	<p>Some counters to keep track of things. The first two counters keep track, respectively, of the total points so far up the current page, and the number of points on the current page. The counter eq@numparts holds the number of parts of the multi-part question.</p>
--	--

```

471 \newcount\eqe@tempcnta
472 \newcounter{eqpointsofar}
473 \newcounter{eqpointsthispage}
474 \newcounter{eq@numparts}
475 \newcounter{eq@count}
476 \newtoks\partNames \partNames={}
477 \newlength{\eq@tmplengthA}
478 \newlength{\eq@tmplengthB}
479 \newbox{\eq@pointbox}
480 \newlength{\eq@pointboxtotalheight}

```

Some scratch registers to do calc calculations.

```

481 \newlength{\eqetmplengtha}
482 \newlength{\eqetmplengthb}

```

## 6 Some Macros to Support the Options

We make a few definitions to support various options.

```

483 \def\PointsOnLeft{\def\@reportpoints{1}\let\marginpoints\eqleftmargin}
484 \def\PointsOnRight{\def\@reportpoints{2}\relax
485   \let\marginpoints\eqrightmarginbox}
486 \def\PointsOnBothSides{\def\@reportpoints{3}\relax
487   \let\marginpoints\eqbothmargins}
488 \newif\ifeq@nopoints \eq@nopointsfalse
489 \def\NoPoints{\if\isInExamEnv\eq@NO
490   \eq@nopointstrue\def\@reporttotals{0}\let\totalsbox=\hfil
491   \let\marginpoints\@empty\let\eq@nosummarytotals\eq@YES}else
492   \PackageWarning{eqexam}{The \string\NoPoints\space ignored;
493     it needs to be executed\MessageBreak outside of an
494     exam environment}\fi}
495 \def\TotalsOnLeft{\def\@reporttotals{1}\def\totalsbox{\totalsboxleft}}
496 \def\TotalsOnRight{\def\@reporttotals{2}\def\totalsbox{\totalsboxright}}
497 \def\noZeroTotals{\let\eq@zeroTotalsAllowed\eq@NO}
498 \def\allowZeroTotals{\let\eq@zeroTotalsAllowed\eq@YES}
499 \allowZeroTotals
500 \def\NoTotals{\def\@reporttotals{0}\let\totalsbox=\hfil}
501 \def\SummaryTotalsOn{\let\eq@nosummarytotals\eq@NO}
502 \def\SummaryTotalsOff{\let\eq@nosummarytotals\eq@YES}
503 \def\eoTotalOff{\let\eq@parttotals\eq@NO}
504 \def\eoTotalOn{\let\eq@parttotals\eq@YES}
505 \def\separationruleOn{\let\eqx@separationrule\eq@YES}
506 \def\separationruleOff{\let\eqx@separationrule\eq@NO}
507 \def\AllowFitItIn{\global\let\eq@fititin\eqfititin}
508 \def\DoNotFitItIn{\global\let\eq@fititin\@gobble}

509 \def\NoSolutions{\eq@nolinktrue\eq@nosolutionstrue
510   \displayworkareatruetrue}
511 \@onlypreamble\NoSolutions

```

Added `\chngToNoSolns`, useful for book authors that want to switch between `\AnswerKey`, `\chngToNoSolns`, and `\SolutionsAtEnd`.

```

512 \def\chngToNoSolns{\solutionsAtEndfalse\answerkeyfalse
513   \eq@proofingfalse\eq@solutionsafterfalse
514   \eq@nosolutionstrue\displayworkareatruetrue}

```

`\vspacewithkeyOn` User interface to keeping the declare `vspace`, even when the `answerkey` (or `solutionsafter`) option is taken. The switch `\ifkeepdeclaredvspacing` is defined in `eqexam.def/exerquiz`.

```

515 \def\vspacewithkeyOn{\keepdeclaredvspacingtrue}
516 \def\vspacewithkeyOff{\keepdeclaredvspacingfalse}

```

`\displayPointsOn` `\displayPointsOn` displays the points, if not otherwise overridden and the command `\displayPointsOff` turns off the display of points. In either case, points are calculated.

```

517 \newif\ifdispl@yPoints\displ@yPointstrue
518 \newcommand{\displayPointsOn}{\displ@yPointstrue}
519 \newcommand{\displayPointsOff}{\displ@yPointsfalse}
520 \newif\ifl@stDispl@yPoints\l@stDispl@yPointsfalse

```

\encloseProblemsWith \encloseProblemsWith to support the solutionsonly option

```

521 \def\encloseProblemsWith#1{%
522   \ifsolutionsonly\excludcomment{#1}\else
523   \includecomment{#1}\fi
524 }

```

## 7 Colors

\proofingsymbolColor Here we list commands for controlling colors. There are some other colors defined in the stand alone code.

```

\instructionsColor
\eqCommentsColor 525 \providecommand{\proofingsymbolColor}[1]{\def\@proofingsymbolColor{#1}}
\universityColor 526 \proofingsymbolColor{red}
\titleColor 527 \providecommand{\instructionsColor}[1]{\def\@instructionsColor{#1}}
\authorColor 528 \instructionsColor{blue}
\subjectColor 529 \providecommand{\eqCommentsColor}[1]{\def\@eqCommentsColor{#1}}
\linkcolor 530 \eqCommentsColor{blue}
\nolinkcolor 531 \providecommand{\eqCommentsColorBody}[1]{\def\@eqCommentsColorBody{#1}}
\fillinColor 532 \eqCommentsColorBody{black}
\forceNoColor 533 \providecommand{\universityColor}[1]{\def\webuniversity@color{#1}}
\eqEndExamTotalColor 534 \universityColor{blue}
535 \providecommand{\titleColor}[1]{\def\webtitle@color{#1}}
536 \titleColor{black}
537 \providecommand{\authorColor}[1]{\def\webauthor@color{#1}}
538 \authorColor{black}
539 \providecommand{\subjectColor}[1]{\def\websubject@color{#1}}
540 \subjectColor{blue}
541 \providecommand{\linkcolor}[1]{\def\@linkcolor{#1}}
542 \linkcolor{blue}
543 \providecommand{\nolinkcolor}[1]{\def\@nolinkcolor{#1}}
544 \nolinkcolor{black}
545 \providecommand{\eqEndExamTotalColor}[1]{\def\endexamtotal@color{#1}}
546 \eqEndExamTotalColor{black}
547 \newcommand\fillinColor[1]{\def\eq@fillinColor{#1}}\fillinColor{red}
548 \providecommand{\sectionColor}[1]{\def\web@sectionsColorOld{#1}}
549 \def\@eb@sectioncolor{#1}}
550 \sectionColor{blue}
551 \def\forceNoColorSet{\proofingsymbolColor{black}%
552 \instructionsColor{black}%
553 \eqCommentsColor{black}\universityColor{black}%
554 \titleColor{black}\authorColor{black}%
555 \subjectColor{black}\linkcolor{black}%
556 \nolinkcolor{black}\fillinColor{black}%
557 \instructionsColor{black}\eqCommentsColor{black}%

```

```

558 \eqCommentsColorBody{black}%
559 \eqEndExamTotalColor{black}\checkboxColor{}%
560 \ckcirColor{}\if\load@web\eqe@YES
561 \sectionColor{black}\fi
562 }
563 \ifForceNoColor
564 \AtBeginDocument{\forceNoColorSet}
565 \fi
566 \newcommand{\forceNoColor}{\ifForceNoColor\else\ForceNoColortrue
567 \AtBeginDocument{\forceNoColorSet}\fi}
\checkboxColor The \checkboxColor provides color for the MC and MS boxes, then a PDF-related
\ckcirColor option is not in effect, while \ckcirColor provides color for circular checkboxes.
Usually, these are black or both the same color.
568 \providecommand{\checkboxColor}[1]{\def\@rgi{#1}\ifx\@rgi\@empty
569 \let\checkbox@Color\relax\else
570 \def\checkbox@Color{\color{#1}}\fi}\checkboxColor{}
571 \providecommand{\ckcirColor}[1]{\def\eq@rgi{#1}\ifx\eq@rgi\@empty
572 \let\ckcir@Color\relax\else
573 \def\ckcir@Color{\color{#1}}\fi}\ckcirColor{}

```

## 8 Version Control

Here are some simple macros use to create two versions, version A and version B, of the same test.

`\examNum` Convenience macro for holding the exam number. It sets the value of `\nExam`.

```

574 \def\examNum#1{\def\nExam{#1}}
575 \examNum{1}

```

`\Exam` Convenience macros for titling the exam. Usage:

```

\Exam
\sExam
\VersionAtext{Test~\nExam--Version A}
\VersionBtext{Test~\nExam--Version B}
\shortVersionAtext{T\nExam A}
\shortVersionBtext{T\nExam B}

```

```

\examNum{1}
\forVersion{c}
\subject[C3]{Calculus III}
\title[\sExam]{\Exam}
\author{Dr.\ D. P. Story}

```

These next two definitions are overwritten by the two commands `\longTitleText` and `\shortTitleText`.

```

576 \def\Exam{\ifAB{\eq@VersionAtext}{\eq@VersionBtext}}
577 \def\sExam{\ifAB{\eq@shortVersionAtext}{\eq@shortVersionBtext}}

```

`\VersionAtext` Convenience macros for entering the text for the title, long and short for versions  
`\VersionBtext` A and B.  
`\shortVersionAtext`  
`\shortVersionBtext`

```

578 \def\VersionAtext#1{\def\eq@VersionAtext{#1}}
579 \def\VersionBtext#1{\def\eq@VersionBtext{#1}}
580 \def\shortVersionAtext#1{\def\eq@shortVersionAtext{#1}}
581 \def\shortVersionBtext#1{\def\eq@shortVersionBtext{#1}}
582 \VersionAtext{Exam~\nExam--Version A}
583 \VersionBtext{Exam~\nExam--Version B}
584 \shortVersionAtext{Exam~\nExam A}
585 \shortVersionBtext{Exam~\nExam B}

```

In this section we introduce a new set of commands that supersedes the commands defined above. Those commands were limited to only two versions. The ones below can handle up to 26 versions.

```

586 \newtoks\eqtemptokena
587 \newtoks\eqtemptokenb

```

`\numVersions` In the preamble, declare the number of versions for this document using `\numVersions`, e.g., `\numVersions{3}`. This sets the value of `\eq@nVersions`

```

588 \def\numVersions#1{\ifnum#1>26\def\eq@nVersions{26}%
589   \PackageWarning{eqexam}{The value of \string\numVersions\space is
590   too large. \MessageBreak Choose a natural number less than 27}
591   \else\def\eq@nVersions{#1}\fi}

```

`\longTitleText` Next we state the long and short titles for our document, one for each of our declare  
`\endlongTitleText` number of versions given earlier. For example, we can use the value `\nExam` in out  
`\shortTitleText` titles. Usage:  
`\endshortTitleText`

```

\longTitleText
  {Test~\nExam--Version A}
  {Test~\nExam--Version B}
  {Test~\nExam--Make Up}
\endlongTitleText
\shortTitleText
  {T\nExam A}
  {T\nExam B}
  {T\nExam MU}
\endshortTitleText

```

I've added markers that delimit the end of the arguments. In this way, the end of the list of titles can be detected, even though the number of titles is not the same as what is declared by the `\numVersions`.

If there are more titles than what is declared, the rest are absorbed (gobbled). If there are fewer titles than declared, a `LaTeX` package error is generated, and substitute titles are generated. Modified `\longTitleText` and `\shortTitleText` to have an optional argument (A–Z;a–z). You can select a particular title from a list of titles. If no optional argument is passed, then the title determined by `\forVersion` is used.

```

592 \newcommand{\longTitleText}[1][ ]{%
593   \ifeglobalversion\let\eq@selectedVersion@save\eq@selectedVersion
594   \else\let\eq@selectedVersion@save\relax\fi

```

```

595 \uppercase{\def\eqe@localTextTitle{#1}}%
596 \ifx\eqe@localTextTitle\@empty\else
597 \expandafter\forVersion\expandafter{\eqe@localTextTitle}\fi
598 \eqe@contTitleText{\Exam}{\endlongTitleText}%
599 }
600 \def\endlongTitleText{1}
601 \newcommand{\shortTitleText}[1][]{\%
602 \ifeqglobalversion\let\eq@selectedVersion@save\eq@selectedVersion
603 \else\let\eq@selectedVersion@save\relax\fi
604 \uppercase{\def\eqe@localTextTitle{#1}}%
605 \ifx\eqe@localTextTitle\@empty\else
606 \expandafter\forVersion\expandafter{\eqe@localTextTitle}\fi
607 \eqe@contTitleText{\sExam}{\endshortTitleText}%
608 }
609 \def\endshortTitleText{s}

```

Both title commands, above, call this macro which sets the environment for `\@gatherTitleText`, which gathers the list of titles.

```

610 \def\eqe@contTitleText#1#2{%
611 \setcounter{eq@count}{0}%
612 \eqtemptokena={}\let\endtitleMarker#2
613 \@gatherTitleText{#1}%
614 }

```

This command gathers each title and places it as the argument of a `\v<LETTER>` command. These are accumulated in token registers then saved in `\Exam` and `\sExam`.

```

615 \def\@gatherTitleText#1#2{%
616 \def\eqe@argii{#2}
617 \if\endtitleMarker\eqe@argii

```

Encountered the end marker. See if we have collected the correct number of titles declared. If we have collected too few, we note an warning in the log, and create titles.

```

618 \ifnum\value{eq@count}>\eq@nVersions\let\eqe@next\relax
619 \else\def\eqe@next{\eq@shortTitlesFix{#1}}\fi
620 \else
621 \stepcounter{eq@count}
622 \eqtemptokenb=\expandafter{#2}
623 \xdef#1{\the\eqtemptokena\expandafter\noexpand
624 \csname v\Alph{eq@count}\endcsname{\the\eqtemptokenb}}
625 \xdef\sExam{\the\eqtemptokena\expandafter\noexpand
626 \csname v\Alph{eq@count}\endcsname{\the\eqtemptokenb}}
627 \eqtemptokena=\expandafter{#1}
628 \ifnum\value{eq@count}<\eq@nVersions
629 \def\eqe@next{\@gatherTitleText{#1}}%
630 \else
631 \def\eqe@next{%
632 \if\endtitleMarker\endlongTitleText
633 \expandafter\eqe@absorbTokensLong

```



```

634             \else
635             \expandafter\eqe@absorbTokensShort
636             \fi
637         }%
638     \fi
639 \fi
640 \eqe@next
641 }
642 \long\def\eqe@absorbTokensLong#1\endlongTitleText{%
643     \protected@xdef\Exam{\Exam}\ifx\eq@selectedVersion@save\relax
644     \eqe@offVersion\else\expandafter\forVersion\expandafter
645     {\eq@selectedVersion@save}\fi}
646 \long\def\eqe@absorbTokensShort#1\endshortTitleText{%
647     \protected@xdef\sExam{\sExam}\ifx\eq@selectedVersion@save\relax
648     \eqe@offVersion\else\expandafter\forVersion\expandafter
649     {\eq@selectedVersion@save}\fi}

```

We have reached `\endtitleMarker`, but the count is still less than `\eq@nVersions`, so we'll warn the user, and create titles for user.

```

650 \def\eq@shortTitlesFix#1{%
651     \PackageWarning{eqexam}{You have defined an insufficient number
652     of titles\MessageBreak for the number of versions declared in
653     \string\numVersions.\MessageBreak Please fix the problem}%
654     \stepcounter{eq@count}%
655     \if\endtitleMarker\endlongTitleText
656         \edef\eqe@tmp{\noexpand\gatherTitleText{\noexpand#1}
657             {??---Title \# \the\value{eq@count}---??}%
658             \noexpand\endlongTitleText}
659     \else
660         \edef\eqe@tmp{\noexpand\gatherTitleText{\noexpand#1}
661             {T\# \the\value{eq@count}??}\noexpand\endshortTitleText}
662     \fi
663     \addtocounter{eq@count}{-1}%
664     \eqe@tmp
665 }

```

Here, we define `\ifAB` so that document under the old system still work properly, I hope. Usage of `\ifAB` at this point is discouraged.

```

666 \def\ifAB#1#2{\if\eq@selectedVersion A#1%
667     \else\if\eq@selectedVersion B#2\fi\fi}
668 \def\eq@replaceToken#1{#1}

```

`\forVersion` Here is the command that does all the work. It creates alternate text macros for each of the versions declared using `\numVersions`.

For example, assuming `\numVersions{3}` appeared earlier, the command `\forVersion{a}` (or `\forVersion{A}`) defines 3 text commands `\vA`, `\vB` and `\vC`, each taking one argument, the text you want to display:

Name the `\vA{place}\vB{date}\vC{year}` of the signing of the Declaration of independence.

Since we said `forVersion{a}` only the `\vA` text is displayed, the others are gobbled up, etc. But wait, the `\forVersion` does more than that! It also creates a series of comment environments `\begin{verA}/\end{verA}`, `\begin{verB}/\end{verB}`, `\begin{verC}/\end{verC}`, etc., where only the version for which this compile applies will be typeset, the others are commented out.

```

\numVersions{3}
\forVersion{b}
...
\begin{document}
...
Solve the equation for  $\$ \backslash vA\{x\} \backslash vB\{y\} \backslash vC\{z\} \$$ :
\[
\begin{verA}
2x + 4 = 7
\end{verA}
\begin{verB}
5y + 2 = 4
\end{verB}
\begin{verC}
3z - 2 = 2
\end{verC}
\]

669 \newif\ifeqglobalversion \eqglobalversionfalse
670 \newif\ifeqlocalversion \eqlocalversionfalse
671 \newif\if@templocalversion \@templocalversionfalse
672 \def\eqe@initializeMultiVersions{%
673   \let\save@message\message\let\message\@gobble
674   \@tfor\eqe@tmp:=ABCDEFGHIJKLMNQRSTUWXYZ\do{%
675     \csarg\let{v\eqe@tmp}\@gobble
676     \edef\exp@temp{\noexpand\excludecomment{ver\eqe@tmp}}\exp@temp
677     \csarg\let{Afterver\eqe@tmp Comment}\aebc@end
678   }\let\message\save@message
679 }
680 \AtBeginDocument{\let\eqe@initializeMultiVersions\relax}

```

(09/10/04) Trying to fix a bug in the case when the version selected is greater than the number of versions available for a given problem; that is, when modular arithmetic occurs (in `\selectVersion`).

```

681 \let\eqe@@onVersion\@empty
682 \def\eqe@onVersion{\g@addto@macro\eqe@@onVersion}
683 \let\eqe@@offVersion\@empty
684 \def\eqe@offVersion{\g@addto@macro\eqe@@offVersion}
685 \let\eqe@@holdTemp\@empty
686 \def\eqe@holdTemp{\g@addto@macro\eqe@@holdTemp}

```

Two commands to turn on and off versions (the `\v<LETTER>` and the `ver<LETTER>` environment).

Throughout the definitions below, we use `\csarg`, a command that is defined in the `comment` package.

```

687 \def\eqe@showArg#1{#1}
688 \def\eqe@turnOnComment#1{%
689 %   \csarg\let{v#1}\@empty
690   \csarg\let{v#1}\eqe@showArg
691   \edef\exp@temp{\noexpand\includecomment{ver#1}}\exp@temp
692 }
693 \def\eqe@turnOffComment#1{%
694   \csarg\let{v#1}\@gobble
695   \edef\exp@temp{\noexpand\excludecomment{ver#1}}\exp@temp
696   \csarg\let{Afterver#1Comment}\aebc@end
697 }

```

`\selVersion` Finally, the `\forVersion` command. `\selVersion` holds the version selected in upper-case.

```

698 \def\forVersion#1%
699 {%
700   \eqe@initializeMultiVersions
701   \let\eqe@onVersion\@empty
702   \let\eqe@offVersion\@empty
703   \global\eqglobalversiontrue
704   \setcounter{eq@count}{0}%
705   \uppercase{\edef\eq@selectedVersion{#1}}%
706   \edef\selVersion{\eq@selectedVersion}%
707   \@ifundefined{eq@nVersions}{\PackageInfo{eqexam}{%
708     \string\numVersions space has not been declared, \MessageBreak
709     taking the number of versions to be 2.}\def\eq@nVersions{2}}{}%
710   \loop
711     \stepcounter{eq@count}%
712     \expandafter\if\Alph{eq@count}\eq@selectedVersion
713       \xdef\eq@nSelectedVersion{\the\value{eq@count}}%
714       \setcounter{eq@count}{27}\fi
715     \ifnum\value{eq@count}<26\repeat
716     \ifnum\eq@nSelectedVersion >\eq@nVersions
717       \PackageError{eqexam}
718       {The value of \string\forVersion
719       \space(\eq@selectedVersion)\MessageBreak
720       exceeds the value of \string\numVersions\space
721       (\eq@nVersions)}%
722       {Decrease the value of \string\forVersion.}%
723     \fi
724     \setcounter{eq@count}{0}%
725     \let\save@message\message\let\message\@gobble
726     \loop
727       \stepcounter{eq@count}%
728       \csarg\let{After\Alph{eq@count}Comment}\relax
729       \lowercase
730       {%
731         \if#1\alph{eq@count}%

```

```

732             \eqe@turnOnComment{\Alph{eq@count}}}%
(09/10/04) Save the commands for turning on the version with \eqe@onVersion,
and for turning it off with \eqe@offVersion.
733             \edef\temp@exp{\noexpand
734                 \eqe@turnOnComment{\Alph{eq@count}}}%
735             \expandafter\eqe@onVersion\expandafter{\temp@exp}%
736             \edef\temp@exp{\noexpand
737                 \eqe@turnOffComment{\Alph{eq@count}}}%
738             \expandafter\eqe@offVersion\expandafter{\temp@exp}%
739         \else
740             \eqe@turnOffComment{\Alph{eq@count}}}%
741         \fi
742     }%
743     \ifnum\value{eq@count}<\eq@nVersions\repeat
744     \let\message\save@message
745 }

Let us assume version A initially, user with reset this in document.
746 \AtEndOfPackage{\numVersions{26}\forVersion{A}%
747     \eq@renditionOptions}

```

**\selectVersion** When an exam has questions in which the number of variations are not all the same, then you can locally change the version between problems. If the first argument is empty, the first variation is chosen. The syntax is

```
\selectVersion{2}{3}
```

This command says that the next problem has 3 variations, and here we select the second one.

```

748 \def\selectVersion#1#2{% #1 \le #2
749     \xdef\nLocalSelection{#1}\xdef\nLocalVersions{#2}%

```

If obey local version is in effect and the local selection is not empty, no modular arithmetic is needed in the first loop below of `\nLocalSelection`.

```

750     \let\needsModArith\eqe@YES
751     \ifeqobeylocalversion\ifx\nLocalSelection\@empty\else
752     \let\needsModArith\eqe@NO\fi\fi

```

When the solutions appear at the end of the document, the version may not match the version for the question. We need to use a private hook defined in `exerquiz` (and `eqexam.def`) to reproduce the same settings going into each solution at the end. So, we write the `\selectVersion` to the solution file.

```

753     \edef\exer@solnheadhook{%
754         \string\selectVersion{#1}{#2}}%

```

Turn off messaging.

```
755     \let\save@message\message\let\message\@gobble
```

Reset the selected version, the one selected in the preamble. A previous problem may have changed the version due to modular arithmetic.

```

756     \eqe@onVersion
757     \ifx\eqe@holdTemp\@empty\else

```

If `\eqe@holdTemp` is non-empty, this means that modular arithmetic was performed on the previous problem. We need to turn on the original choice, and turn off the temporary choice, then clear the command `\eqe@@holdTemp`.

```
758      \eqe@@onVersion\eqe@@holdTemp
759      \let\eqe@@holdTemp\@empty
760      \fi
```

If `\eqglobalversion` is true, then a `\forVersion` has been executed. If the number of versions declared by `\numVersions` is greater than the number of local versions for this problem, then we perform modular arithmetic to get an appropriate alternative. It may be necessary to temporarily put `\eqobeylocalversion` to true to accomplish, but we use change it back at the end.

```
761 \ifx\needsModArith\eqe@YES
762   \ifeqglobalversion\ifnum\eq@nSelectedVersion>\nLocalVersions
```

If we perform modular arithmetic, turn off original choice.

```
763       \eqe@@offVersion
Now perform mod arithmetic
764       {\count0=\eq@nSelectedVersion \count2=\count0
765        \advance\count0by-1 \divide\count0by\nLocalVersions
766        \multiply\count0by\nLocalVersions %\count2=\eq@nSelectedVersion
767        \advance\count2by-\count0
768        \xdef\nLocalSelection{\the\count2 }%
769        \ifeqobeylocalversion\else
770          \global\@templocalversiontrue
771          \global\eqobeylocalversiontrue\fi}%
772        \global\let\eqe@@holdTemp\@empty
773      \fi\fi
774 \fi
775   \ifeqobeylocalversion
776     \global\eqlocalversiontrue
777     \setcounter{eq@count}{0}%
```

If local selection is empty, use the version requested by `\forVersion`, modular arithmetic may have to be performed. arithmetic to resolve the matter

```
778       \ifx\nLocalSelection\@empty
779         \def\nLocalSelection{\eq@nSelectedVersion}%
780       \fi
```

If local selection exceeds number of local versions, use modular arithmetic to resolve the matter

```
781       \ifnum\nLocalSelection>\nLocalVersions
782         {\count0=\nLocalSelection
783          \count2=\count0
784          \advance\count0by-1 \divide\count0by\nLocalVersions
785          \multiply\count0by\nLocalVersions
786          \advance\count2by-\count0
787          \xdef\nLocalSelection{\the\count2 }}%
788       \fi
789       \let\save@message\message\let\message\@gobble
```

```

790     \loop
791         \stepcounter{eq@count}%
792         \csarg\let{Afterver\Alph{eq@count}Comment}\aebc@end
793         \lowercase
794         {%
795             \ifnum\value{eq@count}=\nLocalSelection
796                 \eqe@turnOnComment{\Alph{eq@count}}}%
797                 \edef\temp@exp{\noexpand
798                     \eqe@turnOffComment{\Alph{eq@count}}}%
799                 \expandafter\eqe@holdTemp\expandafter{\temp@exp}%
800             \else
801                 \eqe@turnOffComment{\Alph{eq@count}}}%
802             \fi
803         }%
804         \ifnum\value{eq@count}<\nLocalVersions\repeat
805         \let\message\save@message
806     \fi
807     \if@templocalversion\global\eqobeylocalversionfalse\fi
    added 09/10/03 reset back to default
808     \@templocalversionfalse
809     \let\message\save@message
810 }

```

## 9 Title Definitions from Web

```

\title Make Title Definitions taken from the Web package. This is to maintain compati-
\subject bility with Web.
\author 811 \@ifpackageloaded{web}{-}{-}%
\email 812 \@ifpackageloaded{hyperref}{-}{\let\texorpdfstring\@firstoftwo}%
\keywords 813 \let\web@save@title\title
\university 814 \def\title{\@ifnextchar[{\@web@title}{\@web@title[]}}
815 \def\@web@title[#1]#2{\gdef\webtitle{#2}}%
816 \@ifundefined{hypersetup}{-}{\hypersetup{pdftitle={#2}}}%
817 \def\webArg{#1}\ifx\webArg@empty\gdef\shortwebtitle{#2}\else
818 \gdef\shortwebtitle{#1}\fi\web@save@title{#2}}
819 \let\web@saved@author\author
820 \def\author#1{\gdef\webauthor{#1}}%
821 \@ifundefined{hypersetup}{-}{\hypersetup{pdfauthor={#1}}}%
822 \web@saved@author{#1}}
823 \def\subject{\@ifnextchar[{\@subject}{\@subject[]}}
824 \def\@subject[#1]#2{\def\webArg{#1}}%
825 \ifx\webArg@empty\gdef\shortwebsubject{#2}\else
826 \gdef\shortwebsubject{#1}\fi\gdef\websubject{#2}}%
827 \@ifundefined{hypersetup}{-}{\hypersetup{pdfsubject={#2}}}}
828 \def\email#1{\gdef\webemail{#1}}
829 \def\keywords#1{\gdef\webkeywords{#1}}%
830 \@ifundefined{hypersetup}{-}{\hypersetup{pdfkeywords={#1}}}}
831 \def\university#1{\gdef\webuniversity{#1}}

```

```

832 \def\copyrightyears#1{\gdef\webcopyrightyears{#1}}
833 \def\version#1{\gdef\web@version@value{#1}%
834 \edef\webversion{\ifx\web@version@value\empty\else
835 \noexpand\web@versionlabel\noexpand\
836 \noexpand\web@version@value\fi}%
837 }
838 \let\web@version@value\empty
839 \def\versionLabel#1{\def\web@versionlabel{#1}}
840 \versionLabel{Version}
841 \def\web@toc{Table of Contents}
842 \def\web@continued{cont.}
843 % set some defaults
844 \title{}\author{}\email{}\subject{}\keywords{}\university{}
845 \providecommand{\optionalPageMatter}[2][]{%
846 \def\optionalpagematter{#2}}
847 \def\optionalpagematter{}
848 }

```

`\date` L<sup>A</sup>T<sub>E</sub>X (T<sub>E</sub>X) defines a `\date` command that is also used by eqexam.

```

849 \def\duedate#1{\def\theduedate{#1}}
850 \duedate{}

```

`\duedate` In addition to these, we also define a `\duedate` macro, may be useful for writing assignments with a due date.

```

851 \def\duedate#1{\def\theduedate{#1}}
852 \duedate{}

```

`\thisterm` The command `\thisterm` can be used in the `\date` field to indicate the term of this test, for example, `\date{\thisterm, \the\year}` This command may be redefined to conform to your own academic terms.

```

853 \newcommand\thisterm{%
854 % if prior to June (Jan--May)
855 \ifnum\month<6Spring\else
856 % if prior to August (June-July)
857 \ifnum\month<8Summer\else
858 % August or later; if Sept--Dec
859 \ifnum\month>8Fall\else
860 % Month of August; if after 25th
861 \ifnum\day>25Fall\else
862 % if it's 25th or earlier in month of August
863 Summer\fi\fi\fi\fi
864 }

```

## 10 Identification Information

We define a series of commands in support of building an exam: Lines to identify the student and his/her student id (SID), the instructors email address, the name of the test and the course.

`\eqExamName` provides a line for the student to enter his/her name into the exam. The command `\examNameLabel` can be used to define the name label, the default is Name:

Will insert a text box as well if the `option` is taken in addition to `nosolutions` and with `solutionsafter` not taken. This macro defines `\eq@ExamName`, which actually contains the code. The first (optional) parameter is passed to `\insTxtFieldIdInfo`, and can be used to change the appearance of the text field created; the second required parameter is the width of the field.

```

865 \newcommand{\examAnsKeyLabel}[1]{%
866     \def\examAnsKeyLabel{\ifanswerkey\space #1\fi}}
867 \examAnsKeyLabel{Answer Key}
868 \newcommand\examNameLabel[1]{\gdef\examNameLabel{#1\examAnsKeyLabel}}
869 \examNameLabel{Name:}
870 \newcommand{\idinfoHighlight}[1]{\def\eqe@idinfohl{#1}}
871 \idinfoHighlight{\underbar}
872 \newcommand\eqExamName[2][]{%
873     \def\eqExamName@argi{#1}\def\eqExamName@argii{#2}}
874 \def\eq@ExamName{\bgroup
875     \settowidth\eq@tmplengthA{\examNameLabel\ }%
876     \@tempdima=\eqExamName@argii\relax
877     \advance\@tempdima by-\eq@tmplengthA
878     \eqe@idinfohl{\makebox[\eqExamName@argii][l]{\examNameLabel}}%
879     \expandafter\insTxtFieldIdInfo\expandafter[\eqExamName@argi]%
880     {\@tempdima}{IdInfo.Name}\egroup}

```

Here we set the field to be a required field with width of 2.25 inches

```

881 \eqExamName[\Ff\FfRequired]{2.25in}

```

`\eqSID` provides a line for the student to enter his/her ID number (SID).

Will insert a text box as well if the `option` is taken in addition to `nosolutions` and with `solutionsafter` not taken. The first (optional) parameter is passed to `\insTxtFieldIdInfo`, and can be used to change the appearance of the text field created; the second required parameter is the width of the field.

```

882 \newcommand\examSIDLabel[1]{\gdef\examSIDLabel{#1}}
883 \examSIDLabel{SID:}
884 \newcommand\eqSID[2][]{\def\eqSID@argi{#1}\def\eqSID@argii{#2}}
885 \def\eq@SID{\bgroup\settowidth\eq@tmplengthA{\examSIDLabel\ }%
886     \@tempdima=\eqSID@argii\relax\advance\@tempdima by-\eq@tmplengthA
887     \eqe@idinfohl{\makebox[\eqSID@argii][l]{\examSIDLabel}}%
888     \expandafter\insTxtFieldIdInfo\expandafter[\eqSID@argi]%
889     {\@tempdima}{IdInfo.SID}\egroup}

```

Here we set the field to be a required field with width of 2.25 inches

```

890 \eqSID[\Ff\FfRequired]{2.25in}

```

`\eqEmail` provides a line for the student to enter his/her email address. Useful for documents submitted by email, the instructor can reply.

Will insert a text box as well if the `option` is taken in addition to `nosolutions` and with `solutionsafter` not taken. The first (optional) parameter is passed to `\insTxtFieldIdInfo`, and can be used to change the appearance of the text field created; the second required parameter is the width of the field.



```

891 \newcommand\examEmailLabel[1]{\gdef\@examEmailLabel{#1}}
892 \examEmailLabel{Email:}
893 \newcommand\eqEmail[2][]{%
894     \def\eqExam@argi{#1}\def\eqEmail@argii{#2}}
895 \gdef\eq@Email{\bgroup\settowidth\eq@tmplengthA{\@examEmailLabel\ }%
896     \@tempdima=\eqEmail@argii\relax\advance\@tempdima by-\eq@tmplengthA
897     \underbar{\makebox[\eqEmail@argii][l]{\@examEmailLabel}}%
898     \expandafter\insTxtFieldIdInfo\expandafter[\eqEmail@argi]%
899     {\@tempdima}{IdInfo.email}\egroup}

```

Here we set the field to be a field with width of 2.25 inches. (Not set to be a required field.)

```
900 \eqEmail{2.25in}
```

**\insTxtFieldIdInfo** The above macros (**\eqExamName**, **\eqSID** and **\eqEmail**) all call this macro, which inserts a Acroforms text field if the **option** is taken in addition to **nosolutions** and with **solutionsafter** not taken.

The first (optional) parameter is used to change the appearance of the text field. The second parameter is the width of the field, and the third is the field name.

```

901 \def\insTxtFieldIdInfo[#1]#2#3{%
902     \@ifundefined{@quiz}{}{\if\eq@online\eqe@YES
903         \ifeq@nosolutions\ifeq@solutionsafter\else
904             \raisebox{-1bp}{\makebox[0pt][r]{%
905                 \textField[BC]{#1}{#3}{#2}{11bp}}}%
906             \fi\fi\fi
907     }%
908 }

```

**\SubmitInfo** is required when the **email** option is taken, and should appear in the preamble. The first argument is the URL to the **eqAttach.asp** code on the server, and the second is the email of the instructor is to receive the results. (Multiple recipients can be specified by separating each with a comma.)

```

909 \def\SubmitInfo#1#2{%
910     \def\EqExam@SubmitURL{#1}\def\@EmailInstr{#2}%
911 }

```

**\EmailCourseName** is used to specify the course name of the course. The default value for this is **\websubject**, obtained from the **\subject** macro used in the preamble; however, if you want a different name in the email, perhaps with more information included, you can redefine the value using this macro.

```

912 \def\EmailCourseName#1{\def\@EmailCourseName{#1}}

```

Here's the default value.

```

913 \EmailCourseName{\websubject}

```

**\EmailExamName** is used to specify the exam name of the course. The default value for this is **\webtitle**, obtained from the **\title** macro used in the preamble; however, if you want a different name in the email, perhaps with more information included, you can redefine the value using this macro. (Multiple recipients can be specified by separating each with a comma.)

```

914 \def\EmailExamName#1{\def\@EmailExamName{#1}}
    Here's the default value.
915 \EmailExamName{\webtitle}
\EmailSubject  The document author mail want a custom subject in the email, instead of the
                standard one. By using this macro, he can design his own email subject.
916 \def\EmailSubject#1{\def\@EmailSubject{#1}}
    Here's the default value, which generates no custom subject line.
917 \EmailSubject{}
    In this case eqAttach.asp inserts the standard one.

    Exam Results: \webtitle of \websubject

    The email would read like "Exam Results: Test 1 of Calculus I", for exam-
    ple.
\ServerRetnMsg  Unless submitted in silent mode, the eqAttach.asp returns a message ac-
                knowledging the receipt of the data. \ServerRetnMsg is used to customize this
                message.
918 \def\ServerRetnMsg#1{\def\@ServerRetnMsg{#1}}
    Here's the default value, which generates no custom return message
919 \ServerRetnMsg{}
\SubmitButtonLabel  is the label that appears on the submit button.
920 \def\SubmitButtonLabel#1{\def\@SubmitButtonLabel{#1}}
    Here's the default value.
921 \SubmitButtonLabel{Submit}
\SubmitButton  is the macro that provides the submit button when the email option is taken. It
                appears automatically at the top of the first page of the exam, and appears only
                if nosolutions has has been taken, and solutionsafter has not been taken.
922 \let\priorSubmitJS\@gobble
923 \let\postSubmitJS\@empty
924 \def\SubmitButton
925 {%
926     \ifx\use@email\eqe@YES\ifeq@nosolutions\ifeq@solutionsafter\else
927         \makebox[Opt][l]{\pushButton
928             [CA{\@SubmitButtonLabel}\A{\JS{%
929                 var _eqEok2Submit = true;\r
930                 var aSubmitFields = new Array("eqexam", "IdInfo");\r
931                 \priorSubmitJS\r
932                 if(_eqEok2Submit) this.submitForm("\EqExam@SubmitURL",
933                     true, false, aSubmitFields);\r
934                 \postSubmitJS
935             }}{Submit}{1.5in}{16bp}}%
936     \makebox[Opt][l]{\textField[F\FHIDDEN\DV{\@EmailInstr}
937         \V{\@EmailInstr}]{IdInfo.mailTo}{11bp}{11bp}}%
938     \makebox[Opt][l]{\textField[F\FHIDDEN\DV{\@EmailCourseName}
939         \V{\@EmailCourseName}]{IdInfo.courseName}{11bp}{11bp}}%

```

```

940      \makebox[Opt][l]{\textField[F\FHidden\DV{\@EmailExamName}
941        \V{\@EmailExamName}]{IdInfo.examName}{11bp}{11bp}}%
942      \makebox[Opt][l]{\textField[F\FHidden\DV{\@EmailSubject}
943        \V{\@EmailSubject}]{IdInfo.subject}{11bp}{11bp}}%
944      \makebox[Opt][l]{\textField[F\FHidden\DV{\@ServerRetnMsg}
945        \V{\@ServerRetnMsg}]{IdInfo.retnmsg}{11bp}{11bp}}%
946      \fi\fi\fi
947 }

948 \def\thequizno{\if\probstar*\Alph{quizno}\else\alph{quizno}\fi}
949 \def\linkContentFormat{%
950   \if\probstar*\Alph{quizno}\else\alph{quizno}\fi}
951 \def\linkContentWrapper{(\hfil\linkContentFormat\hfil)}%
952 \def\Ans@r@l@Defaults
953 {%
954   \BC{}\S{S}\W{1}\Ff{FfNoToggleToOff}
955   \textSize{12}\textColor{0 g}
956 }

(2014/03/12) Added \eq@hspanner\ignorespaces, in the next two commands.
Fixes problems with spacing following the choice box.

957 \newcommand{\optsRadioBtnf}[1]{\def\eqe@optsRadiof{#1}}
958 \newcommand{\optsRadioBtnl}[1]{\def\eqe@optsRadiol{#1}}
959 \let\eqe@optsRadiof@empty\let\eqe@optsRadiol@empty
960 \def\eqExam@Ans@sq@l{\leavevmode
961   \if\eq@listType1\stepcounter{quizno}\else\ifwithinsoldoc
962     \stepcounter{quizno}\else\refstepcounter{quizno}\fi\fi
963   \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@l}%
964   \sbox{\eq@tmpbox}{\eq@l@l}\eq@tmpdima=\wd\eq@tmpbox
965   \def\link@@Content{\linkContentWrapper}%
966   \hangindent=\eq@tmplength\hangafter=1\relax
967   \edef\fieldName{%
968     \if\probstar*eqexam.\curr@quiz.\theeqquestionnoi.part\thepartno%
969     \else
970       eqexam.\curr@quiz.\theeqquestionnoi%
971     \fi
972   }%
973   \if\eq@online\eqe@YES\relax
974   \makebox[Opt][l]{%
975     \radio@@Button{\presets{\eqe@optsRadiol}}{\fieldName}%
976     {\eq@tmpdima}{\RadioFieldSize}{\Ans@choice\alph{quizno}}%
977     {\eq@protect\A}{\eq@setWidgetProps\eq@l@check@driver}%
978     {\Ans@r@l@Defaults\every@RadioButton\every@qRadioButton}}%
979   \else
980     \edef\@linkcolor{\@nolinkcolor}%
981     \fi
982   \ifeq@nosolutions\edef\@linkcolor{\@nolinkcolor}\fi
983   \textcolor{\@linkcolor}{\makebox[\eq@tmpdima]{\link@@Content}}%
984   \Ans@proofing{\eq@tmpdima}%
985   \eq@hspanner\ignorespaces
986 }

```

(2019/10/28) Legacy assignment, in case eqexam.def/exerquiz are still using  
 $\backslash\text{eq@RadioCheck@driver}$ .

```

987 \ifundefined{eq@RadioCheck@driver}{%}
988 {\let\eq@Radio@driver\eq@RadioCheck@driver}
989 \def\eqExam@Ans@sq@f{\if\eq@listType1\stepcounter{quizno}\else
990   \ifwithinsoldoc\stepcounter{quizno}\else
991     \refstepcounter{quizno}\fi\fi
992   \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@f}%
993   \eq@tmpdima=\wd\eq@tmpbox%
994   \hangindent=\eq@tmplength\hangafter=1\relax
995   \if\eq@online\eqe@NO\previewtrue
996     \insertGrayLetters
997     \Ans@sq@f@driver
998   \else
999     \ifanswerkey\previewtrue
1000       \Ans@sq@f@driver
1001     \else
1002       \edef\fieldName{%
1003         \if\probstar*eqexam.\curr@quiz.\thequestionnoi.%
1004         part\thepartno%
1005       \else
1006         eqexam.\curr@quiz.\thequestionnoi%
1007       \fi
1008       }\insertGrayLetters
1009       \radio@@Button{\presets{\eqe@optsRadiof}}%
1010       {\fieldName}{\RadioFieldSize}%
1011       {\RadioFieldSize}{\Ans@choice\alph{quizno}}{\eq@protect\A}%
1012       {\eq@setWidgetProps\eq@Radio@driver}%
1013       {\@@Ans@sq@f@Defaults\Ans@sq@f@Actions\every@RadioButton
1014       \every@sqRadioButton\insert@circlesymbol}%
1015     \fi
1016   \fi
1017   \Ans@proofing{\RadioFieldSize}%
1018   \eq@hspanner\ignorespaces
1019 }
1020 \newcommand{\optsCkBxf}[1]{\def\eqe@optscbf{#1}}
1021 \newcommand{\optsCkBxl}[1]{\def\eqe@optschl{#1}}
1022 \let\eqe@optschl\empty\let\eqe@optscbf\empty
1023 \def\eqExam@Ans@ck@sq@l{\leavevmode
1024   \if\eq@listType1\stepcounter{quizno}\else
1025   \ifwithinsoldoc\stepcounter{quizno}\else
1026     \refstepcounter{quizno}\fi\fi
1027   \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@l}%
1028   \sbox{\eq@tmpbox}{\eq@l@l}\eq@tmpdima=\wd\eq@tmpbox
1029   \def\link@@Content{\linkContentWrapper}%
1030   \hangindent=\eq@tmplength\hangafter=1\relax
1031   \edef\fieldName{%
1032     \if\probstar*eqexam.\curr@quiz.\thequestionnoi.%
1033     part\thepartno.\alph{quizno}%

```

```

1034         \else
1035             eqexam.\curr@quiz.\theeqquestionnoi.\alph{quizno}%
1036         \fi
1037     }%
1038     \if\eq@online\eqe@YES\relax
1039     \makebox[Opt][l]{\check@Box{\presets{\eqe@optscbl}}{\fieldName}%
1040         {\eq@tmpdima}{\RadioFieldSize}{\Ans@choice\alph{quizno}}%
1041         {\eq@protect\A}{\eq@setWidgetProps\eq@l@check@driver}%
1042         {\Ans@r@l@Defaults\every@RadioButton\every@qRadioButton}}%
1043     \else
1044         \edef\@linkcolor{\@nolinkcolor}%
1045     \fi
1046     \ifeq@nosolutions\def\@linkcolor{\@nolinkcolor}\fi
1047     \textcolor{\@linkcolor}{\makebox[\eq@tmpdima]{\link@Content}}%
1048     \Ans@proofing{\eq@tmpdima}%
1049     \eq@hspanner\ignorespaces
1050 }
1051 \def\eqExam@Ans@ck@sq@f{\if\eq@listType1\stepcounter{quizno}\else
1052     \ifwithinsoldoc\stepcounter{quizno}\else
1053         \refstepcounter{quizno}\fi\fi
1054     \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@f}%
1055     \eq@tmpdima=\wd\eq@tmpbox%
1056     \hangindent=\eq@tmplength\hangafter=1\relax
1057     \if\eq@online\eqe@NO\previewtrue
1058         \insertGrayLetters
1059         \Ans@sq@f@driver
1060     \else
1061         \ifanswerkey\previewtrue
1062             \Ans@sq@f@driver
1063         \else
1064             \edef\fieldName{%
1065                 \if\probstar*eqexam.\curr@quiz.\theeqquestionnoi.%
1066                 part\thepartno.\alph{quizno}%
1067             \else
1068                 eqexam.\curr@quiz.\theeqquestionnoi.\alph{quizno}%
1069             \fi
1070             }\insertGrayLetters
1071             \mbox{\check@Box{\presets{\eqe@optscbf}}{\fieldName}%
1072                 {\RadioFieldSize}{\RadioFieldSize}%
1073                 {\Ans@choice\alph{quizno}}{\eq@protect\A}%
1074                 {\eq@setWidgetProps\eq@Radio@driver}%
1075                 {\@@Ans@sq@f@Defaults\Ans@sq@f@Actions\every@RadioButton
1076                 \every@sqRadioButton}}%
1077         \fi
1078     \fi
1079     \Ans@proofing{\RadioFieldSize}%
1080     \eq@hspanner\ignorespaces
1081 }

```

`\optsMlTextField` is used to pass options to the multi-line text field.

```

1082 \def\optsMlTextField#1{\def\eqe@optsmltf{#1}}
1083 \let\eqe@optsmltf\@empty
1084 \def\eqExamPriorVspace#1{%
1085     \edef\fieldName{%
1086         \if\probstar*eqexam.\curr@quiz.%
1087         \theeqquestionnoi.part\thepartno.solution%
1088         \else
1089         eqexam.\curr@quiz.\theeqquestionnoi.solution%
1090         \fi
1091     }%
1092     \nobreak\noindent\textField[BC]{\presets{\eqe@optsmltf}
1093         \Ff\FfMultiline}{\fieldName}{\linewidth}{#1}\@gobble
1094 }

```

Test to see if `exerquiz` is loaded. If not, we input the ‘stand alone’, `eqalone.def`, followed by `eqexam.def`. The latter definition file is maintained in `exerquiz.dtx` under the `eqexam` option.

```

1095 \@ifpackageloaded{exerquiz}{%
1096     \let\Ans@sq@l\eqExam@Ans@sq@l
1097     \let\Ans@sq@f\eqExam@Ans@sq@f
1098     \let\Ans@ck@sq@l\eqExam@Ans@ck@sq@l
1099     \let\Ans@ck@sq@f\eqExam@Ans@ck@sq@f
1100     \def\eqexheader@wrapper{\makebox[0pt][r]{%
1101         \hypertarget{qex.\the@exno}{\eqexheader}}}%
1102     \if\eq@online\eqe@YES\relax
1103         \newcounter{cntfillin}%
1104         \let\eqPriorVspace\eqExamPriorVspace
1105     \fi
1106 }%
1107 {%
1108     \input{eqalone.def}
1109     \input{eqexam.def}
1110
1110     \@ifl@ter{def}{eqexam}{\eqexamdefReq}{%
1111         {\PackageWarningNoLine{eqexam}
1112         {This version of eqexam requires eqexam.def\MessageBreak
1113         dated \eqexamdefReq\space or later}}
1114     \def\eqexheader@wrapper{\makebox[0pt][r]{\eqexheader}}
1115 }

```

We wrote `\begin{eqequestions}` to the top of the solutions file (`\jobname.sol`).

```

1116 %\writeBeginEqeQuestions

```

If the `vspacewithsolns` is in effect, we write solutions to the end of the document.

```

1117 \AtBeginDocument{%
1118     \ifvspacewithsolns\writeAllAnsAtEnd\else
1119     \ifeqfortextbook\writeAllAnsAtEnd\else
1120     \writeWithSolDocTrue\fi\fi
1121 }

```

We execute `\vspacewithkeyOff`, which sets `\ifkeepdeclaredvspacing` to false, the default behavior of `eqexam` before the new feature.

```
1122 \vspacewithkeyOff
1123 \end{package}
```

## 11 Stand alone Code

```
1124 \begin{standalone}
```

Now we begin the listing of the stand alone code. This code is necessary if `exerquiz` has not been loaded, which is the case if there is no PDF options or if the `pdf` option is taken.

Many of the following definitions are given in `eforms`, which was recently separated from `exerquiz` and is now maintained as a separate package.

```
1125 \ProvidesFile{eqalone.def}
1126 [2012/25/01 v3.0t Minimal code used by eqexam (dps)]
1127 \ifundefined{eq@tmpbox}{\newsavebox{eq@tmpbox}}{}% defined in eforms
1128 \ifundefined{eq@tmpdima}{\newdimen{eq@tmpdima}}{} % defined in eforms
1129 \def\RadioFieldSize{11bp}
1130 \newdimen{eqcenterWidget}
```

This macro is used to vertically center the response box on the line. Seems to work well.

```
1131 \def\centerWidget
1132 #1{%
1133     \eqcenterWidget=#1
1134     \eqcenterWidget=.5\eqcenterWidget
1135     \advance\eqcenterWidget by-4bp
1136 }
```

`\eqe@Bbox` When the `preview` option has been used, draw a frame box around the bounding rectangle.

```
1137 \def\eqe@BboxRect#1#2{\hbox{\ckbox@Color\ vbox{\hrule width #1
1138     \hbox to#1{\vrule height#2\hfill\vrule height#2}\vfill\hrule}}}
1139 \def\useRectForMC{\let\Bbox\eqe@BboxRect}
1140 \def\selectedMC{\useRectForMC}
```

`\Rect` is used internally to color a link.

```
1141 \def\Rect#1{\textcolor{@linkcolor}{#1}}
```

`\ReturnTo` The auxiliary file `eqexam.def`, created by `exerquiz`, writes `\ReturnTo` to the SOL file in the form `\ReturnTo{page.1}{\mbox{}}`. We want to remove the `\mbox` because it causes, at times, more vertical space that is wanted in an exam document.

```
1142 \def\eqe@striphbox\mbox#1{#1}
1143 \newcommand{\ReturnTo}[2]{\eq@fititin{\eqe@striphbox#2}}
```

## 12 Switching proofing symbols

`\proofingsymbol` The definition of the proofing symbol, this symbol marks the correct answer of a

multiple choice question when the `proofing` option is used.

```
1144 \newcommand{\proofingsymbol}[1]{%
1145     \def\@proofingsymbol{\textcolor{\@proofingsymbolColor}{#1}}}
```

This is the answers macro for the link-style and is called from the `eqexam.def` file.

```
1146 \</standalone>
```

```
1147 \<*package>
```

In response to the `allowcircmc`, we load `lcircle10` and use the ‘h’ and ‘x’ glyph.

```
1148 \def\selectedMC{\useRectForMC}
1149 \ifallowcircmc
1150     \font\eqe@lcir=lcircle10 at 12pt
1151     \bgroup
```

Get the width of the ‘h’, the circle has zero height and depth. Set the diameter and radius of the circle.

```
1152     \setbox0=\hbox{\eqe@lcir h}
1153     \xdef\eqe@cirDiam{\the\wd0}
1154     \@tempdima=.5\wd0
1155     \xdef\eqe@cirRadius{\the\@tempdima}
1156 \egroup
```

A command to use the circle (h) and the filled circle (x).

```
1157 \def\circ@Glyph#1#2{\hbox{\smash{\raisebox{\eqe@cirRadius}%
1158     {\makebox[\eqe@cirDiam]{\llap{\ckcir@Color
1159     \rlap{\eqe@lcir#1}\hskip#2\relax}}}}}}
```

Added preview color to the circle version of the bounding box. The color accessed through `\previewColor`, its default is black.

```
1160 \def\eqe@BboxCirc#1#2{\hbox{\ckcir@Color{\circ@Glyph{h}{0pt}}}}
```

For proofing purposes, prepare the filled circle in the form of the command `\circProofingForCirc`.

```
1161 \def\circProofingForCirc{\centerWidget\RadioFieldSize
1162     \lower\eqcenterWidget\circ@Glyph{x}{1.4pt}}
1163 \if\eq@online\eqe@YES
```

If online (online or email option), we cancel these commands.

```
1164     \def\useCircForMC{\let\Bbox\eqe@BboxCirc\useMCCircles}
1165     \def\selectedMC{\useCircForMC}
1166     \@ifundefined{eqe@BboxRect}
1167         {\def\useRectForMC{\let\Bbox\ef@Bbox\useMCRects}}
1168         {\def\useRectForMC{\let\Bbox\eqe@BboxRect\useMCRects}}
1169 %     \let\useRectForMC\relax
1170 \else % if not online
1171     \@ifundefined{eqe@BboxRect}
1172         {\def\useRectForMC{\let\Bbox\ef@Bbox}}
1173         {\def\useRectForMC{\let\Bbox\eqe@BboxRect}}
1174     \def\useCircForMC{\let\Bbox\eqe@BboxCirc}
1175     \def\selectedMC{\useCircForMC}
1176 \fi
```



```

1177 \def\useCircForProof{\symbolchoice{circle}%
1178 \proofingsymbol{\circProofingForCirc}}
1179 \else
    If allowcircmc is not taken
1180 \if\eq@online\eqe@YES
1181 \def\useCircForMC{\let\Bbox\eqe@BboxCirc\useMCCircles}
1182 \ifundefined{eqe@BboxRect}
1183 {\def\useRectForMC{\let\Bbox\ef@Bbox\useMCRects}}
1184 {\def\useRectForMC{\let\Bbox\eqe@BboxRect\useMCRects}}
1185 \else
1186 \@ifundefined{eqe@BboxRect}
1187 {\def\useRectForMC{\let\Bbox\ef@Bbox}}
1188 {\def\useRectForMC{\let\Bbox\eqe@BboxRect}}
1189 \def\useCircForMC{\useRectForMC}
1190 \def\selectedMC{\useRectForMC}
1191 \fi
1192 \let\useCircForProof\relax
1193 \fi

\useRectForMS \useRectForMS When declared, rectangles are used for multiple selection, simi-
\useCircForMS larly, \useCircForMS uses circles if allowcirc4mc is in effect.

1194 \newif\ifuserectforms
1195 \def\useRectForMS{\userectformstrue}
1196 \def\useCircForMS{\userectformsfalse}

1197 \newcommand{\useCheckForProof}{\symbolchoice{check}%
1198 \proofingsymbol{\ding{52}}}}
1199 \useCheckForProof

Similarly, for a cross, we get a nice ‘handwritten’ cross \ding{56}, !when we latex
the document; for the online or email option, we get a simple cross in the active
radio button field.

1200 \newcommand{\useCrossForProof}{\symbolchoice{cross}%
1201 \proofingsymbol{\raisebox{-1pt}{\rlap{\kern-1pt\Large\ding{56}}}}}

```

## 13 The Main Code

We now continue with the main package. Mostly, we define macros specific to the `eqexam` package: define the `problem` and `problem*` environments, macros for calculating totals per page, etc.

```

1202 \def\MCcolor{black}
1203 \def\Ans@sq@l@driver{\edef\@linkcolor{\MCcolor}%
1204 \Rect{\makebox[\eq@tmpdima]{\linkContentWrapper}}}%
1205 \Ans@proofing{\eq@tmpdima}%
1206 }

```

This is the answers macro for the form-style and is called from the `eqexam.def` file.

```

1207 \def\Ans@sq@f@driver{%
1208     \centerWidget\RadioFieldSize
1209     \leavevmode\lower\eqcenterWidget\Bbox %\eqe@Bbox
1210         {\RadioFieldSize}{\RadioFieldSize}%
1211     \Ans@proofing{\RadioFieldSize}%
1212 }

```

Write quiz solutions to the exercise solutions file

```

1213 \def\eq@sq@label{\string\textbf{Solution to Quiz:}}
1214 \def\sq@label{\eq@sq@label}

```

\writeToSolnFile General purpose command for writing to the solution file.

\preExamSolnHead Executed just before a user friendly name

\examSolnHeadFmt Format for the user friendly name

\postExamSolnHead Executed just after a user friendly name

```

1215 \let\quiz@solns\ex@solns
1216 \newcommand{\preExamSolnHead}{\goodbreak\noindent}
1217 \newcommand{\examSolnHeadFmt}[1]{\textbf{#1}}
1218 \newcommand{\postExamSolnHead}{\par\medskip}

```

Write to solution file if *not* solutions-only

```

1219 \@ifundefined{ifOKToWriteExamData}{\newif\ifOKToWriteExamData
1220     \OKToWriteExamDatatrue}{\}
1221 \newcommand{\writeToSolnFile}[1]{%
1222     \ifsolutionsonly\else\ifOKToWriteExamData
1223         \let\quiz@solns\ex@solns
1224         \set@display@protect
1225         \eqe@IW0\quiz@solns{#1}%
1226         \set@typeset@protect
1227     \fi\fi}
1228 \let\writeT@SolnFile\writeToSolnFile

```

We will write all solutions to the .sol auxiliary file.

```

1229 \def\eqe@writetoSolns#1{% dpsD17
1230     \ifsolutionsonly\else
1231         \ifOKToWriteExamData
1232             \set@display@protect
1233             \eqe@IW0\quiz@solns{\string\preExamSolnHead
1234                 \string\examSolnHeadFmt{#1}\string\postExamSolnHead}%
1235             \set@typeset@protect\fi\fi}
1236 \def\eqe@writetoAux#1{%
1237     \set@display@protect
1238     \eqe@IW0\@auxout{#1}%
1239     \set@typeset@protect}

```

Turn off interactivity of short quiz.

This macro is defined in `exerquiz`, but has a little different definition for `eqexam`.

```

1240 \def\Ans@proofing
1241 #1{%
1242   \ifeq@proofing\if\Ans@choice1\relax
1243     \llap{\rlap{\, \@proofingsymbol}\hskip#1\relax}%
1244   \fi\fi
1245 }

\eq@ExamLastPage This macro gets the page number of the last page of the exam. It is read in
                  through a macro definition made and written to the .aux file.

1246 \def\eq@ExamLastPage{\csname eqExamLastPage\endcsname}

\nPagesOnExam is the number of pages in the exam.

1247 \newcommand{\nPagesOnExam}{\csname eqExamLastPage\endcsname}

\nPagesOnQues is the number of pages in the questions.

1248 \newcommand{\nPagesOfQues}{\ifvspacewithsolns
1249   \csname eqExamQuesLastPage\endcsname\else
1250   \csname eqExamLastPage\endcsname\fi}

\nPagesOnSols is the number of pages of solutions.

1251 \newcommand{\nPagesOfSols}{\def\eqExamNumPagesSolns{0}%
1252   \csarg\ifx{eqExamLastPage}\relax\else
1253   \csarg\ifx{eqExamLastPage}\relax\else
1254   {\count\z@=\nPagesOnExam\relax\advance
1255    \count\z@-\nPagesOfQues\relax
1256    \xdef\eqExamNumPagesSolns{\the\count\z@}}\fi\fi
1257   \eqExamNumPagesSolns}

\lastPageOfExam Returns the page number of the end of the exam with a name of #1.

\firstPageOfExam Returns the page number of the beginning of the exam with a name of #1.

1258 \newcommand{\lastPageOfExam}[1]{\pageref{#1PageEnd}}
1259 \newcommand{\firstPageOfExam}[1]{\pageref{#1PageBegin}}

The last two commands are meant to produce typeset numbers; however, there
is a need to convert these to numbers that tex's registers can manipulate. Here
\eqe@defNumRefii goes. \eqe@defNumRefii takes its argument and strips away the other arguments
of \pageref; it picks off the second of two or five, depending if hyperref is loaded.
It defines a macro \csname nRefii@#1\endcsname whose value is a page number
of the referenced object.

1260 \newcommand{\eqe@defNumRefii}[1]{%
1261   \@ifundefined{hyperref}{\let\@getsecondOf\@secondoftwo}
1262   {\let\@getsecondOf\@secondoffive}%
1263   \@ifundefined{r@#1}{%
    If the reference r@#1 is undefined, define the value to be 0
1264     \csarg\gdef{nRefii@#1}{0}}}%

```

If the reference `r@#1` is defined, define the value to be the second argument of `\r@#1` expanded

```
1265 \expandafter\xdef\csname nRefii@#1\endcsname
1266 {\expandafter\expandafter\expandafter
1267 \getsecondOf\csname r@#1\endcsname}%
1268 }%
1269 }
```

`\eqe@numRefii` takes one argument, the control name. Its value is zero or `\nRefii@#1`. This expands to a number in all cases. It can be used in tex comparisons.

```
1270 \def\eqe@numRefii#1{%
1271 \csarg\ifx{nRefii@#1}\relax 0}else
1272 \@nameuse{nRefii@#1}\fi}
```

`\numLastPageOfExam` This is the user-interface to acquiring the first and last page numbers of the exam  
`\numFirstPageOfExam` with name #1. These can be used in comparisons, e.g.

```
\rfooteqe{\ifnum\value{page}<\numLastPageOfExam{<myTest>}}%
\textbf{Test Continues}\fi}

1273 \newcommand{\numLastPageOfExam}[1]{\eqe@numRefii{#1PageEnd}}
1274 \newcommand{\numFirstPageOfExam}[1]{\eqe@numRefii{#1PageBegin}}

1275 \newcommand{\makeRefsNums}{%
1276 \@ifundefined{thePartNames}{\begingroup
1277 \def\##1{\PackageInfo{eqexam}{processing exam: ##1}%
1278 \eqe@defNumRefii
1279 {##1PageEnd}\eqe@defNumRefii{##1PageBegin}}}%
1280 \thePartNames
1281 \endgroup}%
1282 }

1283 \AtBeginDocument{\makeRefsNums}
```

## 13.1 Running Heads and Feet

We develop a series of macros for creating running headers and footers for the exam.

`\lheadeq` Set the left, center, and right running headers.

```
\cheadeqe 1284 \newcommand{\lheadeq}[1]{\def\eq@lhead{#1}}
\rheadeq 1285 \lheadeq{\shortwebsubject/\shortwebtitle}
1286 \newcommand{\cheadeqe}[1]{\def\eq@chead{#1}}
1287 \cheadeqe{-- Page \arabic{page} of {\nPagesOnExam} --}
1288 \newcommand{\rheadeq}[1]{\def\eq@rhead{#1}}
```

The default is `\eq@ExamName`, which is defined by `\eqExamName`, the default displays the word "Name" and an underlined horizontal space for the student to enter his/her name.

```
1289 \rheadeq{\eq@ExamName}
```

`\lhead` These are the original names for the headers, we'll keep them if `fancyhdr` is not  
`\chead` already loaded to maintain compatibility with previous versions of `eqexam`. The  
`\rhead` use of these commands is *discouraged*.

```
1290 \ifpackageloaded{fancyhdr}{\def\lhead\lheadqe
1291 \let\lhead\lheadqe
1292 \let\chead\cheadeqe
1293 \let\rhead\rheadeqe
1294 }
```

`\runExamHeader` The running header of the exam, may be redefined.

```
\eqExamRunHead 1295 \newcommand{\runExamHeader}{\eq@lhead\hfill\eq@chead\hfill\eq@rhead}
1296 \newcommand\eqExamRunHead{%
1297 \addtolength\textwidth{\oddsidemargin}%
1298 \noindent\hspace*{-\oddsidemargin}\makebox[\textwidth]
1299 {\runExamHeader}%
1300 }
```

**Running footers.** One or two users wanted to use running footers, so here they are.

`\lfooteqe` There has been some demand for running footers. You have to be a little careful,  
`\cfooteqe` `eqexam` uses the footer for the command `\settotalsbox`, which puts in the totals  
`\rfooteqe` for the pages either on the left (`totalsonleft`) or right (`totalsonright`) side,  
depending on the option.

```
1301 \newcommand{\lfooteqe}[1]{\def\eq@lfoot{#1}}
1302 \lfooteqe{}
1303 \newcommand{\cfooteqe}[1]{\def\eq@cfoot{#1}}
1304 \cfooteqe{}
1305 \newcommand{\rfooteqe}[1]{\def\eq@rfoot{#1}}
1306 \rfooteqe{}
```

`\runExamFooter` Considering the defaults for the left, center, and right footer elements, the default  
footer contributes nothing, except inserting `\settotalsbox` (see the definition of  
`\@oddfoot`)

```
1307 \newcommand{\runExamFooter}{\eq@lfoot\hfill\eq@cfoot\hfill\eq@rfoot}
```

**Running headers for solutions.** We provide a special set of headers for the  
solution pages. The document author needs to manage running footers for the  
solution pages.

`\lheadSol` The running header of the exam, when solutions are included at the end of the doc-  
`\cheadSol` ument, perhaps for posting the solutions to the exam, or publication of a “pretest”.  
`\rheadSol` Note that `\eqsolutionshook` is defined in `exerquiz/eqexam.def`. May be rede-  
`\runExamHeaderSol` fined.

```
\eqsolutionshook 1308 \newcommand{\lheadSol}[1]{\def\eq@lheadSol{#1}}
1309 \lheadSol{\shortwebsubject/\shortwebtitle}
1310 \newcommand{\cheadSol}[1]{\def\eq@cheadSol{#1}}
1311 \cheadSol{-- Page \arabic{page} of {\eq@ExamLastPage} --}
```

```

1312 \newcommand{\rheadSol}[1]{\def\eq@rheadSol{#1}}
1313 \rheadSol{SOLUTIONS}

```

The `\runExamHeaderSol` distributes the three solution headers across the page.

```

1314 \newcommand{\runExamHeaderSol}
1315 {\eq@lheadSol\hfill\eq@cheadSol\hfill\eq@rheadSol}

```

The solutions headers are inserted using the `\eq@solutionshook`, see the definition of `\exerSolnInput`.

```

1316 \def\eq@solutionshook
1317 {%
1318   \gdef\eqExamRunHead{\addtolength\textwidth{\oddsidemargin}%
1319   \noindent\hspace*{-\oddsidemargin}\makebox[\textwidth]
1320   {\runExamHeaderSol}}%
1321 }

```

## 13.2 \maketitle definitions

`\maketitle` Standard L<sup>A</sup>T<sub>E</sub>X macro, but this time it is used to create the header at the top of the first page of the exam. Typically, consisting of two rows of info. (1) first row `\maketitledesign` has course name, exam name, and a place for the student to put his/her name. (2) second row has date and instructor. May be redefined.

Modify the title by redefining `\maketitledesign`, the `\maketitle` command itself has L<sup>A</sup>T<sub>E</sub>X commands in it that should not be changed.

```

1322 \newcommand\maketitledesign
1323 {%
1324   \makebox[\textwidth]{\normalsize
1325     \shortstack[l]{\strut\websubject\\\@date}\hfill
1326     \shortstack[c]{\webtitle\\\strut\@altTitle}\hfill
1327     \shortstack[l]{\strut\eq@ExamName\\\webauthor}}%
1328 }
1329 \def\altTitle#1{\def\@altTitle{#1}}
1330 \let\@altTitle\@empty
1331 \def\eqemaketitle
1332 {%

```

`\EQEcalculateAllTotals`: We don't actually calculate all totals, just some of them. We do calculate the grade total of all the `exam` environments in the document, we also calculate the percentage that each exam contributes to to the total. If `\maketitle` is not used, for whatever reason, this command should be calculated explicitly just after `\begin{document}`.

```

1333   \EQEcalculateAllTotals
1334   \begingroup
1335   %   \addtolength\textwidth{\oddsidemargin}%
1336   %   \noindent%\hspace*{-\oddsidemargin}%
1337   \raisebox{.7in}[0pt][0pt]{\SubmitButton}%
1338   \maketitledesign
1339   \endgroup
1340 }

```

If the `fortextbook` option is not taken, we set up the usual `\maketitle` definition.  
 If the `useclassmaketitle` option is taken, we don't redefine either.

```

1341 \ifeqfortextbook
1342   \@ifpackageloaded{web}{\let\maketitle\web@save@maketitle}{}%
1343 \else
1344   \if@equseclassmaketitle
1345     \@ifpackageloaded{web}{\let\maketitle\web@save@maketitle}{}%
1346   \else
1347     \let\maketitle\eqemaketitle
1348   \fi
1349 \fi

```

### 13.3 The cover page definitions

`\eqexcoverpage` The `eqexam` package allows for the possibility of a cover page, if the `coverpage` option is taken.

`\placeCoverPageLogo` A simple command to insert a logo on the cover page. The logo can be used to cover the score in the next page, if the instructor places the score under the logo.  
 Example of usage

```
\placeCoverPageLogo{5in}{-1.5in}{\includegraphics{nwfs_c_logo}}
```

Working from the upper left corner, the first parameter is the amount to move to logo to the right, the second parameter is the amount to move the logo vertically.  
 The Third parameter is the content; perhaps an `\includegraphics` command.

```

1350 \newcommand\placeCoverPageLogo[3]{%
1351   \def\eqe@insertLogo{\hbox to0pt{%
1352     \hspace*{#1}\smash{\raisebox{#2}{#3}}\hss}}
1353 \let\eqe@insertLogo\relax

```

Define `\eqexcoverpage`, this command places `\eqe@insertLogo` and `\eqexcoverpagedesign` in a group. It is this command that gets executed when the user calls for the `coverpage` option.

```

1354 \def\eqexcoverpage{%
1355   \begingroup
1356   \pagenumbering{roman}
1357   \eqe@insertLogo
1358   \eqexcoverpagedesign
1359   \endgroup\newpage
1360   \pagenumbering{arabic}
1361 }

```

`\eqexcoverpagedesign` The `eqexam` package allows for the possibility of a cover page, if the `coverpage` option is taken. This macro can and should be redefined to fit your needs. The definition below is just a representative example.

`\coverpagesubject` The following four commands are used with the cover page. The `\coverpage-subject` is used to provide a special subject for the cover page, different from `\websubject`. The others are used for formatting.  
`\coverpageUniversityFmt`  
`\coverpageSubjectFmt`  
`\coverpageTitleFmt`

```

1362 \newcommand{\coverpagesubject}[1]{\def\coverpage@subject{#1}}
1363 \let\coverpage@subject\@empty
1364 \def\eqexamsubject{\ifx\coverpage@subject\@empty\websubject
1365   \else\coverpage@subject\fi}
1366 \newcommand{\coverpageUniversityFmt}[1]{%
1367   \def\eqex@coverpageUniversityFmt{#1}}
1368 \coverpageUniversityFmt{\bfseries\large}
1369 \newcommand{\coverpageSubjectFmt}[1]{%
1370   \def\eqex@coverpageSubjectFmt{#1}}
1371 \coverpageSubjectFmt{\bfseries\large}
1372 \newcommand{\coverpageTitleFmt}[1]{%
1373   \def\eqex@coverpageTitleFmt{#1}}
1374 \coverpageTitleFmt{\bfseries\large}
1375 \newcommand\cpCID[1]{\def\cp@@CID{#1}}
1376 \let\cp@@CID\@empty
1377 %\newcommand\coverpageCID[1]{\def\cpCID@argi{#1}}
1378 \newcommand{\cpSetCIDWidth}[1]{\bgroup\setlength{\@tempdima}{#1}
1379   \xdef\cpCID@argi{\the\@tempdima}\egroup}
1380 \def\cp@CID{\bgroup\settowidth\eq@tmplengthA{\cp@@CID}%
1381   \@tempdima=\cpCID@argi\relax
1382   \advance\@tempdima-\eq@tmplengthA
1383   \eqe@idinfohl{\makebox[\cpCID@argi][l]{\cp@@CID}}\egroup}
1384 \cpSetCIDWidth{2.25in}
1385 %\coverpageCID{2.25in}

```

When `coverpage` and `coverpagesumry` are used, an **Exam Record** is generated, just a summary may appear beneath the name and ID section of the cover page (`\qeSumryVert`) or to the right (`\eqeSumryHoriz`).

```

\qeSumryVert 1386 \def\eqeSumryVert{\def\eqe@SumryVert{\par\vfill}%
\eqeSumryHoriz 1387   \let\eqe@SumryHoriz\relax
1388   \def\eqe@@SumryVert{\vspace{\stretch{-1}}\bigskip}}
1389 \def\eqeSumryHoriz{\def\eqe@SumryHoriz{\hfill}\let\eqe@SumryVert\relax
1390   \let\eqe@@SumryVert\relax}

```

The default is a horizontal orientation.

```
1391 \eqeSumryHoriz
```

We can name the components of the **Exam Record** by page or by parts. For the case they are named by parts, there are three options: You can use the exam name (the default); you can use the friendly name of the exam (the optional argument), executing `\useUIPartNames` invokes this option; you can use custom names (useful if the friendly names are too long), execute the command `\useCustomPartNames` for this option.

```

\useUIPartNames
\useCustomPartNames

```

```

1392 \def\eqe@coverPageNaming{0}
1393 \newcommand{\useUIPartNames}{\def\eqe@coverPageNaming{1}}
1394 \newcommand{\useCustomPartNames}{\def\eqe@coverPageNaming{2}}

```

When there is custom naming (`\customNaming`), we need to provide the user with a way of defining these custom names. `\customNaming` provides that mechanism. The command takes two arguments, the first is *name*, the exam name, the second



is the associated `text` the text that is to appear in the **Exam Record**

```

1395 \newcommand{\customNaming}[2]{\@namedef{userCustom#1}{#2}}
1396 \newcommand{\cpSetNameAndIDWidth}[1]{\bgroup
1397   \setlength{\@tempdima}{#1}%
1398   \xdef\cpSetNameAndIDWidth{\the\@tempdima}\egroup}
1399 \cpSetNameAndIDWidth{.45\textwidth}
1400 \newcommand{\cpEnclNameAndID}[1]{\def\cpEnclNameAndID{#1}}
1401 \let\cp@EnclNameAndID\empty
1402 \newcommand{\cpSetHghtFrstLn}[1]{\def\cp@HghtFrstLn{#1}}
1403 \def\cp@HghtFrstLn{Opt}
1404 \def\cp@setHghtFrstLn{\rule[\cp@HghtFrstLn]{Opt}{Opt}}
1405 \newcommand\cpNameAndID{\noindent
1406   \edef\eqExamName@argii{\cp@SetNameAndIDWidth}%
1407   \edef\eqSID@argii{\cp@SetNameAndIDWidth}%
1408   \edef\cpCID@argi{\cp@SetNameAndIDWidth}%
1409   \cp@EnclNameAndID{\parbox[b]{\cp@SetNameAndIDWidth}{\parindentOpt
1410     \cp@setHghtFrstLn\eq@ExamName\\[2ex]
1411     \ifx\cp@CID\empty\eq@SID\else\cp@CID\fi\\[2ex]
1412     \ifx\use@email\eq@YES\eq@Email\\[2ex]\fi
1413     \textcolor{\webauthor@color}{\webauthor}, \@date\vskipOpt}}}
```

Now we present the definition of `\eqexcoverpagedesign`.

```

1414 \newcommand{\eqexcoverpagedesign}
1415 {%
1416   \begingroup
1417   \parindentOpt
1418   \thispagestyle{empty}
1419   % \addtolength\textwidth{\oddsidemargin}
1420   \vspace*{.1\textheight}
1421   \noindent%\hspace*{-\oddsidemargin}%
1422   \makebox[\linewidth]{\parbox{\linewidth}%
1423     {\eqex@coverpageUniversityFmt
1424       \color{\webuniversity@color}%
1425       \centering\webuniversity}}
1426   \par\vspace{.1\textheight}
1427   \noindent%\hspace*{-\oddsidemargin}%
1428   \makebox[\linewidth]{\parbox{\linewidth}%
1429     {\eqex@coverpageSubjectFmt
1430       \color{\websubject@color}%
1431       \centering\eqexamssubject}}
1432   \par\vspace{\bigskipamount}
1433   \noindent%\hspace*{-\oddsidemargin}%
1434   \makebox[\linewidth]{\parbox{\linewidth}%
1435     {\eqex@coverpageTitleFmt
1436       \color{\webtitle@color}%
1437       \centering\webtitle}}
1438   \par\vspace{\stretch{1}}
1439   \optionalpagematter
1440   \par\vspace{\stretch{1}}\cpNameAndID
```

```

1441 \eqe@SumryHoriz\eqe@SumryVert
1442 \sumryAnnots
1443 \endgroup
1444 }

```

The following are various local strings used in the **Exam Record**, the default is to use English words.

```

\cpSumryHeader  The header that appears at the top of the box
\cpSumryPts     The formatting for the number of points
\cpSumryPage    The word for “Page”
\cpSumryTotal   The word for “Total”
\cpSumryGrade   The word for “Grade”
\cpSetSumryWidth  The width of the summary box
\cpUsefbox      Enclose in an \fbox? The default is yes
\cpNofbox       Do not enclose in an \fbox

```

```

1445 \newcommand{\cpSumryHeader}{\textbf{Exam Record}}
1446 \newcommand{\cpSumryPts}{\,\text{pts}}
1447 \newcommand{\cpSumryPage}{Page}
1448 \newcommand{\cpSumryTotal}{Total:}
1449 \newcommand{\cpSumryGrade}{Grade:}
1450 \newcommand{\cpSetSumryWidth}[1]{\bgroup\setlength{\@tempdima}{#1}
1451 \xdef\cp@SetSumryWidth{\the\@tempdima}\egroup}
1452 \cpSetSumryWidth{.5\textwidth}
1453 \newcommand{\cpUsefbox}{\let\cp@Usefbox\fbox}
1454 \cpUsefbox
1455 \newcommand{\cpNofbox}{\let\cp@Usefbox\mbox}
1456 \newcommand{\cprulelength}{1.5in}

```

A helper command used in `\cpSumrybyparts`

```

1457 \def\cp@InsertNaming#1{%
1458 \ifcase\eqe@coverPageNaming
1459 #1\or\@nameuse{userFriendly#1}\or
1460 \@nameuse{userCustom#1}\else#1\fi
1461 }

```

```

\cpSumrybyparts  The routine for building the summary box, where we list the statistics for each
part.

```

```

1462 \newcommand{\cpSumrybyparts}{%
1463 \eqe@SumryVert\cp@Usefbox{%
1464 \ifx\cp@Usefbox\fbox
1465 \def\cp@sumryWdth{\cp@SetSumryWidth-2\fboxrule-2\fboxsep}\else
1466 \def\cp@sumryWdth{\cp@SetSumryWidth}\fi
1467 \begin{minipage}[b]{\cp@sumryWdth}\kern0pt
1468 \begin{flushleft}

```

```

1469 \setlength{\eqetmplengtha}{\cprulelength}%
1470 \edef\eqe@cprulelength{\the\eqetmplengtha}%
1471 \csarg\ifx{NumberOfParts}\relax
1472 \else
1473 \count2=0
1474 \medskip
1475 \cpSumryHeader\par\parskip\bigskipamount
1476 \def\##1{\advance\count2by\csname ##1total\endcsname
1477 \underbar{\makebox[0pt][l]{%
1478 \cp@IsertNaming{##1}}\hspace*{\eqe@cprulelength}}}%
1479 / $\eqe@ptsFmt{\csname ##1total\endcsname}%
1480 \cpSumryPts$\par}\thePartNames
1481 \fi
1482 \underbar{\makebox[0pt][l]{\cpSumryTotal}%
1483 \hspace{\eqe@cprulelength}}/
1484 $\eqe@ptsFmt{\the\count2\relax}\cpSumryPts$ \par
1485 \underbar{\makebox[0pt][l]{\cpSumryGrade}%
1486 \hspace{\eqe@cprulelength}}}%
1487 \ifx\cp@Usefbox\fbbox\medskip\fi
1488 \end{flushleft}\kern0pt
1489 \end{minipage}}\par
1490 }

```

`\cpSumrybypages` The routine for building the summary box, where we list the statistics for each page.

```

1491 \newcommand{\cpSumrybypages}{%
1492 \eqe@@SumryVert\cp@Usefbox{%
1493 \ifx\cp@Usefbox\fbbox
1494 \def\cp@sumryWdth{\cp@SetSumryWidth-2\fbboxrule-2\fbboxsep}\else
1495 \def\cp@sumryWdth{\cp@SetSumryWidth}\fi
1496 \begin{minipage}[b]{\cp@sumryWdth}\kern0pt
1497 \begin{flushleft}
1498 \setlength{\eqetmplengtha}{\cprulelength}%
1499 \edef\eqe@cprulelength{\the\eqetmplengtha}%
1500 \csarg\ifx{NumberOfParts}\relax
1501 \else\medskip\count0=0\relax\count2=0\relax
1502 \cpSumryHeader\par\parskip\bigskipamount
1503 \ifsolutionsAtEnd
1504 \ifundefined{eqExamQuesLastPage}{\def\NUMPAGES{0}}
1505 {\edef\NUMPAGES{\@nameuse{eqExamQuesLastPage}}}%
1506 \else
1507 \ifundefined{eqExamLastPage}{\def\NUMPAGES{0}}
1508 {\edef\NUMPAGES{\@nameuse{eqExamLastPage}}}%
1509 \fi
1510 \@whilenum\count0<\NUMPAGES\relax\do{%
1511 \advance\count0by1\relax
1512 \@ifundefined{Page\the\count0total}{\count4=0\relax}
1513 {\count4=\@nameuse{Page\the\count0total}\relax}%
1514 \csarg\ifx{Page\the\count0spilltotal}\relax
1515 \else\advance\count4by\csname%

```

```

1516         Page\the\count0spilltotal\endcsname\relax\fi
1517         \advance\count2by\count4\relax\underbar{%
1518         \makebox[0pt][l]{\cpSumryPage~\the\count0}%
1519         \hspace*{\eqe@cprulelength}}/
1520         $\eqe@ptsFmt{\the\count4\relax}%
1521         \cpSumryPts$\csarg\ifx{Page\the\count0spilltotal}\relax
1522         \else\ $(\@nameuse{Page\the\count0spilltotal}%
1523         \cpSumryPts+\@nameuse{Page\the\count0total}%
1524         \cpSumryPts)$\fi\par
1525     }% while
1526 \fi
1527 \underbar{\makebox[0pt][l]{\cpSumryTotal}%
1528 \hspace{\eqe@cprulelength}}/
1529 $\eqe@ptsFmt{\the\count2\relax}\cpSumryPts$\par
1530 \underbar{\makebox[0pt][l]{\cpSumryGrade}%
1531 \hspace{\eqe@cprulelength}}%
1532 \ifx\cp@Usefbox\fbbox\medskip\fi
1533 \end{flushleft}\kern0pt
1534 \end{minipage}}\par
1535 }

```

If the author takes the `coverpage` option, `\eqex@coverpage` is set equal to `\eqexcoverpagedesign`, otherwise, it is set equal to `\relax`.

```
1536 \AtBeginDocument{\eqex@coverpage}
```

## 13.4 Insert Points in Margins and Compute Page Totals

Here we attempt to place point values of a problem in the margins and to compute the page totals.

`\probvalue` This is a fundamental macro for keeping track of the points of the problem. It increments the counter `eqpointvalue`, which keeps a running total of the points of the current part of the exam, puts the value in the margins, and sets a `mark`, so that at the end of the page, we can compute the number of points on the current page. This macro is used in several situations, for example, in the `problem` environment, `\manualcalcparts` and in `\autocaleparts`. **Should not be redefined.**

Parameters

`#1` = total points for this problem  
`#2` = 0 if total points, otherwise, `#2` is the number of points each problem.

```

1537 \def\@markttotalvalue{\global\let\eqe@innermarkpts\relax
1538 \ifbypasseqexamheading\else\def\eqe@marktxt{%
1539 \theeqpointvalue\@nameuse{eqExam}\theeq@numparts}%
1540 \ifx\endparts\endexercise@parts@tabular
1541 \xdef\eqe@innermarkpts{\noexpand\mark{\eqe@marktxt}}\else
1542 \mark{\eqe@marktxt}\fi\fi}
1543 \def\probvalue#1#2{\addtocounter{eqpointvalue}{#1}%
1544 \ifdisplayPoints\marginpoints{#1}{#2}\fi

```

```

1545      \@markttotalvalue}

\widthtpboxes is the width of the box in the margins that contains the points or totals. The
               design of the box keys off this width.
1546 \newcommand{\widthtpboxes}{35pt}

\marginboxdesign is the basic box that encloses the points on the right, and the totals. This box may
                be redefined as desired, in which case, \measurePtBoxHt should be re-executed.
1547 \newcommand{\marginboxdesign}[2][]{%
1548     \parbox{\widthtpboxes}{\tabcolsep=0pt\relax
1549         \begin{tabular}{|c|}\hline
1550             \vrule height15pt width0pt#1\\\hline
1551             \makebox[\widthtpboxes-2\fbboxrule]{#2}\\\hline
1552         \end{tabular}}%
1553     }%
1554 }
```

`\eqleftmarginbox` This macro places the problem value in the left margin, can be redefined, if you dare.

Parameters

#1 = total points for this problem  
 #2 = 0 if total points, otherwise, #2 is the number of points each problem.

Currently, this macro is not used so I'll make it into verbatim text.

```

\newcommand{\eqleftmarginbox}[2]{\makebox[0pt][r]{%
    \setlength\tabcolsep{0pt}%
    \raisebox{-.5\height}[0pt][0pt]{%
        \marginboxdesign{\marginpointsboxtext{#1}{#2}}%
    }\hspace*{\marginparsep}\hspace*{\eqemargin}}%
}
```

## Formatting the points

`\ptsLabel` The following three convenience commands can be used to localize some of the  
`\eachLabel` strings to other languages.

```

\pointsLabel 1555 \providecommand{\ptsLabel}[1]{\def\eqptsLabel{#1}\ptsLabel{pts}
1556 \providecommand{\ptLabel}[1]{\def\eqptLabel{#1}\ptLabel{pt}
1557 \newcommand{\eachLabel}[1]{\def\eqeachLabel{#1}\eachLabel{ea.}}
```

These next two are used by the `instructions` environment to display the points for the exam.

```

1558 \newcommand{\pointsLabel}[1]{%
1559     \def\eqpointsLabel{#1}\pointsLabel{points}
1560 \newcommand{\pointLabel}[1]{%
1561     \def\eqpointLabel{#1}\pointLabel{point}}
```

### pointsonleft or pointsonboth

`\marginpointtext` Creates the text for `\eqleftmargin` to use. From the macro definition, if #2 is 0, then we write the points for the problem, else, we write the points each for the problem.

`\leftmarginPtsTxt` `\leftmarginPtsEaTxt` `\leftmarginPtsEaTxt` `\leftmarginPtsEaTxt` is the formatting for the total points for the problem; while `\leftmarginPtsEaTxt` is the formatting for the `{problem*}{<num>ea}`-type problems.

`\pointsAsText` We begin by giving the user a choice, typeset the points in math mode (the default) or typeset in the normal font.

```
1562 \def\pointsAsText{\def\eqe@ptsFmt{\text}}
1563 \let\eqe@ptsFmt\relax
1564 \@onlypreamble\pointsAsText
```

`\marginpointtext` determines, based on #2 whether the points are ‘points each’ or not, and calls either `\leftmarginPtsTxt` or `\leftmarginPtsEaTxt`, as appropriate.

```
1565 \newcommand{\marginpointtext}[2]{\ifnum#2=0\leftmarginPtsTxt{#1}\else
1566   \leftmarginPtsEaTxt{#2}\fi
1567 }
```

`\leftmarginPtsTxt` displays points in the left margin: ( $\#1^{\text{Pt}}$ ) or ( $\#1^{\text{pts}}$ ), where #1 is the number of points for this problem.

```
1568 \newcommand{\leftmarginPtsTxt}[1]{(\small$\eqe@ptsFmt{#1}^{\text{\%
1569   \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi}}$)}
```

`\leftmarginPtsEaTxt` displays ‘points each’ in the left margin: ( $\#1^{\text{Pt}}$ ) or ( $\#1^{\text{pts}}$ ), where #1 is the number of points for each part of this problem.

```
1570 \newcommand{\leftmarginPtsEaTxt}[1]{(\small$\eqe@ptsFmt{#1}_{\text{\%
1571   \eqeachLabel}}^{\text{\ifnum#1=1\relax\eqptLabel\else
1572   \eqptsLabel\fi}}$)}
```

`\eqleftmargin` Places the number of points (or points each) in the left margin. Can be redefined as desired. This macro is used when author chooses the `pointsonleft` or `pointsonbothsides` option. The text for the points is defined above, `\marginpointtext`.

```
1573 \newcommand{\eqleftmargin}[2]{\makebox[0pt][r]{\marginpointtext{#1}{#2}%
1574   \setlength{\@tempdima}{\marginparsep+\eqemargin}%
1575   \hspace*{\@tempdima}}}
```

### pointsonright or pointsonboth

`\marginpointsboxtext` When the author selects `pointsonright` or `pointsonbothsides`, a box appears in the right margin containing problem totals, this is the text for the box.

```
1576 \newcommand{\marginpointsboxtext}[2]{\small$\eqe@ptsFmt{#1}\,\text{\%
1577   \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi}}$}
```

`\eqrightmarginbox` When the author selects `pointsonright` or `pointsonbothsides`, a box appears in the right margin containing problem totals, this is the box that appears.

`\insertPointsBoxPDF` These two commands are `\let`ing them be either a form field (that the document author can fill in during online grading, or as `\relax`. It depends on whether the `email` option is taken or not.

```

1578 \def\@insertPointsBoxPDF
1579 {
1580     \def\fieldName{pointsgiven.\curr@quiz.page\thepage}%
1581     \thequestionnoi}%
1582     \calcTextField[\F\FHidden\BC{ }\Q1\textColor{1 0 0 rg}]
1583     {\fieldName}{\widthtpboxes}{15pt}%
1584 }
1585 \def\@insertTotalsBoxPDF
1586 {
1587     \def\fieldName{pagetotals.\curr@quiz.page\thepage}%
1588     \calcTextField[\F\FHidden\BC{ }\Q1\textColor{1 0 0 rg}]
1589     \AA{\AACalculate{AFSimple_Calculate("SUM",
1590         new Array("pointsgiven.\curr@quiz.page\thepage"));}}]
1591     {\fieldName}{\widthtpboxes}{15pt}%
1592 }
1593 \ifx\use@email y
1594     \let\insertPointsBoxPDF\@insertPointsBoxPDF
1595     \let\insertTotalsBoxPDF\@insertTotalsBoxPDF
1596 \else
1597     \let\insertPointsBoxPDF\@empty
1598     \let\insertTotalsBoxPDF\@empty
1599 \fi

```

The points box that appears in the left margin, may be re-defined, if you dare. At the bottom of the box goes the points for the problem, if the `email` option is used, in the top, a text field is inserted.

```

1600 \newcommand{\eqrightmarginbox}[2]{\makebox[0pt][l]{%
1601     \setlength\tabcolsep{0pt}%
1602     (2011/05/08) new calculation for the right point box.
1603     \setlength{\@tempdima}{\textwidth-\eqmargin+\marginparsep}%
1604     \hspace*{\@tempdima}%
1605     \raisebox{-.5\height}[0pt][0pt]{%
1606         \marginboxdesign[\insertPointsBoxPDF]%
1607         {\marginpointstext{#1}{#2}}}%
1608     }\hfil}%
1609 }

```

At the end of each exam `\end{exam}` the part totals can be optionally displayed on the right. This is the box for doing that.

```

1610 \newcommand{\eqeomarginboxright}[2]{\makebox[0pt][r]{%
1611     \setlength\tabcolsep{0pt}%
1612     \setlength{\@tempdima}{\textwidth+\marginparsep-\parindent}%
1613     \hspace*{\@tempdima}%
1614     \raisebox{-.5\height}[0pt][0pt]{%
1615         \marginboxdesign[\insertTotalsBoxPDF]%

```

```

1616             {\marginpointsboxtext{#1}{#2}}%
1617         }\hfil
1618     }%
1619 }

```

At the end of each exam `\end{exam}` the part totals can be optionally displayed on the left. This is the box for doing that.

```

1620 \newcommand{\eqgeomarginboxleft}[2]{\noindent\makebox[0pt][r]{%
1621     \raisebox{-.5\height}[0pt][0pt]{%
1622         \marginboxdesign[\insertTotalsBoxPDF]%
1623         {\marginpointsboxtext{#1}{#2}}%
1624     }\settowidth{\@tempdimb}{\eq@hspannerPrb}%
1625     \setlength{\@tempdima}{\marginparsep+\eqemargin-\@tempdimb}%
1626     \hspace*{\@tempdima}}\hfil
1627 }

```

`\measurePtBoxHt` We measure the height of our point/total boxes and store it in the length `\eq@pointboxtotalheight`, which is used to separate the boxes so they don't overlap.

```

1628 \newcommand{\measurePtBoxHt}{\setbox\eq@pointbox\hbox{%
1629     \marginboxdesign{\marginpointsboxtext{00}{0}}%
1630 }\setlength\eq@pointboxtotalheight{.5\ht\eq@pointbox}%
1631 \xdef\halfHtPtBox{\the\eq@pointboxtotalheight}%
1632 \setlength\eq@pointboxtotalheight{\dp\eq@pointbox+\ht\eq@pointbox}%
1633 \xdef\totHtPtBox{\the\eq@pointboxtotalheight}%
1634 \xdef\dpPtBox{\the\dp\eq@pointbox}%
1635 \xdef\htPtBox{\the\ht\eq@pointbox}%
1636 }
1637 \measurePtBoxHt

```

A helper command to set both margin boxes.

```

1638 \newcommand{\eqbothmargins}[2]{\eqleftmargin{#1}{#2}%
1639     \eqrightmarginbox{#1}{#2}}

```

The macro `\probvalue`, defined above, says

```
\mark{\theeqpointvalue\csname eqExam\endcsname\theeq@numparts}
```

so the `\botmark` contains the point value of this page and the part number for this page.

```
1640 \def\lastparttotaled{0}
```

The `\botmark` looks like `18\eqExam0`, where 18 will be the total number of points accumulated for this exam part. We need to retrieve these number, I don't remember why I save them this way.

`\parsetotals` The `\parsetotals` macro is called in `\settotalsbox` to retrieve the ongoing point values from `\botmark`, and returns two parameters, contained in `\argi` and `\stripeqExam` `\argiii`. Whereas, `\stripeqExam` just retrieves the first parameter only.

```

1641 \def\parsetotals#1\eqExam#2\end{\def\argi{#1}\def\argii{#2}}
1642 \def\stripeqExam#1\eqExam{\def\argii{#1}}

```



`\settotsbox` This is the box containing the page total, it may appear on the left or right side bottom corner.

```

1643 \def\settotsbox{%
1644     \expandafter\parsetotals\botmark\eqExam\end
1645     \ifx\argi\@empty\hfil
1646     \else\ifx\argii\@empty\hfil
1647         \else
1648             \expandafter\stripeqExam\argii

```

Set `eqpointsthispage` equal to `\argi`, which should be the accumulated total for this part of the test so far.

```

1649         \setcounter{eqpointsthispage}{\argi}%

```

This subtract `eqpointsofar`, which should be the total for this test part through the previous page. The difference is the number of points for this page.

```

1650         \addtocounter{eqpointsthispage}{-\value{eqpointsofar}}%

```

Now we set `eqpointsofar` to the new accumulated total `\argi`.

```

1651         \setcounter{eqpointsofar}{\argi}%

```

And save to the auxiliary file for later usage.

```

1652         \xdef\lastparttotaled{\argii}%
1653         \eqe@IW0\@auxout{\string\csarg\string
1654             \gdef{Page\thepage total}%
1655             {\theeqpointsthispage}}%

```

Finally, place the totals box.

```

1656         \totalsbox
1657     \fi
1658 \fi
1659 }

```

`\totalsboxtext` This is the text that appears in the totals boxes

```

1660 \newcommand\totalsboxtext{\small$
1661     \eqe@ptsFmt{\theeqpointsthispage}\,\text{%
1662     \ifnum\theeqpointsthispage=1\relax\eqptLabel\else
1663     \eqptsLabel\fi}$}

```

`\eqevtranstotbox` `\eqevtranstotbox` is a length that can be used to raise or lower the position of the total boxes in the left/right bottom corner. The default is `0pt`.

```

1664 \newlength\eqevtranstotbox
1665 \setlength{\eqevtranstotbox}{0pt}

```

`\totalsboxleft` There are two totals boxes, one for the left and one for the right side. These can

`\totalsboxright` be redefined as desired.

```

1666 \def\eqe@chkZeroTotals{\let\eqe@allow\eqe@YES
1667     \ifx\eqe@zeroTotalsAllowed\eqe@NO
1668     \ifnum\theeqpointsthispage=0 \let\eqe@allow\eqe@NO\fi\fi}
1669 \newcommand{\totalsboxleft}{\eqe@chkZeroTotals
1670     \ifx\eqe@allow\eqe@YES
1671     \makebox[0pt][r]{\setlength\tabcolsep{0pt}%
1672         \raisebox{-\height+\eqevtranstotbox}[0pt][0pt]{%

```

```

1673         \marginboxdesign[\insertTotalsBoxPDF]{\totalsboxtext}%
1674     }\hspace*{\marginparsep}%
1675 }\fi\hfil
1676 }
1677 \newcommand{\totalsboxright}{\eqe@chkZeroTotals
1678     \ifx\eqe@allow\eqe@YES
1679     \makebox[Opt][l]{\setlength\tabcolsep{0pt}%
1680         \hspace*{\textwidth}\hspace*{\marginparsep}%
1681         \raisebox{-\height+\eqevtranstotbox}[Opt][Opt]{%
1682             \marginboxdesign[\insertTotalsBoxPDF]{\totalsboxtext}%
1683         }%
1684     }\fi\hfil
1685 }

```

We begin the document by declaring the current part is part 0 with 0 points. This is the default, in case the author writes a document with no parts or points!

```

1686 \ifeqe@nopoints\else\if@bypasseqexamheading\else
1687 \AtBeginDocument{\mark{0\csname eqExam\endcsname0}}
1688 \fi\fi

```

At the end of the document, we write out the number of parts for this test, and the names of the parts the author has given each part.

```

1689 \AtEndDocument{%
1690     \eqe@IWO\@auxout{\string\csarg\string\gdef{NumberOfParts}%
1691         {\arabic{eq@numparts}}}%
1692     \eqe@IWO\@auxout{\string\csarg\string\gdef{thePartNames}%
1693         {\the\partNames}}%
1694     \clearpage\addtocounter{page}{-1}\writelastpage\relax
1695     \addtocounter{page}{1}%
1696 }

```

**\theGrandTotal** When an exam has multiple parts, the total of each part is computed and the grand total is computed with **\theGrandTotal**.

```

1697 \newcommand\theGrandTotal{%
1698     $\eqe@ptsFmt{\csname eqeGrandTotal\endcsname}$}

```

**\totalForPart** User access to the total for a exam environment, one argument, the name of the exam.

**\percentForPart** Attempts to calculate the percent of the total that the referenced exam (#1) contributes to the grand total.

```

1699 \newcommand{\totalForPart}[1]{\eqe@ptsFmt{\csname#1total\endcsname}$}
1700 \def\tot@lForPart#1{\csname#1total\endcsname}
1701 \newcommand{\percentForPart}[1]{\eqe@ptsFmt
1702     {\csname#1percent\endcsname}$\%}

```

### 13.5 Computing Number of Points within a Segment of the Exam

We add some commands for calculating number of points in a segment of the exam. A segment being defined subset of consecutive problems within an exam.

We can define segments by placing markers (using `\placeMarkerHere`) between problems. We can then calculate the total number of points between markers.

The counter below is a scratch counter for making the calculations. We assume the `calc` package is loaded, it does the work for us.

```
1703 \newcounter{markerCnt}
```

`\placeMarkerHere` Placed outside of any `problem/problem*` environment, this command takes one argument, the symbolic name for this marker. We write to the auxiliary file and define a command whose name is based on `#1`, and whose value is the cumulative total `\theeqpointvalue`.

```
1704 \def\placeMarkerHere#1{%
1705     \eq@IW0\@auxout{\string\csarg\string\gdef
1706         {#1SaveTotalHere}{\theeqpointvalue}}%
1707 }
```

`\calcFromMarkers` Once the markers are in place, we can calculate the number of points defined between two such markers. The `\calcFromMarkers` takes three arguments, `#2` and `#3` are the symbolic names of the two markers. While, `#1` is an optional argument for formatting the calculation. The default is `\@markerTotalFmt`, defined below in `\markerTotalFmt`. The value of the total is `\themarkerCnt`.

```
1708 \newcommand{\calcFromMarkers}[3][\@markerTotalFmt]{%
1709     \ifundefined{#2SaveTotalHere}{}%
1710     {\ifundefined{#3SaveTotalHere}{\setcounter{markerCnt}%
1711     {\@nameuse{#2SaveTotalHere}-\@nameuse{#3SaveTotalHere}}%
1712     \ifnum\value{markerCnt}<0\relax
1713     \setcounter{markerCnt}{-\value{markerCnt}}\fi#1}}%
1714 }
```

`\markerTotalFmt` Used to set the global format of the marker totals. The value of the marker total is `\themarkerCnt`. The default follows the definition of `\markerTotalFmt`

```
1715 \newcommand{\markerTotalFmt}[1]{\def\@markerTotalFmt{#1}}
1716 \markerTotalFmt{\space\themarkerCnt\space points}
```

`\calcQsBtwnMarkers` `\calcQsBtwnMarkers[Mrk2]{Mrk1}` does a number of things; primarily, it determines the range of the questions between the two marks. The names of the

`\Mrk1Start` commands produced are all based in the first marker name `Mrk1`. `\Mrk1Start` is the first question number that follows the the placement of `\calcQsBtwnMarkers`.

`\Mrk1End` Similarly, `\Mrk1End` is the last question number between the two marks `Mrk1` and `Mrk2`.

`\Mrk1nQs` We also calculate `\Mrk1nQs`, the number of questions appearing between `Mrk1` and `Mrk2`.

```
1717 \newcommand{\calcQsBtwnMarkers}[2][\setcounter{markerCnt}{0}%
1718     \def\eq@argi{#1}%
1719     \Mrk1thisQnum is the question number in effect at the point where the command
1720     \calcQsBtwnMarkers is inserted.
1719     \eq@IW0\@auxout{\string\csarg\string\gdef
1720         {#2thisQnum}{\theeqquestionnoi}}%
```

$\backslash Mrk1Start$  is 0 if undefined.

```
1721 \ifundefined{#2thisQnum}{\csarg\xdef{#2Start}{0}}%
1722 \eqe@IW0\@auxout{\string\csarg\string\gdef
1723 {#2Start}{0}}}%
1724 }{%
```

$\backslash Mrk1Start$  is the current question number plus 1.

```
1725 \setcounter{markerCnt}{\value{eqquestionnoi}+1}\csarg
1726 \xdef{#2Start}{\the\value{markerCnt}}}%
1727 \eqe@IW0\@auxout{\string\csarg\string\gdef
1728 {#2Start}{\@nameuse{#2Start}}}%
1729 }
```

Using now both  $Mrk1$  and  $Mrk2$  we calculate the difference in the two. Result held in  $markerCnt$ . If there is no optional parameter,  $Mrk2$ , do not produce the  $\backslash Mrk1thisQnum$ .

```
1730 \ifx\eqe@argi\@empty\else
1731 \ifundefined{#2thisQnum}{\setcounter{markerCnt}{0}}}%
1732 {\ifundefined{#1thisQnum}{\setcounter{markerCnt}{0}}}{%
1733 \setcounter{markerCnt}%
1734 {\@nameuse{#1thisQnum}-\@nameuse{#2thisQnum}}}%
1735 }}%
1736 \fi
```

$\backslash Mrk1nQs$  is the number of questions that appear between  $Mrk1$  and  $Mrk2$ .

```
1737 \csarg\xdef{#2nQs}{\the\value{markerCnt}}}%
1738 \eqe@IW0\@auxout{\string\csarg\string\gdef
1739 {#2nQs}{\the\value{markerCnt}}}%

```

$\backslash Mrk1End$  is 0 if undefined. If there is no optional parameter,  $Mrk2$ , do not produce the  $\backslash Mrk1End$

```
1740 \ifx\eqe@argi\@empty\else
1741 \ifundefined{#1thisQnum}{\csarg\xdef{#2End}{0}}%
1742 \eqe@IW0\@auxout{\string
1743 \csarg\string\gdef{#2End}{0}}}%
1744 }{%
```

$\backslash Mrk1End$  is  $\backslash Mrk2thisQnum$ .

```
1745 \csarg\xdef{#2End}{\@nameuse{#1thisQnum}}}%
1746 \eqe@IW0\@auxout{\string\csarg\string\gdef
1747 {#2End}{\@nameuse{#2End}}}%
1748 }%
1749 \fi
1750 }
```

$\backslash markStartFor$  We provide a user interface to the three macros defined above. The required parameter is a mark (a name that was used as the first argument of  $\backslash markEndtFor$   $\backslash calcQsBtwnMarkers$ ).

```
1751 \newcommand{\markStartFor}[1]{\@nameuse{#1Start}}
1752 \newcommand{\markEndtFor}[1]{\@nameuse{#1End}}
1753 \newcommand{\markNumQsFor}[1]{\@nameuse{#1nQs}}
```

## 13.6 Useful Commands to Write Problems

### 13.6.1 The `\placeAtxy` command

`\placeAtxy` Use to place material within the solutions area that is visible to the student on the test document. Syntax:

`\placeAtxy{x_dim}{y_dim}{content}`

`\placeAtxy` goes immediately after `\end{solution}`

```
1754 \newcommand{\placeAtxy}[3]{%
1755     \par\nointerlineskip
1756     \ifdisplayworkarea
1757         \ifx\eq@insertverticalspace\eqe@YES
1758             \makebox[0pt][l]{\hspace*{-\parindent}\hspace*{#1}%
1759                 \raisebox{#2}[0pt][0pt]{#3}}\fi
1760     \fi
1761 }
```

### 13.6.2 The `workarea` environment

As of this writing, 2012/12/09, this segment of code has been re-written. The `splitsolution` and `panel` environments have a new syntax. I've never gotten any feedback from user who use these environments, so I suspect the impact of these changes are minimal.

`workarea` A work area is an environment used when we display vertical space such as when  
`\workareasb` the `nosolutions` and `vspacewithsolns` options are in effect. `\workareasb` is a save box used in the argument of `lrbox` to save the contents of the `minipage`.

`\begin{workarea}[width]{depth}`

...

`\end{workarea}`

```
1762 \newsavebox{\workareasb}
```

`\workareaVadj{<skip>}` adjusts the vertical skip of the first line that appears in the `workarea` environment.

```
1763 \def\workareaVadj#1{\bgroup\def\eqe@rgi{#1}\ifx\eqe@rgi\@empty
1764     \def\eqe@rgi{0pt}\fi\setlength{\@tempdima}{\eqe@rgi}%
1765     \xdef\workareaV@dj{\the\@tempdima}\egroup
1766 } % dps1
1767 \workareaVadj{3pt}
1768 \def\workareaCmds#1{\def\@rgi{#1}\ifx\@rgi\@empty
1769     \let\work@reaCmds\relax\else
1770     \def\work@reaCmds{#1}}
1771 \let\work@reaCmds\relax
1772 \def\priorworkareaCmds#1{\def\@rgi{#1}\ifx\@rgi\@empty % dpsj6
1773     \let\priorWorkAreaCmds\relax\else
1774     \def\priorWorkAreaCmds{#1}\fi}
1775 \let\priorWorkAreaCmds\relax
```

```

1776 \newenvironment{workarea}[2][\linewidth]
1777 {%
1778     \removeatlastparskip
1779     \edef\workareadepth{\if\currhideopt HOpt\else#2\fi}%
1780     \begin{lrbox}{\workareasb}\setlength{\eqetmplengtha}{#1}%

```

A change in support of `\fillTypeGrid`. When `\ifforceEqualCells` is true, we adjust the width of this minipage to conform to the shorter line produced by the grid.

```

1781     \ifx\makeVgrid\relax\else % dps2
1782         \ifforceEqualCells
1783         \ifx\fillLinesLineWidth\@empty\else
1784             \setlength{\eqetmplengtha}{\fillLinesLineWidth}\fi
1785         \fi
1786     \fi
1787     \begin{minipage}[b][\workareadepth][t]{\eqetmplengtha}%

```

Apply vertical skip `\workareaV@dj` here, set by `\workareaVadj`.

```

1788     \vglue\workareaV@dj
Adjust \leftskip when the problem is a lead-in.
1789     \@ifundefined{leadinIndentPrtSep}{}{\ifx\solutionparshape\@empty
1790         \else\leftskip\leadinIndentPrtSep\fi}\work@reaCmds
1791 }{%
1792     \end{minipage}\end{lrbox}%

```

4/5/11 Changed the logic here, so the `workarea` is available for the `vspacewith-solns` option.

```

1793     \ifeq@solutionsafter\else
1794     \par\ifdisplayworkarea
1795         \ifx\eq@insertverticalspace\eqe@YES
1796         \removeatlastparskip
1797         \vglue-\baselineskip
1798         \if\currhideopt H\else
1799             \noindent\strut\smash{\usebox{\workareasb}}%
1800         \fi\fi
1801     \fi\fi
1802 }

```

### 13.6.3 The splitsolution environment

**splitsolution** This is a special solution environment designed for use with the `online` or `email` options, but it is compatible with all other options as well.

### New Syntax

```
\begin{splitsolution}[width][depth]
\begin{panel}[l|r]
...
\end{panel}
\begin{solution}
...
\end{solution}
\end{splitsolution}
```

### Old Syntax

```
\begin{splitsolution}{depth}
\begin{panel}[l|r]{width}
...
\end{panel}
\begin{solution}
...
\end{solution}
\end{splitsolution}
```

The `panel` writes its contents to a CUT file, then inputs it back in to get its dimensions, it places the contents in the box `\eqpanelbox`.

Below is an example of the use of the `splitsolution` environment (which is defined well below here).

```
\begin{problem}[5]
This is a question worth $5$ points.
\begin{splitsolution}
\begin{panel}\relax
\includegraphics[scale=.2]{fig1}
\end{panel}
\begin{solution}
This a really good solution and worthy of a $5$ points.
\end{solution}
\end{splitsolution}
\end{problem}
```

Here, we include a graphic. We put it into a box, `\eqpanelbox` to get its dimensions. The `splitsolution` environment has one required argument, the total height of the solution. We then place the graphic in the `panel` environment, and write the solution in the usual way.

`splitsolution` We arrive at the `splitsolution` environment.

```
1803 \def\splitsolution{%
1804     \@ifnextchar[{\splitsolutioni}%
1805         {\splitsolutioniii{\panelwidth}{\panelheight}}}%
1806 }
1807 \def\splitsolutioni[#1]{%
1808     \@ifnextchar[{\splitsolutionii{#1}}
1809         {\splitsolutioniii{\panelwidth}{#1}}}%
1810 }
1811 \def\splitsolutionii#1[#2]{%
1812     \splitsolutioniii{#1}{#2}%
1813 }
1814 \def\splitsolutioniii#1#2{%
1815     \@ifundefined{panel@write}{\newwrite\panel@write}{}%
1816     \gdef\ss@Argiii{#1}\gdef\ss@Argii{#2}%

```

Here, we `\let \solution` to `\ssssolution` and replace it with `\@ssSolution`

```
1817     \let\@ssssolution\solution
1818     \let\end@ssssolution\endsolution
```

```

1819 \let\solution\@ssSolution
1820 \let\endsolution\end@ssSolution
1821 \par\noindent\ignorespaces

```

The `panel` environment should appear next, just after the `\begin{splitssolution}`

```

1822 }
1823 \def\endsplitsolution{%
1824 \ifeq@solutionsafter\solutionsafterSkip\fi % dpsj10
1825 \ifx\solutionparshape\@empty\else
1826 \advance\linewidth-\leadinIndentPrtSep\fi %%dps
1827 \edef\eqe@workwidth{\if\ss@Argi 1\noexpand\ss@Argiii
1828 \else\noexpand\linewidth\fi}%
1829 \edef\eqe@workfill{\if\ss@Argi 1\else\hfill\fi}%
1830 \ifeq@solutionsafter\else
1831 \begin{workarea}[\eqe@workwidth]{\eqe@depth}%
1832 \let\panelGetDimen\relax%
1833 \eqe@workfill\input{panel\thepanel@cnt.cut}%
1834 \end{workarea}
1835 \fi\ifeq@solutionsafter\else\solutionsafterSkip\fi
1836 }

```

`\panelgap` `\panelgap` is the separation between solution area and the panel. is a box to  
`\eqpanelbox` put things into, to measure their dimensions. The user accessible commands  
`\panelwidth` `\panelwidth` returns the width and `\panelheight` the height. There is a special  
`\panelheight` counter `panel@cnt` to keep track of the panels generated.

```

panel@cnt 1837 \newcommand\panelgap{3pt}
1838 \newsavebox{\eqpanelbox}
1839 \newdimen\eqpanelwidth
1840 \newdimen\eqpanelheight

```

`\getDimSSPanel` The `\getDimSSPanel` is an internal command. It places it argument in the box  
`\eqpanelbox` and gets its dimensions. It save the dimensions as `\eqpanelwidth`  
and `\eqpanelheight` (the total height).

```

1841 \newcommand{\getDimSSPanel}[1]{%
1842 \sbox{\eqpanelbox}{#1}%
1843 \eqpanelwidth=wd\eqpanelbox
1844 \eqpanelheight=ht\eqpanelbox
1845 \advance\eqpanelheight by\the\dp\eqpanelbox
1846 }

```

`\panelwidth` `\panelwidth` and `\panelheight` are aliases for the dimension registers. This is  
`\panelheight` for compatibility with previous versions.

```

1847 \def\panelwidth{\eqpanelwidth}
1848 \def\panelheight{\eqpanelheight}
1849 \newcounter{panel@cnt}

```

`panel` environment goes inside the `splitssolution` environment. Permissible values of  
the first parameter are `r` and `l`, the default being `l`. This is the alignment param-  
eter. The second argument is the width of the panel.

```

1850 \def\panel{\def\eqe@panelir{\eqe@paneli[r]}}%

```



```

1851 \ifnextchar[{\eqe@paneli}
1852     {\ifnextchar\relax{\expandafter\eqe@panelir\@gobble}%
1853      {\eqe@paneli[r]}}}%
1854 }

```

We continue the panel environment.

```

1855 \def\eqe@paneli[#1]{%
1856     \gdef\ss@Argi{#1}\def\eqe@l{l}\def\eqe@r{r}%
1857     \ifx\ss@Argi\empty\gdef\ss@Argi{r}\else
1858     \ifx\eqe@l\ss@Argi\else\ifx\eqe@r\ss@Argi\else
1859     \PackageError{eqexam}{Permissible arguments for panel are l and r}
1860     {Use l or r for the argument of panel.}\fi\fi\fi
1861     \stepcounter{panel@cnt}%
1862     \immediate\openout \panel@write \panel\thepanel@cnt.cut

```

Manage the h, H, and global overrides.

```

1863 \if\currhideopt H%
1864     \eqe@IW0\panel@write{\vfill}%
1865     \immediate\closeout\panel@write
1866     \gdef\ss@Argii{0pt}%
1867 \else
1868     \if\currhideopt h\ifeq@solutionsafter\else
1869     \ifeq@globalshowsolutions\else
1870     \eqe@IW0\panel@write{\vfill}%
1871     \immediate\closeout\panel@write
1872     \gdef\ss@Argii{0pt}%
1873     \fi\fi
1874 \fi
1875 \fi
1876 \begingroup

```

Write the contents of this environment to the file `panel\thepanel@cnt.cut`.

```

1877 \let\verbatim@out\panel@write

```

We begin by writing `\panelGetDimen{%` to the CUT file

```

1878 {\lccode'C='\% \lccode'P='\{
1879     \lowercase{\eqe@IW0\verbatim@out{\string\panelGetDimen PC}}}%

```

Followed by the verbatim listing of the panel environment.

```

1880 \verbatimwrite
1881 }
1882 \def\endpanel{%
1883     \endverbatimwrite

```

After the verbatim write, we write `}` to close off the argument of `\panelGetDimen`.

```

1884 {\lccode'P='\} \lowercase{\eqe@IW0\verbatim@out{P}}}%
1885 \immediate\closeout\panel@write
1886 \endgroup

```

The command `\eqePANEL cut` does all the work at the end of this environment. We first `\let \panelGetDimen to \getDimenSSPanel` and input the CUT file. This loads the panel contents into `\eqpanelbox`, and gets their dimensions. We

set the value of `\ss@Argiii` which is the width parameter of the `splitsolution` environment. `\mp@Width` calculates the width.

```

1887 \gdef\eqePANELCUT{%
1888     \let\panelGetDimen\getDimSSPanel
1889     \input{panel\thepanel@cnt.cut}%
1890     \setlength{\eq@tmpdima}{\ss@Argiii}%
1891     \xdef\ss@Argiii{\the\eq@tmpdima}%
1892     \xdef\mp@Width{\ifeq@solutionsafter\linewidth
1893         \else\linewidth-\ss@Argiii-\panelgap\fi}%
1894 } \aftergroup\eqePANELCUT
1895 }

```

`\eqe@IW` is a macro that either writes verbatim text (#1), or it gobbles the argument, depending on whether have `solutionsafter`, `nosolutions`, `hidden` or not `hidden`.

```

1896 \long\def\eqe@IW#1{%
1897     \ifeq@solutionsafter
1898         \let\eqe@next\@empty
1899     \else
1900         \def\eqe@next{\eqe@IW0\verbatim@out{#1}}%
1901         \ifeq@nosolutions\else
1902             \if\currhideopt H%
1903                 \gdef\ss@Argii{Opt}\let\eqe@next\@empty
1904             \else
1905                 \if\currhideopt h%
1906                     \ifeq@globalshowssolutions\else
1907                         \gdef\ss@Argii{Opt}\let\eqe@next\@empty
1908                     \fi
1909                 \fi
1910             \fi
1911         \fi
1912     \fi
1913     \eqe@next
1914 }

```

The `splitsolution` environment `\lets` the solution environment to the internal `@ssSolution` environment. It performs various tasks then turns it over to the old solution environment, which has been `\let` to `\@sssolution`.

```

1915 \newenvironment{@ssSolution}[1][\ss@Argii]%
1916 {%
1917     \def\eqe@argi{#1}\ifx\eqe@argi\@empty\else
1918         \let\soln@keys@nLines\@empty
1919         \edef\panelheight{\the\panelheight}%
1920         \edef\panelwidth{\the\panelwidth}%
1921         \edef\temp@exp{\noexpand
1922             \setkeys*{soln@keys}{#1}}\temp@exp
1923         \ifx\soln@keys@nLines\@empty
1924             \edef\ss@Argii{\XKV@rm}%
1925         \else
1926             \@tempdima\w1Vspace\relax

```

```

1927      \@tempdima=\soln@keys@nLines\@tempdima
1928      \edef\ss@Argii{\the\@tempdima}%
1929      \ifx\XKV@rm\@empty\else
1930          \if@equsedim\edef\ss@Argii{\XKV@rm}\fi\fi
1931      \fi
1932      \setlength{\eq@tmpdima}{\ss@Argii}%
1933 %      \ifx\sameVspace\@empty\gdef\sameVspace{0pt}\fi

    If the panelheight is greater than the requested height, we make the requested
    height equal to panelheight.
1934      \ifdim\eqepanelheight>\eq@tmpdima
1935          \eq@tmpdima\eqepanelheight\fi
1936      \xdef\ss@Argii{\the\eq@tmpdima}%
1937      \fi

    We calculate the depth of the solutions; this is generally the values passed to use
    as the optional argument, and encapsuled by \ss@Argii.
1938      \xdef\eqedepth{%
1939          \ifvspacewithsolns\ss@Argii\else
1940              \ifeq@nosolutions\ss@Argii\else
1941                  \ifeq@solutionsafter\ss@Argii\else0pt\fi\fi\fi}%

    The width is either \linewidth or \linewidth-\ss@Argiii-\panelgap, where
    \ss@Argiii is the width of the panel.
1942      \xdef\mp@Width{\ifeq@solutionsafter\noexpand\linewidth\else
1943          \noexpand\linewidth-\ss@Argiii-\panelgap\fi}%

    If its a left panel...
1944      \if\ss@Argi 1%
1945          \ifeq@solutionsafter
1946              \def\eqe@lPanel{\parbox[b][\eqedepth][t]{\ss@Argiii}

    Here, and elsewhere, we \let \panelGetDimen to \relax when we want to purely
    input the CUT file in the solution.
1947              {\let\panelGetDimen\relax
1948                  \input{panel\thepanel@cnt.cut}\vfill}\hfill}%
1949              \def\eqe@rPanel{\@empty}%
1950          \else

    For the left panel, we need to subtract \eqemargin
1951          \def\eqe@subleftgutter{-\string\eqemargin}%
1952          \def\eqe@lPanel{%
1953              \string\parbox[b][\string\eqedepth]{\ss@Argiii}%
1954              {\string\let\string\panelGetDimen\string\relax
1955              \string\input{panel\thepanel@cnt.cut}%
1956              \string\vfill}\string\hfill^^J}%
1957          \def\eqe@rPanel{\@empty}%
1958      \fi
1959      \else

    If its a right panel...
1960      \ifeq@solutionsafter

```

```

1961         \def\eqe@lPanel{\@empty}%
1962         \def\eqe@rPanel{%
1963             \hfill\parbox[b][\eqedepth][t]{\ss@Argiii}
1964             {\let\panelGetDimen\relax
1965             \hfill\input{panel\thepanel@cnt.cut}\vfill}}%
1966     \else
    For the right panel, we do not to subtract \eqemargin as above.
1967         \let\eqe@subleftgutter\@empty
1968         \def\eqe@lPanel{\@empty}%
1969         \def\eqe@rPanel{%
1970             \string\hfill\string\parbox[b][\string\eqedepth]%
1971             {\ss@Argiii}{\string\endgraf%
1972             \string\noindent\string\hfill%
1973             \string\let\string\panelGetDimen\string\relax
1974             \string\input{panel\thepanel@cnt.cut}%
1975             \string\vfill}}%
1976     \fi
1977 \fi
1978 \let\verbatim@out\ex@solns
1979 \par\ifeq@solutionsafter\smallskip\fi

    Since everything is put a box, we set the \linewidth, and set \solutionparshape
    to \@empty.
1980 \ifx\solutionparshape\@empty\else
1981     \advance\linewidth-\leadinIndentPrtSep\fi %%dps
1982 \let\solutionparshape\@empty
1983 \noindent\minipage{\linewidth}%
1984 \if\ss@Argi 1\noindent\parbox[b][\eqedepth][t]{\ss@Argiii}%
1985     {\vfill}\hfill\fi
1986 \minipage[b][\eqedepth][t]{\mp@Width}%
1987 {\lccode'C='\% \lowercase{\eqe@IW{%
1988     \string\def\string\panelgap{\panelgap}%
1989     \string\setlength{\string\panelwidth}{\ss@Argiii}%
1990     \string\def\string\eqedepth
1991     {\ifeq@nosolutions\ss@Argii\else\ss@Argii\fi}C}}}%
1992 {\lccode'C='\% \lowercase{\eqe@IW{%
1993     \string\strut\string\noindent%
1994 %         \string\strut\par\string\nobreak\string\noindent%
1995         \string\hbox\space to\string\linewidth\bgroup^^J%
1996         \eqe@lPanel\string\minipage[b][\string\eqedepth][t]%
1997         {\string\linewidth\eqe@subleftgutter%
1998         -\string\panelwidth-\string\panelgap}C
1999     }}}%
2000 \ifeq@solutionsafter
2001     \noindent\strut\hbox to\linewidth\bgroup
2002     \eqe@lPanel
2003     \minipage[b][\eqedepth][t]{\linewidth-\ss@Argiii-\panelgap}%
2004     \def\solutionsafterSkip{}%
2005 \fi

```

Here, we start the original definition of the solutions environment, which was `\let` to `\@sssolution`.

```
2006      \ifx\soln@keys@nLines\@empty
2007      \edef\passedss@Argii{\ss@Argii}\else
2008      \edef\passedss@Argii{\ss@Argii,nLines=\soln@keys@nLines}\fi
```

For a split solution, cannot allow align left

```
2009      \if\ss@Argi l\fillerLinesAlignDef\fi % dpsj10
```

I cannot get the filler lines to work when the answerkey is in effect. The next few lines locally turns off filler lines for this problem.

```
2010      \ifanswerkey % dpsj10
2011      \gdef\p@ssToFLs{\let\eq@insertverticalspace\eqe@NO
2012      \@eqlinedfillerfalse}\fi % dpsj10
2013      \eqe@flexextendedfalse
2014      \expandafter\@sssolution\expandafter[\passedss@Argii]%
2015 }{%
2016      \eqe@IW{\string\endminipage\eqe@rPanel\egroup}%
2017      \ifeq@solutionsafter\endminipage\eqe@rPanel\egroup\fi
2018      \end@sssolution
2019      \endminipage\endminipage
2020 }
2021 % Redefine the \cs{paragraph} command
2022 %      \begin{macrocode}
2023 \renewcommand{\paragraph}
2024      {\@startsection{paragraph}{4}{0pt}{12pt}{-3pt}{\bfseries}}
```

`\defaultInstructions` For the instructions environment, defined next, the default string for the instructions is `\eq@default@Instructions`, this command is defined using `\defaultInstructions`.

```
2025 \newcommand{\defaultInstructions}[1]{\def\eq@default@Instructions{#1}}
2026 \defaultInstructions{Instructions.}
```

`instructions` Each test, or a part of a test usually have instructions. This instruction environment is used in this purpose. Normally, the number of points for the part is displayed following the heading (the default is **Instructions.**). You can eliminate the total points from the instructions by taking the `nosummarytotals` option.

```
2027 \newcommand{\beforeInstrSkip}{1ex}
2028 \newcommand{\afterInstrSkip}{-0em}
2029 \newcommand{\hInstrSpace}{\ }
2030 \newcommand{\styleInstr}{\bfseries}
2031 \newenvironment{instructions}[1][\eq@default@Instructions]{%
    If \summaryPointTotal is zero, show no summary points.
2032      \@ifundefined{\thisexamlabel total}{\let\eq@nosummarytotals\eqe@YES}
2033      {\ifnum\summaryPointTotal=0 \let\eq@nosummarytotals\eqe@YES\fi}%
2034      \expandafter\def\expandafter\eq@argi\expandafter{#1}%
2035      \def\hsi{\hInstrSpace}%
```

We start a paragraph environment

```

2036 \startsection{paragraph}{4}{\z@}%
2037 {\beforeInstrSkip\space\@plus1ex \@minus.2ex}%
2038 {\afterInstrSkip}{\normalfont\normalsize\styleInstr}*%
2039 {\textcolor{\@instructionsColor}{#1}\normalcolor%
2040 \ifx\eq@nosummarytotals\eqe@YES\else\ifx\eq@argi\@empty\else\hsi\fi
2041 {\normalfont\summaryTotalsTxt}\hsi\fi}\hskip-\lastskip
2042 \ifx\eq@nosummarytotals\eqe@YES\ifx\eq@argi\@empty\else\hsi\fi\fi
2043 \normalfont\normalsize\ignorespaces
2044 }{\par\vskip\beforeInstrSkip\relax}

\summaryPointTotal \summaryPointTotal contains the total points for the current exam; the com-
\summaryTotalsTxt mand \summaryTotalsTxt defines its the formatting for the points. The command
\nQuesInExam \nQuesInExam is the total number of questions in the exam.

2045 \newcommand\summaryPointTotal{\csname\thisexamlabel total\endcsname}
2046 \newcommand\summaryTotalsTxt{(\$ \eqe@ptsFmt{\summaryPointTotal}\, \text{%
2047 \@ifundefined{\thisexamlabel total}{\eqpointsLabel}%
2048 {\ifnum\summaryPointTotal=1\relax\eqpointLabel\else
2049 \eqpointsLabel\fi}}\$)}
2050 \newcommand{\nQuesInExam}[1][\thisexamlabel]{%
2051 \def\eqe@argi{#1}%
2052 \ifx\thisexamlabel\@empty\ifx\eqe@argi\thisexamlabel
2053 \PackageError{eqexam}{The optional argument for
2054 \string\nQuesInExam\MessageBreak must be specified}{\fi\fi
2055 \csname#1nQuestions\endcsname}

```

Make this definition for `hyperref`, so its anchors will be unique. Useful when there are multiple parts of the test.

```

2056 \ifx\hyper@anchor\@undefined\else
2057 \renewcommand\theHeqquestionnoi
2058 {\curr@quiz.\theeqquestionnoi\eqe@fpmrk}
2059 \renewcommand\theHquizno{%
2060 \if\probstar*\curr@quiz.%
2061 \theeqquestionnoi.part\thepartno.\arabic{quizno}%
2062 \else
2063 \curr@quiz.%
2064 \theeqquestionnoi.\arabic{quizno}%
2065 \fi
2066 }
2067 \renewcommand\theHpartno{\curr@quiz.%
2068 \theeqquestionnoi.part\thepartno}
2069 \fi

```

`eqComments` Often, I want to make additional instructions between problems, you can use this comment environment.

```

2070 \newcommand{\beforeCommentSkip}{1.25ex}
2071 \newcommand{\afterCommentSkip}{-1ex}
2072 \newcommand\hCommSpace{\ }
2073 \newcommand\styleComm{\bfseries}

```

```

2074 \newenvironment{eqComments}[1][\strut]{\removelastskip
2075     \def\eqe@cargi{#1}\def\eqe@Strut{\strut}\def\hsc{\styleComm}%
2076     \@startsection{paragraph}{4}{\z@}%
2077     {\beforeCommentSkip\space\@plus1ex \@minus.2ex}%
2078     {\afterCommentSkip}{\normalfont\normalsize\styleComm}*%
2079     {\textcolor{\@eqCommentsColor}{#1}}%
2080     \ifx\eqe@cargi\eqe@Strut\hskip\afterCommentSkip
2081     \else\ifx\eqe@cargi\@empty\hskip\afterCommentSkip\else
2082     \hsc\strut\fi\fi\normalfont\normalsize
2083     \color{\@eqCommentsColorBody}\ignorespaces
2084 }{\par\vskip\beforeCommentSkip\space\@plus1ex \@minus.2ex}

```

### 13.7 The exam Environment

Each part of the exam is enclosed in an `exam` environment. The environment is a customized version of the `shortquiz` environment.

`\exambegdef` Some definitions that are executed at the beginning of each exam environment.

```

2085 \let\tb@beginexam@code\relax
2086 \def\partialspillovertotals{0}
2087 \def\exambegdef
2088 {%
2089     \csname\thisexamlabel pagemark\endcsname
2090     \@ifundefined{partialtotalpg}{\}%
2091     \ifnum\partialtotalpg=\arabic{page}%
2092         {\count0=\partialspillovertotals
2093          \advance\count0by\partialtotaleoe
2094          \xdef\partialspillovertotals{\the\count0}%
2095          \eqe@IW0\@auxout{\string\csarg\string\gdef
2096           {Page\partialtotalpg spilltotal}{\partialtotaleoe}}%
2097         }%
2098     \fi
2099 }%
2100 \csarg\ifx{\thisexamlabel pageno}\relax
2101 \else
2102     \csarg\ifx{pagenofirstprob\thisexamlabel}\relax
2103     \else
2104         \csarg\ifnum{\thisexamlabel pageno}
2105             < \@nameuse{pagenofirstprob\thisexamlabel}%
2106         \else
2107             \csarg\ifx{\thisexamlabel pagemark}\relax
2108             \else
2109                 \ifnum\value{page}=
2110                     \csname\thisexamlabel pageno\endcsname
2111                     \eqe@IW0\@auxout{\string\csarg\string\gdef
2112                      {\thisexamlabel pagemark}{\string\newpage}}%
2113             \fi
2114         \fi
2115     \fi
2116 \fi

```

```

2117 \fi
2118 \setcounter{eqquestionnoi}{0}\setcounter{eqpointvalue}{0}%
2119 \setcounter{eqpointsofar}{0}\setcounter{eqpointsthispage}{0}%
2120 \setcounter{eq@count}{0}%

We wrote \begin{eqquestions} to the top of the solutions file (\jobname.sol.

2121 \writeBeginEqeQuestions
2122 \label{\thisexamlabel PageBegin}%
2123 \eqe@IWO\@auxout{\string\csarg\string\gdef
2124 {\thisexamlabel pageno}{\thepage}}%
2125 % \let\sq@priorhook\@empty
2126 \ifeqfortextbook
2127 \global\examenvtrue\tb@beginexam@code
2128 \fi
2129 }

```

**\examenddef** Some definitions that are executed at the end of each exam environment. We place a totals box to report the total since the last page.

```

2130 \def\tb@insmargmark{\ifisinstred\ifismarginans
2131 \insMidMarg{\mark{}}}\fi\fi}
2132 \def\eqe@adjForSepRule{\if\eq@parttotals\eqe@YES
2133 \eq@pointboxtotalheight\halfHtPtBox\relax
2134 \advance\eq@pointboxtotalheight\dpPtBox\relax
2135 \advance\eq@pointboxtotalheight\pointsmarginparpush
2136 \vspace*{\eq@pointboxtotalheight}\fi}
2137 \def\eqe@afterexamsepcode{%
2138 \if\eq@parttotals\eqe@YES
2139 \@actionsAtPageBreak{\global\let\@spacetobreak1}%
2140 {\global\let\@spacetobreak0}%
2141 \ifx\@spacetobreak0\relax
2142 \bgroup\@tempdima=\pagetotal
2143 \advance\@tempdima\eq@pointboxtotalheight
2144 \ifdim\@tempdima>\pagegoal\aftergroup\newpage
2145 \else\ifnum\arabic{eq@count}>0\relax
2146 \removelastskip\vskip6pt\kern0pt
2147 \if\@reportpoints\eqe@One\else\@checkSpacing{0}\fi
2148 \textcolor{\endexamtotal@color}%
2149 {\eqeomarginbox{\arabic{eq@count}}{0}}\fi
2150 \ifx\eqx@separationrule\eqe@YES
2151 \separationrule\eqe@adjForSepRule\fi
2152 \fi\egroup
2153 \fi
2154 \else\ifx\eqx@separationrule\eqe@YES
2155 \separationrule\eqe@adjForSepRule\fi
2156 \fi
2157 }
2158 \def\examenddef
2159 {%
2160 \global\let\partialtotaleoe\relax
2161 \global\let\partialtotalpg\relax

```



```

2162 \global\let\afterexamsepcode\relax
2163 \csarg\ifx{NumberOfParts}\relax
2164 \else
2165 \ifnum\value{eq@numparts}<\NumberOfParts
2166 \setcounter{eq@count}{\value{eqpointvalue}}%
2167 \addtocounter{eq@count}{-\value{eqpointsofar}}%
2168 \xdef\partialtotaletoe{\arabic{eq@count}}%
2169 \xdef\partialtotalpg{\arabic{page}}%

```

See if there is enough room at the bottom of the page to place the end of exam totals and to start a new exam, if any. If not, forget it, and start a new page.

```

2170 \global\let\afterexamsepcode\eqe@afterexamsepcode
2171 \fi
2172 \fi
2173 \ifeqfortextbook\global\examenvfalse\fi

```

(2011/05/08) Just before the file is closed and input, we write the end of the `eqquestions` environment, `\end{eqquestions}`.

```

2174 \writeEndEqeQuestions
2175 \writetotalstoaux
2176 \addtocounter{page}{-1}%
2177 \writelastpage[\thisexamlabel]\addtocounter{page}{1}%
2178 \ifeqfortextbook\tb@insmargmark\fi
2179 }
2180 \def\@actionsAtPageBreak#1#2{%
2181 \bgroup\@tempdima\pagegoal\advance\@tempdima-\pagetotal
2182 \@tempdimb\@fvsizeskip\vsiz
2183 \ifdim\@tempdima < \@tempdimb #1\else #2\fi\egroup
2184 }

```

`\separationrule` For an exam with multiple parts, a separation rule is created, unless absorbed into a page break. The command `\separationrule` defines this separation rule, it can be redefined as desired.

```

2185 \newcommand{\separationrule}{\makebox[\linewidth-\eqemargin][c]%
2186 {\rule{.67\linewidth}{.4pt}}}

```

(2011/05/08) This is a new environment that makes an exam into a list of problems. This is an attempt to expand the use of `eqexam` to L<sup>A</sup>T<sub>E</sub>X documents. We give control over the page layout so an `eqexam` document can be used within a textbook.

```

2187 \def\eqe@hspannerPrb{\ }
2188 \ifundefined{ifwithinsoldoc}{\newif\ifwithinsoldoc\withinsoldocfalse}{\ }
2189 \newcommand{\eqquestopsep}[1]{\def\eqques@topsep{#1}}
2190 \providecommand{\eqquesparsep}[1]{\def\eqques@parsep{#1}}
2191 \newcommand{\eqquesitemsep}[1]{\def\eqques@itemsep{#1}}
2192 \newcommand{\eqqueslistparindent}[1]{\def\eqques@listparindent{#1}}
2193 \eqquestopsep{0pt}
2194 \eqquesparsep{0pt}
2195 \eqquesitemsep{0pt}

```

```

2196 \eqqueslistparindent{0pt}
2197 \newif\iffirstitem
2198 \def\eqeSolnItemMngt{\iffirstitem\global\firstitemfalse
2199   \let\eqe@next\@empty\else\let\eqe@next\item\fi\eqe@next\relax}
2200 \newenvironment{eqequestions}{%
2201   \begin{list}{\}%
2202   \ifwithinsoldoc\let\solnItemMngt\eqeSolnItemMngt\fi
2203   \setlength{\labelwidth}{\eqemargin}%
2204   \setlength{\parsep}{\eqeques@parsep}%
2205   \setlength{\itemsep}{\eqeques@itemsep}%
2206   \setlength{\topsep}{\eqeques@topsep}%
2207   \setlength{\itemindent}{0pt}%
2208   \setlength{\listparindent}{\eqeques@listparindent}%dps%
2209   \ifwithinsoldoc\settowidth{\labelsep}{\eqe@hspannerSoln}\else
2210   \settowidth{\labelsep}{\eqe@hspannerPrb}\fi
2211   \setlength{\leftmargin}{\labelwidth}%
2212   }\ifwithinsoldoc\global\firstitemtrue\fi\item\relax}{\end{list}}

```

**exam** Each part of the exam is enclosed in an **exam** environment. The one required parameter is the name of the part, for example, ‘Part1’, ‘Part2’. These should be one word, no white spaces, just letters and possibly numbers.

```

2213 \def\setDefaultfvssizeskip#1{\def\default@fvssizeskip{#1}%
2214   \def\@fvssizeskip{#1}}
2215 \def\default@fvssizeskip{.3}
2216 \edef\@fvssizeskip{\default@fvssizeskip}
2217 \newcommand{\fvssizeskip}[1]{\def\@fvssizeskip{#1}}
2218 \def\autoExamName{exam\the\value{eq@numparts}}
2219 \def\nNumberOfPrts{\csname NumberOfParts\endcsname}
2220 \abovesqskip{}
2221 \let\eqeWrtExamTitleToSolns\eqe@YES
2222 \def\wrtExamTitleInSolns{\let\eqeWrtExamTitleToSolns\eqe@YES}
2223 \def\noExamTitleInSolns{\let\eqeWrtExamTitleToSolns\eqe@NO}
2224 \let\thisexamlabel\@empty
2225 \let\isInExamEnv\eqe@NO

```

**\RecordThisExamOff** causes eqexam to place **\iffalse** and **\fi** around the solutions to this exam in the solution file. This makes the solution results of this exam invisible. **\RecordThisExamOff** sets an internal switch **\ifDoNotRecordThisExam** to **true**. This switch is set back to **false** at the end of the environment.

```

2226 \newif \ifDoNotRecordThisExam \DoNotRecordThisExamfalse % dpsf02
2227 \def\RecordThisExamOff{\DoNotRecordThisExamtrue} % dpsf02
2228 \def\bIFFalseWrtSolns{\writeT@SolnFile{\protect\iffalse^^J}}
2229 \def\eIFFalseWrtSolns{\writeT@SolnFile{\protect\fi^^J}}

```

The beginning of the **exam** environment.

```

2230 \newenvironment{exam}[2] []
2231 {%
2232   \ifDoNotRecordThisExam % dpsf02
2233   \expandafter\bIFFalseWrtSolns\fi
2234   \makeRoomForProb{\@fvssizeskip\textheight}{0}%

```

```

2235 \let\isInExamEnv\eqe@YES
2236 \stepcounter{eq@numparts}%
    If #2 is empty, use \autoExamName
2237 \def\eqexamargii{#2}\ifx\eqexamargii@empty
2238 \edef\eqexamargii{\autoExamName}\fi
2239 \xdef\thisexamlabel{\eqexamargii}\xdef\curr@quiz{\eqexamargii}%
2240 \def\eqexamargi{#1}\ifx\eqexamargi@empty
2241 \edef\eqexamargi{\eqexamargii}\fi
2242 \expandafter\gdef\expandafter\thisUFexamlabel
2243 \expandafter{\eqexamargi}%
2244 \edef\eq@tmp{\the\partNames\string\\{\eqexamargii}}%
2245 \global\partNames=\expandafter{\eq@tmp}%
2246 \csarg\ifx{NumberOfParts}\relax\else
2247 \ifx\eqeWrtExamTitleToSolns\eqe@YES
2248 \ifnum\nNumberOfP@rts=1\relax\else
2249 \def\eqe@argi{#1}\ifx\eqe@argi@empty
2250 \eqe@writetoSolns{\eqexamargii}\eqe@writetoAux{\string
2251 \csarg\string\gdef{userFriendly\eqexamargii}{\eqexamargii}}
2252 \else
2253 \eqe@writetoSolns{#1}\eqe@writetoAux{\string
2254 \csarg\string\gdef{userFriendly#2}{#1}}
2255 \fi
2256 \fi\fi
2257 \fi
2258 \exambegdef
2259 \edef\temp@Exp{\noexpand\shortquiz\sqstar[\eqexamargii]\temp@Exp
2260 % \expandafter\shortquiz\sqstar[\eqexamargii]%
2261 }{%
2262 \examenddef

2263 \vskip\eqeques@parsep\relax\kernOpt %dps88
2264 \endshortquiz
2265 \aftergroup\afterexamsepcode
2266 \par\penalty-100\vskipOpt
2267 \ifDoNotRecordThisExam % dpsf02
2268 \expandafter\eIFFalseWrtSolns\fi
2269 \global\DoNotRecordThisExamfalse
2270 }

```

**\EQEcalculateAllTotals** (4/22/11) Added the command \EQEcalculateAllTotals. The command is executed as part of the \maketitle command. If \maketitle is not used for some reason \EQEcalculateAllTotals can be executed just after \begin{document}.

```

2271 \newcommand{\EQEcalculateAllTotals}{%
2272 \beginingroup

```

We calculate the grand total of all the parts of the exam environments, and we define \eqeGrandTotal, which contains the total.

```

2273 \count\z@=0\relax
2274 \def\##1{\csarg\ifx{##1total}\relax\else
2275 \advance\count\z@\csname##1total\endcsname

```

`\thePartNames` list all named exam environments in the document, e.g.,

```
\\{Part1}\\{Part2}...\\{LastPart}
```

```
2276      \fi}\csname thePartNames\endcsname
2277      \xdef\eqeGrandTotal{\the\count\z@}%
2278      \ifnum\eqeGrandTotal=0 \else
```

If there is a nonzero grandtotal, we move on to calculate the percentages.

```
2279      \def\\##1{\eqe@calc@percent{##1}}%
2280      \csname thePartNames\endcsname\fi
2281  \endgroup
2282 }
```

`\eqe@calc@percent` We go through the parts listed in `\thePartNames` and create a calculation of the percentage for that part, and leave it in `\csname#1percent\endcsname`, which can be accessed through the `\percentForPart` command, for example `\percentForPart{<part_name>}` might expand to 45.6%.

`\nPctDecPts` The number of decimal points to carry in the representation of the percentage.

```
2283 \newcommand{\nPctDecPts}{1}
2284 \def\eqe@calc@percent#1{\@ifundefined{#1total}{%
2285     \csarg\gdef{#1percent}{??}}{%
```

If the `fp` package is not loaded, we use register arithmetic, percentages are truncated to integers.

```
2286     \csarg\ifx{FPdiv}\relax
2287     \count2=\tot@lForPart{#1}%
2288     \edef\expGT{\csname eqeGrandTotal\endcsname}%
2289     \multiply\count2by100\relax\divide\count2by\expGT\relax
2290     \csarg\xdef{#1percent}{\the\count2 } \else
```

If the `fp` package is loaded, we use this package to calculate the percentage, accurate to one decimal place.

```
2291     \FPdiv{\eqe@pForPart}{\csname#1total\endcsname}%
2292     {\csname eqeGrandTotal\endcsname}%
2293     \FPMul{\eqe@pForPart}{\eqe@pForPart}{100}%
2294     \FPrnd{\eqe@pForPart}{\eqe@pForPart}{\nPctDecPts}%
2295     \csarg\xdef{#1percent}{\eqe@pForPart}\fi
2296     }%
2297 }
```

```
2298 \def\writetotalstoaux{\eqe@IW0\@auxout{\string
2299     \csarg\string\gdef{\thisexamlabel total}{\theeqpointvalue}}%
2300     \eqe@IW0\@auxout{\string\csarg
2301     \string\gdef{\thisexamlabel nQuestions}{\theeqquestionnoi}}%
2302 }
2303 \newcommand{\writelastpage}[1][ ]{\def\eqe@argi{#1}%
2304     \ifx\eqe@argi@empty\else\label{#1PageEnd}\fi
2305     \eqe@IW0\@auxout{\string\csarg
2306     \string\gdef{eqExamLastPage}{\arabic{page}}}%
}
```

```

2307 }
2308 \def\exlabel{}
2309 \def\sqlabel{}
2310 \def\exsolafter{\textit{Solution}:}
2311 \def\sqsolafter{\textit{Solution}:}

The exercise labels in the body of the text. (2015/02/27) changed the command
\exlabelformatwp to incorporate other parameters.
2312 %\def\exlabelformat{\textbf{\thequestionnoi.\ }}
2313 \def\exlabelformat{\textbf{%
2314   \thequestionnoi\eq@decPointPrb\eq@hspannerPrb}}
2315 \def\exlabelformatwp{\exlabelformat}

The exercise labels for solutions at the end of the document
2316 \def\exsllabelformat
2317   {\string\makebox[Opt][r]{\string\textbf{%
2318     \thequestionnoi\eq@decPointPrb\eq@hspannerSoln}}}
2319 \def\exsllabelformatwp
2320   {\string\makebox[Opt][r]{\string\textbf{%
2321     \thequestionnoi\eq@decPointPrb\eq@hspannerSoln}}}%
2322   (\thepartno)\eq@hspannerSoln}
2323 \ifanswerkey
2324   \def\exrtnlabelformat{}
2325   \def\exrtnlabelformatwp{}
2326   \def\eq@sqslrtnlabel{}
2327 \else
2328   \def\exrtnlabelformat{$\square$}
2329   \def\exrtnlabelformatwp{$\square$}
2330   \def\eq@sqslrtnlabel{$\square$}
2331 \fi
2332 \def\sqslrtnlabel{\eq@sqslrtnlabel}

(2010/08/21) Enable some localizations of strings
2333 \newcommand{\exsectitletext}{Solutions to \webtitle}
2334 \def\exsectitle{\normalsize\exsectitletext}
2335 %\def\exsectitle{\normalsize\hspace*
2336 %  {-\oddsidemargin}\exsectitletext}
2337 \@ifpackageloaded{exerquiz}{\%{Solutions to \websubject}
2338   \renewcommand{\exsecrunhead}{}}{\newcommand{\exsecrunhead}{}}
2339 %\providecommand{\exsecrunhead}{Solutions to \websubject}%
2340 \def\eq@sqslsectitle{}
2341 \def\eq@sqslsecrunhead{}
2342 \def\eq@sqsllabel{\string\llap{\string\textbf{\thequestionnoi.\ }}}
2343 \def\eq@sqlabel{}
2344 \let\include@quizsolutions\relax
2345 \let\solnhspace\empty

```

## 13.8 problem Environments

A single question is posed with the `problem` environment, and a question with multiple parts with the `problem*` environment.

`\fillin` This macro is used for fill-in type questions. The first argument is the length of the underline blank to leave to fill-in, the second argument is the correct answer.

```

2346 \newcommand{\optsFillIn}[1]{\def\eqe@optsFillIn{#1}}
2347 \let\eqe@optsFillIn@empty
2348 \newcommand{\fillin}[3][u]{%
2349     \ifx#1u\let\@fillinFmt\underbar
2350     \else\ifx#1b\let\@fillinFmt\relax
2351     \else\let\@fillinFmt\relax\fi\fi
2352     \ifeq@proofing
2353         \@fillinFmt{\makebox[#2]{%
2354             \strut\hfil\bfseries\color{red}#3\hfil}}%
2355     \else
2356         \@fillinFmt{\makebox[#2]{\strut\hfil}}%
2357         \ifundefined{@quiz}{}{%
2358             \if\eq@online\eqe@YES\relax
2359                 \ifeq@nosolutions
2360                     \ifeq@solutionsafter\else
2361                         \ifx\eq@insertverticalspace\eqe@YES\relax
2362                             \stepcounter{@cntfillin}%
2363                             \edef\fieldName{%
2364                                 \if\probstar*eqexam.\curr@quiz.fillin.%
2365                                 \theeqquestionnoi.part\thepartno.%
2366                                 fi\the@cntfillin%
2367                             \else
2368                                 eqexam.\curr@quiz.fillin.%
2369                                 \theeqquestionnoi.fi\the@cntfillin%
2370                             \fi
2371                             }\makebox[0pt][r]{\textField[\BC{}
2372                                 \presets{\eqe@optsFillIn}}{%
2373                                 \fieldName}{#2}{11bp}}%
2374                             \fi
2375                         \fi
2376                     \fi
2377                 \fi
2378             }%
2379         \fi\space\ignorespaces}

```

`\TF` A specialized version of `\fillin` for True/False questions.

```

2380 \newcommand\defaultTFwidth{30pt}
2381 \newcommand\TF[2][\defaultTFwidth]{%
2382     \def\eqe@next{\fillin{#1}{#2}}%
2383     \ifdim\eq@extralabelsep=0pt\relax\else
2384         \if\probstar*\relax\if\exerwparts@cols0
2385             \def\eqe@next{\makebox[0pt][r]{%
2386                 \fillin{#1}{#2}}\ignorespaces}%
2387         \fi\fi\fi
2388 \eqe@next}
2389 \def\fillinWidth#1{%
2390     \if\probstar*

```

```

2391      \settowidth{\eq@tmplengthA}{\normalfont\ }%
2392      \addtolength{\eq@tmplengthA}{#1}%
2393      \edef\eq@extralabelsep{\the\eq@tmplengthA}%
2394      \fi
2395  }
2396  \let\fillInFormatDefault\@empty

```

\Do<num> The following commands supports the optional argument \Do<num>. When I teach senior or graduate-level classes, I often give a problem with multiple parts (each of equal value) and ask them to “do 3 of the following 5” parts.

```

2397 \def\eqe@DoWarning#1{\PackageWarning{eqexam}{\string#1\space
2398   requires the first argument\MessageBreak of problem* to be <num>ea}}
2399 \def\eqe@DoNum{\textbf{??}\eqe@DoWarning{\DoNum}}
2400 \def\eqe@nDoNum{\textbf{??}\eqe@DoWarning{\nDoNum}}
2401 \let\DoNum\eqe@DoNum
2402 \let\nDoNum\eqe@nDoNum
2403 \def\makeDoNum#1{\xdef\nDoNum{#1}%
2404   \def\ifc@sewrap{\ifcase#1??\or}%
2405   \xdef\DoNum{\expandafter\ifc@sewrap\eqe@wordNums\else
2406     \eqe@wordNumsError\fi}}
2407 \def\makeOutOfNum#1{\xdef\nOutOfNum{#1}%
2408   \def\ifc@sewrap{\ifcase#1??\or}%
2409   \xdef\OutOfNum{\expandafter\ifc@sewrap\eqe@wordNums\else
2410     \eqe@wordNumsError\fi}}
2411 \def\eqe@OutOfNum{\textbf{??}\eqe@DoWarning{\OutOfNum}}
2412 \def\eqe@nOutOfNum{\textbf{??}\eqe@DoWarning{\nOutOfNum}}
2413 \let\OutOfNum\eqe@OutOfNum
2414 \let\nOutOfNum\eqe@nOutOfNum

```

\eqe@wordNums is used to typeset the English word for the numbers (1–10). This command may be redefined to other languages.

```

2415 \newcommand{\eqe@wordNums}{one\or two\or three\or
2416   four\or five\or six\or seven\or eight\or nine\or ten}
2417 \newcommand{\eqe@wordNumsError}{\noexpand\PackageError{eqexam}%
2418   {Number out of range, 1--10}%
2419   {Use a smaller number, or redefine the command
2420     \string\eqe@wordNums.}}

```

Added two hooks \priorPNPAction and \postPNPAction to \makeRoomForProb to allow some defined actions before and after a page break generated by \makeRoomForProb. . Examples of usage are found in \vspaceFillerLines and \eq@linesXPgs.

```

2421 \let\priorPNPAction\relax % dps1
2422 \let\postPNPAction\relax
2423 \def\eqe@mkRoomPgBrk{\priorPNPAction\newpage\postPNPAction}
2424 \def\makeRoomForProb#1#2{\par %\endgraf % dps 11/11/10
2425   \bgroup\@nobreakfalse\addpenalty{-500}%
2426   \setlength{\@tempdimb}{#1}%
2427   \@tempdima \pagegoal \advance \@tempdima -\pagetotal

```

```

2428 \ifdim \@tempdima<\@tempdimb\ifnum\col@number>\@ne\columnbreak
2429 \else\aftergroup\eqe@mkRoomPgBrk\fi\fi\egroup
2430 \ifnum\@reportpoints>1
2431 \ifx\eqe@prevProbZero\eqe@YES
2432 \if\eqe@isPts0\else\vskip-\halfHtPtBox\relax\fi
2433 \else
2434 \ifl@stDispl@yPoints
2435 \if\eqe@isPts0\@checkSpacing{1}\else
2436 \if#21 \@checkSpacing{0}\fi
2437 \fi\fi
2438 \fi
2439 \fi
2440 }

```

`\emitMessageNearBottom` The syntax for this command is

```
\emitMessageNearBottom*[vspace]{msg}
```

If there is less than `vspace` remaining on the page, a message, `msg`, is emitted. Then the optional `*` appears, a `\newpage` is also emitted just after the `msg` and `\insertContAnnot` is expanded just after `\newpage`.

```

2441 \newcommand{\emitMessageNearBottom}{\@ifstar
2442 \def\eqe@emnb{\ifnum\col@number>\@ne\columnbreak
2443 \else\newpage\fi\insertContAnnot}\eqe@emitMessageNearBottom}
2444 {\let\eqe@emnb\relax\eqe@emitMessageNearBottom}}
2445 \newcommand{\eqe@emitMessageNearBottom}[2][2\w1Vspace]{%
2446 \eqe@emitMessageNearBottom{#1}{#2}}
2447 \def\eqe@emitMessageNearBottom#1#2{\par
2448 \bgroup\@nobreakfalse\addpenalty{-500}%
2449 \setlength{\@tempdimb}{#1}%
2450 \@tempdima \pagegoal \advance \@tempdima -\pagetotal
2451 \ifdim \@tempdima<\@tempdimb #2 \eqe@emnb
2452 \xdef\eqe@currProbStartPage{0}\fi\egroup} % dps4

```

`\getSpaceLeftOnPage` `\getSpaceLeftOnPage` calculates the amount of space left on the current page. It  
`\amtSpaceLeftOnPage` saves the calculation in the text macro `\amtSpaceLeftOnPage`.

```

2453 \newcommand{\getSpaceLeftOnPage}{\par\bgroup
2454 \@tempdima \pagegoal \advance \@tempdima -\pagetotal
2455 \xdef\amtSpaceLeftOnPage{\the\@tempdima}\egroup}

```

`\promoteNewPage` A simple variation on `\makeRoomForProb` designed for user use.

```

2456 \newcommand{\pnpDflt}{\@fvsize\skip\textheight}
2457 \newcommand{\promoteNewPage}[1][\pnpDflt]{%
2458 \makeRoomForProb{#1}{0}}

```

`\pointsmarginparpush` Used in `\makeRoomForProb` and elsewhere. `\pointsmarginparpush` adds a little more separation between point boxes. `\@checkSpacing` is the algorithm for adjusting the vertical spaces between problems when points are on the right.

```
2459 \def\pointsmarginparpush{3pt}
```



```

2460 \def\@checkSpacing#1{\bgroup\ifinner\else
2461   \@tempdima\lastPageTotal
2462   \@tempdimb\pagetotal
    If \lastPageTotal is less than \pagetotal, continue
2463   \ifdim\@tempdima < \@tempdimb
    Compute \pagetotal - \lastPageTotal
2464     \advance\@tempdimb by-\@tempdima
2465     \ifdim\@tempdimb < \eq@pointboxtotalheight
2466       \if#11%
2467         \eq@pointboxtotalheight\halfHtPtBox\relax
2468         \advance\eq@pointboxtotalheight\dpPtBox\relax
2469       \fi
2470       \@tempdima=\eq@pointboxtotalheight
2471       \advance\@tempdima\pointsmarginparpush\relax
2472       \advance\@tempdima by-\@tempdimb
2473       \vspace*{\@tempdima}%
2474     \fi
2475   \fi\fi
2476 \egroup}
2477 \def\@checkSpacingi{\bgroup
2478   \@tempdima = \lastPageTotal
2479   \@tempdimb = \pagetotal
2480   \ifdim\@tempdima < \@tempdimb
2481     \advance\@tempdimb by-\@tempdima
2482     \eq@pointboxtotalheight\halfHtPtBox\relax
2483     \advance\eq@pointboxtotalheight\dpPtBox\relax
2484     \ifdim\@tempdimb < \eq@pointboxtotalheight
2485       \@tempdima=\eq@pointboxtotalheight
2486       \advance\@tempdima\pointsmarginparpush\relax
2487       \advance\@tempdima by-\@tempdimb
2488       \vspace*{\@tempdima}%
2489     \fi
2490   \fi
2491 \egroup}

```

`\default@nbaselineskip` is the default number of `\baselineskip`s needed to place a new problem. While `\nbaselineskip` `\nbaselineskip` is the number of `\baselineskip`s needed for a new problem (or problem\*).

```

2492 \newcommand{\setDefaultnbaselineskip}[1]{\def\default@nbaselineskip{#1}}
2493 \setDefaultnbaselineskip{6}
2494 \def\nbaselineskip#1{\def\@nbaselineskip{#1}}
2495 \def\nbaselineskipReset{\edef\@nbaselineskip{\default@nbaselineskip}}
2496 \nbaselineskipReset

```

**problem** The `problem` is used to pose a single—non-multi-part—question. The optional argument is the number of points for this problem.

```

2497 \def\eqe@gobbletoend#1\end{}
2498 \def\eqe@grabarg#1\end{\def\numpoints{#1}}
2499 \def\eqe@isPts{1}

```

We reset some exerquiz parameters for eqexam

```

2500 \aboveexskip{3pt}\belowexskip{3pt}
2501 \partstopsep{3pt}\partsiitemsep{3pt}\partsparsep{0pt}
2502 \rowsepDefault{3pt}\partstabcolsep{1.5pt}%
2503 \partstabtopsep{3pt}\partstabrowsep{3pt}
2504 \renewcommand\belowexsolnskip{{}}%
2505 \let\isProbEnv\eqe@NO
2506 \let\topofprobbhook\relax

```

Normally, you don't leave a blank line between the beginning of problem and the statement of problem. If you're in the habit of doing that, we can absorb that blank space. See similar command above the definition of `problem*` for more information.

```

2507 \def\eqe@p@gobnxtprDef{\ifnextchar\par{\eqe@p@gobtop@rnext}{}}
2508 \def\eqe@p@gobtop@rnext#1\par{}
2509 \def\applyparfixesp{\let\eqe@p@gobnxtpr\eqe@p@gobnxtprDef}
2510 %\def\cancelparfixesp{\let\eqe@p@gobnxtpr\relax}
2511 \def\cancelparfixesp{\let\eqe@p@gobnxtpr\ignorespaces}

```

The problem environment now begins.

```

2512 \newcommand{\problem}[1][\lowercase{\def\@rgi{#1}}%

```

Support for h and H argument when it is the first argument.

```

2513 \ifx\@rgi\eqe@h
2514 \def\hidden@ttr{[1]}\let\numpoints\@empty\else
2515 \let\hidden@ttr\relax\def\numpoints{#1}\fi
2516 \problem@cont}
2517 \newcommand{\problem@cont}[1][\let\isProbEnv\eqe@YES
2518 \def\@rgi{#1}\ifx\hidden@ttr\relax % assume a number or empty
2519 \ifx\@rgi\@empty\else\def\hidden@ttr{[1]}\fi\fi
2520 \if\eqe@isPts0\global\let\eqe@prevProbZero\eqe@YES\else
2521 \global\let\eqe@prevProbZero\eqe@NO\fi
2522 \global\thereissolutionfalse
2523 \def\numpointsEmpty{0}%
2524 \ifx\numpoints\@empty\def\numpoints{0}\def\numpointsEmpty{1}%
2525 \gdef\eqe@isPts{0}\else\gdef\eqe@isPts{1}\fi
2526 \makeRoomForProb{\@nbaselineskip\baselineskip}{\eqe@isPts}%
2527 \gdef\probstar{x}\let\afterlabelhskip\@empty
2528 \ifx\marginpoints\@empty\else
2529 \if\numpointsEmpty\eqe@0ne\let\marginpoints\@gobbletwo\fi

```

(2011/5/13) We add a \* feature. When the author types [`*3`], it is a three point problem, but the value is expressed in-line, not in the margins.

```

2530 \def\@lti{\let\@isitstar\eqe@0ne\eqe@grabarg}%
2531 \def\@ltii{\let\@isitstar\eqe@Zero\eqe@gobbletoend}%
2532 \expandafter\@ifstar\expandafter\@lti
2533 \expandafter\@ltii\numpoints\end
2534 \if\@isitstar\eqe@0ne\addtocounter{eqpointvalue}{\numpoints}%
2535 \marktotalvalue
2536 \ifdisplyPoints
2537 \def\marginparafterhook{\PTs{\numpoints}\space}\else

```

```

2538         \def\marginparafterhook{\PTs*\numpoints}\space}\fi\else
2539         \def\marginparpriorhook{\noindent
2540         \probvalue{\numpoints}{0}}\fi
2541     \fi
2542     \setcounter{eq@count}{\value{eqquestionnoi}}%
2543     \addtocounter{eq@count}{1}%
2544     \ifnum\value{eq@count}=1\relax
2545         \eqe@IW0\@auxout{\string
2546         \csarg\string\gdef{pagenofirstprob\thisexamlabel}{\thepage}}%
2547     \fi

```

`\topofprobhook` is a general purpose hook at the top of the `problem` environment, before the beginning of the nested `equestions` environment.

```

2548     \topofprobhook
2549     \begin{equestions}%

```

**Begin exercise env.** Insert `h` or `H` when it is the first argument through `\hidden@ttr`, then begin the exercise environment using `eqquestionnoi`.

```

2550     \edef\ctrl@exp@exercise{\noexpand
2551         \begin{exercise}[eqquestionnoi]\hidden@ttr}%
2552     \ctrl@exp@exercise\ignorespaces\eqe@p@gobnxtpr}

```

**end problem env.** The code for the end of the problem environment.

```

2553 \def\endproblem{\end{exercise}}%
2554 \end{equestions}%
2555 \ifeqlocalversion\ifeqglobalversion
2556     \xdef\eqe@tmp{\noexpand\forVersion{\eq@selectedVersion}}%
2557     \aftergroup\eqe@tmp
2558 \fi\fi
2559 \global\eqlocalversionfalse
2560 \ifdisplyPoints\global\l@stDisplyPointstrue
2561 \else\global\l@stDisplyPointsfalse\fi
2562 \global\let\eqe@fpmrk\@empty}

```

`\PTs` When you specify `\auto` for the optional argument of the `problem*` environment, `\itemPTsTxt` when each item must have the command `\PTs` to assign the value of that question. `\itemPTsFormatted` The `\PTs` has one optional star-parameter, and one required parameter. The required parameter is the number of points for this item, if the `*` is specified, then the point value is not typeset in the document.

The command `\itemPTsTxt` has one argument, the number of points for this item. This argument is passed from the `\PTs` command. You can redefine the way the points appear in the document using `\itemPTsTxt`. As separate command `\itemPTsFormatted` is used to put parentheses around `\itemPTsTxt`. If the `*` option is taken with `\PTs`, then you are free to place `\itemPTsTxt` anywhere in the problem statement.

```

2563 \newcommand\itemPTsTxt[1]{\eqe@ptsFmt{#1}\,\text{%
2564     \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi}$}
2565 \newcommand\itemPTsEaTxt[1]{\eqe@ptsFmt{#1}\,\text{%
2566     \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi\space\eqeachLabel}$}
2567 \newcommand{\itemPTsFormatted}[1]{( #1 )}

```

`\obeyPTsStar` (2013/12/04) Added `\obeyPTsStar` and `\ignorePTsStar`. The former is the default. If `\ignorePTsStar` is in effect, `\PTs*` is treated as `\PTs`. Useful for assigning points to parts of a question, but not explicitly telling the student. When you compile for the key (with `answerkey` option), you can include `\ignorePTsStar` so the instructor can see the assigned points for each part.

```

2568 \newif\ifObeyPTsStar \ObeyPTsStartrue
2569 \def\obeyPTsStar{\global\ObeyPTsStartrue}
2570 \def\ignorePTsStar{\global\ObeyPTsStarfalse}
2571 \def\PTs{\leavevmode\ifstar{\@PTs{*}}{\@PTs{x}}}
2572 \def\@PTs#1#2{%
2573     \if\@reportpoints\eq@Zero\ignorespaces\else
2574         \if\eq@pointsPartsId\eq@One
2575             \addtocounter{eq@pointvalue}{#2}\@marktotalvalue
2576             \addtocounter{eq@count}{#2}\fi

```

`\ignorePTsStar` If `\ignorePTsStar` is in effect (`\ObeyPTsStarfalse`), we gobble the `\else` and let the code flow through.

```

2577     \if#1*\ifObeyPTsStar\ignorespaces\else\expandafter\@gobble\fi
2578     \else
    (2012/04/26) Wrapped \itemPTsFormatted{\itemPTsTxt{#2}} as the argument
    of \eq@movePTs. \eq@movePTs does nothing by default, but may be redefined,
    for example, to place the value of each part on the margin. The default definition
    of \eq@movePTs follows.

```

```

2579     \eq@movePTs{\itemPTsFormatted{\itemPTsTxt{#2}}}\fi
2580     \fi
2581 }
2582 \def\eq@movePTs#1{#1}

```

`\Do<num>` The `\isItD@` tests to see if the next token is `\Do`, if yes, it marks it and calls `\y@st@Do`, which gets the argument if the `\Do` token.

```

2583 \let\auto\relax
2584 \def\isItD@{\@ifnextchar\Do{\let\yest@D@\eq@YES\y@st@Do}
2585     {\let\yest@D@\eq@NO\@gobblet@end}}
2586 \let\yest@D@\eq@NO
2587 \let\D@Num\@empty
2588 \def\y@st@Do\Do#1\end{\gdef\D@Num{#1}}
2589 \def\@gobblet@end#1\end{}

    Test the argument to see if it is the \auto token
2590 \def\is@uto#1\auto#2\@nil{\def\eq@argi{#1}\def\eq@argii{#2}%
2591     \ifx\eq@argii\@empty\let\isit@uto\eq@Zero\else
2592         \let\isit@uto\eq@One\fi
2593 }

```

**problem\*** The `problem*` environment is used to pose a multi-part question. The `parts` environment is used to enumerate the parts.

We create a `Id` for the points specified by the first (and second) optional parameters: 0 (total points specified); 1 (`\auto` specified); 2 (points each specified); 4

(\Do second optional parameter); a value of \relax means no points specified (the default).

2594 \let\eqe@pointsPartsId\relax

```
2595 \def\leadinitemWarningStar{\PackageWarning{eqexam}
2596   {Using the star (*) in front of the points\MessageBreak
2597   designator is not allowed when there is a
2598   \MessageBreak\string\leadinitem}}
2599 \def\tableadinWarningStar{\PackageWarning{eqexam}
2600   {Using the star (*) in front of the points\MessageBreak
2601   designator is not allowed when there is a
2602   \MessageBreak\string\tableadin}}
2603 \let\isProbStarEnv\eqe@NO
2604 \let\topofprobstarhook\relax
```

Added \eqe@gobnxtpar to hopefully, eliminate the need to leave no pars (blank lines) between \begin{problem\*} and either \leadinitem or \tableadin. The default is to not affect the old behavior. You have to declare \applyparfixes to apply the fix; revert back to the old behavior with \cancelparfixes

\applyparfixes  
\cancelparfixes

```
2605 \def\eqe@ps@gobtop@r#1\par{\eqe@isle@dinnext}
2606 \def\eqe@isle@dinnext{\@ifnextchar\leadinitem{\vskip-\baselineskip}}
2607 \def\restorele@dinpfixDef{\@ifnextchar\par{\eqe@ps@gobtop@r}{}}
2608 \let\eqe@ps@gobnxtpar\restorele@dinpfixDef
2609 \def\applyleadinfix{\let\eqe@gobnxtpar\restorele@dinpfixDef}
2610 \def\cancelleadinfix{\let\eqe@gobnxtpar\relax}
2611 \def\applyparfixes{\PackageInfo{eqexam}
2612   {Applying paragraph fixes to problem\MessageBreak
2613   and problem* environments}\applyparfixesp\applyleadinfix}
2614 \def\cancelparfixes{\PackageInfo{eqexam}
2615   {Cancelling paragraph fixes to problem\MessageBreak
2616   and problem* environments}\cancelparfixesp
2617   \cancelleadinfix}
```

The default behavior is to do nothing about blank lines.

```
2618 \cancelparfixesp\cancelleadinfix
2619 \let\nutralizeparfixes\cancelparfixes
```

We now begin the code for the problem\* environment.

```
2620 \csarg\def{problem*}{\let\isProbEnv\eqe@YES
2621   \let\isProbStarEnv\eqe@YES
2622   \global\thereissolutionfalse
2623   \@ifnextchar[{\pr@bl@m@star}{\pr@bl@m@star[]}}
2624 \def\pr@bl@m@star[#1]{%
2625   \@ifnextchar[{\pr@blem@star{#1}}{\pr@blem@star{#1}[]}}
2626 \def\pr@blem@star#1[#2]{%
2627   \if\eqe@isPts0\global\let\eqe@prevProbZero\eqe@YES\else
2628   \global\let\eqe@prevProbZero\eqe@NO\fi
2629   \def\numpoints{#1}\ifx\numpoints\@empty
2630   \gdef\eqe@isPts{0}\else\gdef\eqe@isPts{1}\fi
2631   \makeRoomForProb{\@nbaselineskip\baselineskip}{\eqe@isPts}%
```

Let the \Do commands to internal versions

```

2632 \let\DoNum\eqe@DoNum\let\nDoNum\eqe@nDoNum
2633 \let\OutOfNum\eqe@OutOfNum\let\nOutOfNum\eqe@nOutOfNum

2634 % \proofingsymbol{\ding{52}}}%
2635 \gdef\probstar{*}%
2636 \gdef\pr@b@secondarg{#2}\setcounter{eq@count}{0}%
2637 \let\afterlabelhskip\@empty
2638 \global\let\probpointseach\@empty\def\numpoints{#1}%

```

Determine if the argument begins with \*

```

2639 \@ifstar{\let\@isitstar\eqe@One\eqe@grabarg}%
2640 {\let\@isitstar\eqe@Zero\eqe@gobbletoend}#1\end

```

If this `problem*` environment has a `\leadinitem`, the `*` option is not allowed

```

2641 \if\@isitstar\eqe@One
2642 \@tempcnta=\thequestionnoi\relax
2643 \advance\@tempcnta1\relax
2644 \@ifundefined{leadinitem\thisexamlabel-\the\@tempcnta}{}
2645 {\leadinitemWarningStar\let\@isitstar\eqe@Zero}%
2646 \@ifundefined{tableadin\thisexamlabel-\the\@tempcnta}{}
2647 {\tableadinWarningStar\let\@isitstar\eqe@Zero}%
2648 \fi

```

Check for the `\auto` keyword . There are problems when the first two digits are the same, the old comparison would say that the first parameter is `\auto`, which is it not. Replace the old comparison with a more robust method.

```

2649 \expandafter\is@uto\numpoints\auto\@nil
2650 \if\isit@uto\eqe@One\let\eqe@pointsPartsId\eqe@One
2651 \global\let\probpointseach\relax

```

The author has requested `\auto`

```

2652 \def\eqe@next{\autocalcparts}%
2653 \else
2654 \ifx\pr@b@secondarg\@empty\else
2655 \let\eqe@pointsPartsId\eqe@Four
2656 \isItD@#2\end\fi

```

Not `\auto` so either `<num>ea` or `<num>`

```

2657 \def\eqe@next{\manualcalcparts{\numpoints}}%
2658 \fi\eqe@next

```

2012/11/30 added `\leadinitem` defined `eqexam.def` as a dummy command, which we the `\let` here to `\eqe@leadinitem`.

```

2659 \let\leadinitem\eqe@leadinitem

```

2015/05/31 added `\tableadin` defined `eqexam.def` as a dummy command, which we the `\let` here to `\eqe@leadinitem`.

```

2660 \let\tableadin\eqe@tableadin

```

`\topofprobstarhook` is a general purpose hook at the top of the `problem*` environment, before the beginning of the nested `eqquestions` environment.

```

2661 \topofprobstarhook
2662 \ifeqfortextbook
2663 \writeT@SolnFile{\protect\global
2664 \protect\frstProbNumShownfalse}\fi
2665 \begin{eqquestions}%
2666 \begin{exercise}[eqquestionnoi]*\eqe@gobnxtpar}%

2667 \def\ftb@endprobstarCks{%
2668 \ifWithinANSGrp
2669 \PackageError{eqexam}{\string\bGrpANS\space is still open}
2670 {You need to match it with an \string\bGrpANS,
2671 or remove it.}%
2672 \fi
2673 }

\endproblem* begins here.

2674 \csarg\def{endproblem*}{%
2675 \eqe@IW0\@auxout{\string\csarg\string
2676 \gdef{nPartsThisProb\thisexamlabel.\theeqquestionnoi}%
2677 {\arabic{partno}}}%
2678 \ifx\probpointseach\@empty\else
2679 \ifx\probpointseach\auto
2680 \eqe@IW0\@auxout{\string\csarg\string
2681 \gdef{prob\thisexamlabel.\theeqquestionnoi}%
2682 {\theeq@count}}%
2683 \else
2684 \setcounter{eq@count}{\value{partno}}%
2685 \ifx\pr@b@secondarg\@empty\else
2686 \bgroup\toks0=\expandafter{\pr@b@secondarg}%
2687 \expandafter\isItD@the\toks0 \end
    If there is a \Do, we write this info to AUX.
2688 \ifx\yest@D@\eqe@YES\ifx\eqe@pointsEach\eqe@YES
2689 \eqe@IW0\@auxout{\string
2690 \csarg\string\gdef{DoNumThisProb\thisexamlabel.%
2691 \theeqquestionnoi}{\D@Num}}%
2692 \@tempcnta = \value{eq@count}%
2693 \advance\@tempcnta -\D@Num
2694 \global\advance\value{eq@count}-\@tempcnta
2695 % 3.0k
2696 \@tempcnta=\value{partno}%
2697 \advance\@tempcnta -\D@Num
2698 \multiply\@tempcnta by\argi
2699 \addtocounter{eqpointvalue}{-\@tempcnta}%
2700 \fi\fi
2701 \egroup
2702 \fi
2703 \multiply\value{eq@count}\argi
2704 \eqe@IW0\@auxout{\string\csarg\string
2705 \gdef{prob\thisexamlabel.\theeqquestionnoi}{\theeq@count}}%
2706 \fi\fi

```

```

2707 \end{exercise}%
2708 \end{eqequestions}%
2709 \ifeqfortextbook\ftb@endprobstarCks\fi
2710 \ifeqlocalversion\ifeqglobalversion
2711 \xdef\eqe@tmp{\noexpand\forVersion{\eq@selectedVersion}}}%
2712 \aftergroup\eqe@tmp
2713 \fi\fi
2714 \global\eqlocalversionfalse

2715 \@markttotalvalue
2716 \ifdispl@yPoints\global\l@stDispl@yPointstrue
2717 \else\global\l@stDispl@yPointsfalse\fi
2718 }

```

`\leadinitem` Some authors, no me among them, want to post a question with parts (`problem*`) without an introductory sentence. Scandalous!

```

\begin{problem*}\relax
\leadinitem Prove that  $x^2 \geq (x+1)^2$  for  $x \geq 1$ .
\begin{solution} soln\end{solution}
\begin{parts}
\item Use the principle of mathematical induction to prove that
 $4^n \geq 3n^2$  for all positive integers  $n$ .
\begin{solution} soln\end{solution}
\end{parts}
\end{problem*}

```

Below is my solution to this problem. There were changes in both `eqexam` and `exerquiz` (`eqexam.def`). Here are the listing of changes for my future reference.

- Defined `\eqe@leadinitem`, `\eqe@@leadinitem`, and `\leadinitem` below. The first one mostly tests whether we are permitted to use the `\leadinitem` command. If `\solutionparshape` is empty, the command may be used. (This gives a restriction of only one use per a single `problem*` environment.) Control is passed to `\eqe@@leadinitem` which does all the work.
- In the definition of `\pr@blem@star` (part of the startup for `problem*` we say `\let\leadinitem=\eqe@leadinitem`
- In the startup code for `exercise@parts@list` (`exerquiz`) lines containing `\solutionparshape` and `\let\leadinitem\eqe@leadinitem@parts` are found. Also, when the `parts` environment is complete, we restore the default definition of `\leadinitem`, `\let\leadinitem\leadinitem@external`.
- In the definition of `\eqe@@leadinitem`, we also define `\solutionparshape` to help control the paragraphs for this delicate problem of a lead-in item. Its normal definition is `\@empty`, and this fact used to detect whether the document author is trying to use two `\leadinitems` in one `problem*` environment.



- At the beginning of the definition of the command `\@exercise` (in `exerquiz` we `\let \solutionparshape` to `\@empty`).
- So too does `\solutionparshape` appear in the definition of `\vspaceFmt` (`exerquiz`).
- `\solutionparshape` use used to correctly format in the `solutionsafter`, found in `\solnexer@@woparts` (`exerquiz`).

As can be seen, this simple feature is not so simple.

Set the default value of `\solutionparshape`.

```

2719 \let\solutionparshape\@empty
\leadinitem This is the beginning of \leadinitem, at least the the active definition within the
              problem* environment, and outside the parts environment.
2720 \def\eqe@leadinitem{%
2721     \ifx\solutionparshape\@empty
        \solutionparshape empty means \leadinitem hasn't been used yet, so we can
        use it here by passing control to \eqe@@leadinitem, otherwise, we toss an excep-
        tion.
2722         \def\eqe@next{\eqe@@leadinitem}%
2723         \ifeqfortextbook\writeT@SolnFile{\protect\bpartsmrk}\fi
2724     \else
2725         \def\eqe@next{\PackageError{eqexam}
2726             {The \string\leadinitem\space command may
2727             only be used\MessageBreak once per problem* environment}
2728             {Remove all but one of the \string\leadinitem\space
2729             commands.}}}%
2730     \fi\eqe@next
2731 }
2732 \newif\ifisleadin \isleadinfalse
2733 \let\isparshapeExpanded\eqe@NO
2734 \def\eqe@@leadinitem{\if\itsforleadinitem\eqe@NO
2735     \setcounter{partno}{0}\fi\refstepcounter{partno}%
        After initializing the counter, we make various calculations
2736     \eq@initializeContAnnot
2737     \isleadintrue\let\isitleadin\eqe@YES
        We signal a \leadinitem by writing a special command to the AUX file, this is
        used to disallow the use of the * option of points.
2738     \eqe@writetoAux{\string\csarg\string
2739         \gdef{leadinitem\thisexamlabel-\theeqquestionnoi}{}}%
2740     \settowidth{\eq@tmpdima}{\normalfont\parts@indent\eqe@prtsepPrb}%
2741     \xdef\leadinIndentPrtSep{\the\eq@tmpdima}%
2742     \setlength{\eq@tmpdima}{\eqemargin+\eq@tmpdima}%
2743     \xdef\leadinIndent{\the\eq@tmpdima}%
2744     \settowidth{\eq@tmplength}{\parts@indent}%
2745     \edef\partsleadinIndent{\the\eq@tmplength}%

```

```

2746 \setlength{\eq@tmplength}{\linewidth-\leadinIndentPrtSep}%
2747 \edef\leadinIndentLength{\the\eq@tmplength}%
    Modify \@listii from core LaTeX: Increase \leftmarginii and \linewidth
    appropriately.
2748 \ifundefined{@listii@SAVE}{\global\let\@listii@SAVE\@listii}{}%
2749 \expandafter\def\expandafter\@listii\expandafter{\@listii@SAVE
2750 \advance\leftmarginii\leadinIndentPrtSep\relax
2751 \leftmargin\leftmarginii \labelwidth\leftmarginii
2752 \advance\labelwidth-\labelsep
2753 % \advance\linewidth\leadinIndentPrtSep\relax
2754 }%

```

The parshape for the first paragraph of the \leadinitem.

```

2755 \parshape=2 \eqmargin \linewidth \leadinIndent \leadinIndentLength
    Here is the definition of \solutionparshape, we use the shape of subsequent
    paragraphs.
2756 \xdef\solutionparshape{\noexpand\linewidth\leadinIndentLength
2757 \noexpand\parshape=1 \leadinIndent\space\leadinIndentLength}%

```

After the current paragraph, we execute \solutionparshape, then empty out \everypar. Seems to work.

```

2758 \global\let\isparshapeExpanded\eq@NO
2759 \everypar{\solutionparshape
2760 \global\let\isparshapeExpanded\eq@YES\everypar{}}%

```

These next lines were taken from \eq@item@common in exerquiz. The command \eq@insertContAnnot was removed from the \eq@item@common code, it is not needed here and caused trouble. We \let \eq@item to \eq@leadin@item to continue the flow. \@ckhide checks for an optional argument (h or H) and sets switches as appropriate.

```

2761 \let\eq@item\eq@leadin@item\eq@rtc\lcm@rk
2762 \def\currhideopt{x}\eq@hidesolutionfalse\eq@nolinkfalse
2763 \@ifnextchar[{\@ckhide}{\eq@item}%
2764 }

```

The final step. If the solution is hidden, we do not write the header.

```

2765 \def\eq@leadin@item{\eq@ckglobalhide\ifeq@hidesolution\else
2766 \global\let\eqExerSolnHeader\eq@@writeexheaderlist\fi
2767 \ifeq@nosolutions\eq@nolinktrue\fi
2768 \ifeq@solutionsafter\eq@nolinktrue\fi
2769 \unskip\noindent\makebox[\partslleadinIndent]{\eqexlisttabheader}%
2770 \eq@prtsepPrb\ignorespaces
2771 }

```

\tableadin \tableadin is \let to \eq@tableadin within the problem\* env. The switch \if@tableadinitem is defined in exerquiz.

```

2772 \def\eq@tableadin{\@tableadinitemtrue\eq@writetoAux{\protect
2773 \csarg\protect\gdef{tableadin\thisexamlabell-\theeqquestionnoi}{}}%
2774 }

```

`\pushProblem` There may be an occasion when a multi-part question needs to be broken between parts. use the `\pushProblem` and `\popProblem` for this purpose. The push saves the counter value, and ends the `parts` environment. The pop restarts the `parts`, and resets the parts counter.

In the example below, we have our parts in a `multicols` environment, we `\pushProblem`, close `multicols`, `\popProblem` and continue with the multi-parts in single column.

```

\item Compute  $\lim_{x \rightarrow 2} \{ \text{text}\{-\}\}$   $f(x)$ 
\begin{solution}[1in]\end{solution}
\pushProblem
\end{multicols}
\popProblem
\item What value(s) of  $x$  make the function  $f(x)$ 
continuous at  $x=2$ ?
\begin{solution}[.5in]\end{solution}
\end{parts}

```

2775 `\newcommand\pushProblem{\xdef\nlastItem{\arabic{partno}}\end{parts}}`

(2013/05/30) Adding an optional parameter to `\popProblem` to match the option argument of the `parts` environment. This enables you to push a `list` environment and push a `tabular` environment.

```

2776 \newcommand\popProblem[1] [] {%
2777     \def\@argi{#1}\ifx\@argi\@empty
2778     \def\eqe@bParts{\begin{parts}}\else
2779     \def\eqe@bParts{\begin{parts}[#1]}\fi
2780     \eqe@bParts\setcounter{partno}{\nlastItem}}

```

2781 `\def\lastPageTotal{0pt}`

2782 `\def\marginparafterhook{\xdef\lastPageTotal{\the\pagetotal}}`

`\manualcalcparts` is the command calculates points when the argument is *not* `\auto`. The macro `\prob@Arg` determines if the points argument passed is of the form `<num>ea`.

```

2783 \def\prob@Arg#1ea#2\end{\def\argi{#1}\def\argii{#2}}

```

Now begin `\manualcalcparts`; #1 is the number of points, which may be of the form `<num>ea`, or just `<num>`.

```

2784 \let\eqe@pointsEach\eqe@NO
2785 \def\manualcalcparts#1{%
2786     \expandafter\prob@Arg#1ea\end
2787     \ifx\argii\@empty\edef\numpoints{#1}%
2788     \let\eqe@pointsEach\eqe@NO

```

Argument form is `<num>` assumed: Total points specified, we should ignore any `\PTs` commands.

```

2789     \let\eqe@pointsPartsId\eqe@Zero
2790     \else

```

Argument form is `<num>ea` assumed: Again we should ignore any `\PTs` commands.

```

2791     \gdef\probpointseach{x}\let\eqe@pointsPartsId\eqe@Two
2792     \let\eqe@pointsEach\eqe@YES

```

Points each specified

```

2793      \setcounter{eq@count}{\value{eqquestionnoi}}%
2794      \addtocounter{eq@count}{1}\csarg
2795      \ifx{prob\thisexamlabel.\theeq@count}\relax
2796          \def\numpoints{\argi}\else
2797          \def\numpoints{\expandafter
2798              \csname prob\thisexamlabel.\theeq@count\endcsname}%
2799      \fi

2800      \ifx\yest@D@eqe@YES
2801          \setcounter{eq@count}{\value{eqquestionnoi}}%
2802          \addtocounter{eq@count}{1}%
2803          \@ifundefined{nPartsThisProb\thisexamlabel.\theeq@count}
2804              {\makeOutOfNum{0}\makeDoNum{0}}{%
2805              \expandafter\makeOutOfNum{%
2806              \csname nPartsThisProb\thisexamlabel.%
2807                  \theeq@count\endcsname}%
2808              \expandafter\makeDoNum{%
2809              \csname DoNumThisProb\thisexamlabel.%
2810                  \theeq@count\endcsname}}%
2811      \fi
2812  \fi

```

If `\marginpoints` is `\@empty`, the author has chosen the `nopoints` option or used the `\NoPoints` command.

```

2813      \ifx\marginpoints\@empty
2814          No points for this exam
2815      \else
2816          \ifx\argi\@empty
2817              If \argi is empty, no points were specified, so we \let \marginpoints to \@empty
2818              \let\marginpoints\@empty
2819          \else
2820              Points are displayed in margins or inline
2821              \ifx\argii\@empty

```

**Total points specified**

```

2819      \ifx\marginpoints\@empty\else
2820      \if\@isitstar\eqe@One
2821          Points to appear “in-line” rather than in the margins
2822          \addtocounter{eqpointvalue}{#1}%
2823          \@markttotalvalue\ifdisplyPoints
2824          \def\marginparafterhook{\itemPTsFormatted{
2825              \itemPTsTxt{\numpoints}}\space}\fi
2826      \else
2827          Points appear in the margins
2828          \def\marginparpriorhook{\noindent

```

```

2827             \probvalue{\numpoints}{0}}%
2828         \fi
2829     \fi
2830 \else

```

### Points each specified

```

2831         \ifx\marginpoints\@empty\else
2832             \if\@isitstar\eqe@0ne\ifdisplyPoints
Points to appear “in-line” rather than in the margins
2833                 \def\marginparafterhook{%
2834                     \itemPTsFormatted{\itemPTsEaTxt{\argi}}\space}\fi
2835             \else

```

### Points appear in the margins

```

2836             \ifdisplyPoints
2837                 \def\marginparpriorhook{\noindent
2838                     \marginpoints{\numpoints}{\argi}}\fi
2839             \fi

```

Don’t remember why I named this command the way I did, but it is use to pass the number of points, when there is a `\leadinitem`.

```

2840             \edef\eqp@rtc@lcm@rk{\noexpand
2841                 \addtocounter{eqpointvalue}{\argi}}%
2842                 \noexpand\@markttotalvalue}%
2843             \fi
2844         \fi
2845     \fi
2846 \fi
2847 \ifnum\value{eq@count}=1\relax
2848     \eqe@IW0\@auxout{\string\csarg\string
2849         \gdef{pagenofirstprob\thisexamlabel}{\thepage}}%
2850 \fi
2851 }

```

`\autocalcparts` is the command that computes the total points when the author specifies `\auto` as the optional argument of `problem*`. The commands `\acp@mpah` and `\acp@mpph` were recently (2012/04/21) separated out to allow for additional customization, without re-defining the whole of `\autocalcparts`.

```

2852 \def\acp@mpah{\itemPTsFormatted{\itemPTsTxt{\numpoints}}\space}
2853 \def\acp@mpph{\noindent\marginpoints{\numpoints}{0}}
2854 \def\autocalcparts{%
2855     \setcounter{eq@count}{\value{eqquestionnoi}}%
2856     \addtocounter{eq@count}{1}%
2857     \csarg\ifx{prob\thisexamlabel.\theeq@count}\relax
2858         \def\numpoints{0}% assume zero points until we get the total
2859     \else
2860         \edef\numpoints{\@nameuse{prob\thisexamlabel.\theeq@count}}%
2861     \fi
2862 %     \addtocounter{eqpointvalue}{\numpoints}\@markttotalvalue

```

```

2863 \ifx\marginpoints\@empty\else
2864 \ifdisplyPoints
2865 \if\@isitstar\eqe@One
    If we have *\auto, the total is to appear inline.
2866 \def\marginparafterhook{\acp@mpah}\else
    Otherwise, the total will appear in the margin.
2867 \def\marginparpriorhook{\acp@mpph}\fi
2868 \fi
2869 \fi
2870 \setcounter{eq@count}{0}%
2871 }

```

`\forproblem` When typing solutions from assigned problems in a textbook, the problems assigned are not consecutive. You can set the problem number before the problem environments by using the `\forproblem` command. The one required argument is the problem number: `\forproblem{10}`.

```

2872 \let\eqe@fpmrk\@empty
2873 \newcommand{\forproblem}[1]{\def\eqe@fpmrk{-\the@exno}%
2874 \setcounter{eqquestionnoi}{#1 - 1}}

```

A similar comment for `\item`. These are useful for making out solution sets to homework assignments where problems are assigned from the textbook and you want to give a solution to problem 12, part (b), An example of usage is

```

\forproblem{12}
\begin{problem*}
Factor each.
\begin{parts}
\foritem{b} $ x^2 + 2x + 1 = ( x + 1 )^2 $
\item $ x^2 - x - 2 = ( x - 2 )( x + 1 ) $ % this is part (c)
\foritem{e} $ x^2 + 7x + 10 = ( x - 2 )( x + 7 ) $
\item ... % this will be part (f)
\end{parts}
\end{problem*}

```

```

2875 \newcommand{\foritem}[2][ ]{%
2876 \setcounter{partno}{0}\def\@rgi{#1}\ifx\@rgi\@empty
2877 \def\eqe@nextitem{\item}\else
2878 \def\eqe@nextitem{\item[#1]}\fi
2879 \foritem@cont{#2}}
2880 \newcommand{\forleadinitem}[2][ ]{%
2881 \setcounter{partno}{0}\def\@rgi{#1}\ifx\@rgi\@empty
2882 \def\eqe@nextitem{\leadinitem}\else
2883 \def\eqe@nextitem{\leadinitem[#1]}\fi
2884 \foritem@cont{#2}}

```

(2017/01/04) Modified `\foritem@cont` to allow for numbering parts.

```

2885 \def\foritem@cont#1{\ifuseNumForParts
2886 \edef\fliPartNo{#1}\setcounter{partno}{#1-1}\else
2887 \lowercase{\def\eqe@selectedItem{#1}}%

```

```

2888 \let\eq@initLoop\eqe@NO
2889 \loop
2890   \stepcounter{partno}\expandafter
2891   \if\alph{partno}\eq@selectedItem
2892     \let\eq@initLoop\eqe@YES\fi
2893 \ifx\eq@initLoop\eqe@NO\repeat
2894 \edef\fliPartNo{\the\c@partno}%
2895 \addtocounter{partno}{-1}\fi\let\itsforleadinitem\eqe@YES
2896 \eqe@nextitem}

```

The command `\eqe@insertContAnnot` attempts to insert a string just prior to a part, if that part begins a new page. To get it right, it promotes a new page using the default of .25in. The optional parameter allows you to insert a new value; this may be needed to get the string `\annotContStr` placed properly. The commands

`\annotContStr` The string that is typeset by the `\eqe@insertContAnnot` command.

`\acvspace` User access to changing the vertical spacing `\promoteNewPage` uses within `\eqe@insertContAnnot`.

`\resetacvspace` Resets the vertical spacing back to its default.

```

2897 \newcommand{\annotContStr}{%
2898   \textbf{Problem~{\eqeCurrProb} continued\strut}}
2899 \newcommand{\acvspace}[1]{\def\ic@vspace{#1}\setlength\iacvspace{#1}}

```

`\ic@vspacedefault` is the default vertical spacing used by `\eqe@insertContAnnot`

```

2900 \newlength\iacvspace
2901 \newcommand{\ic@vspacedefault}{1in} % changed .25in to 1in 2012/12/04
2902 \newcommand{\resetacvspace}{\setlength\iacvspace{\ic@vspacedefault}}
2903 \resetacvspace

```

`\eqe@insertContAnnot` promotes a new page, and if the current page is different than the starting page, `\eqe@currProbStartPage`, we insert `\annotContStr`.

```

2904 \def\@nnotContStrSkip{\vskip3pt}
2905 \newcommand{\eqe@insertContAnnot}[1][\iacvspace]{\promoteNewPage[#1]%
2906   \ifnum\arabic{page}>\eqe@currProbStartPage
2907     \xdef\eqe@currProbStartPage{\arabic{page}}%
2908     {\settowidth{\eqe@tmplength}{\parts@indent\eqe@prtsepPrb}%
2909     \xdef\eqe@partsIndent{\the\eqe@tmplength}}%
2910   \ifwithinparts

```

`\eqe@item` is the list definition of `\item` which has been `\let` to `\eqe@item` within the `parts` environment.

```

2911   \@ifundefined{eq@item@latex}{\item[]\hspace*{-\eqemargin}%
2912   \ifx\solutionparshape\@empty\else\hspace*{-\eqe@partsIndent}\fi}
2913   {\eq@item@latex[]\hspace*{-\eqemargin}%
2914   \hspace*{-\eqe@partsIndent}}\else

```

If placed between problem sets, we indent as appropriate.

```

2915   \hspace*{-\eqemargin}\fi
2916   \annotContStr\@nnotContStrSkip % dpsj20
2917 \fi

```

```

2918 }
2919 \edef\eq@currProbStartPage{\arabic{page}}

\turnContAnnotOff Turn off and on this feature. The default is off.
\turnContAnnotOn 2920 \newif\ifcont@nnot \cont@nnotfalse
2921 %\newcommand{\@gobbloptone}[1] [] {}
2922 \newcommand{\turnContAnnotOff}{\global\cont@nnotfalse
2923   \global\let\eq@insertContAnnot\relax}
2924 \newcommand{\turnContAnnotOn}{\global\cont@nnottrue
2925   \global\let\eq@insertContAnnot\eqe@insertContAnnot}
2926 \turnContAnnotOff

Used for manually inserting annot
2927 \def\insertContAnnot{\eq@insertContAnnot}

\aNNewPage A simple command for inserting \newpage, only if the \answerkey option has
\qNewPage been taken.
2928 \newcommand\aNNewPage{\ifanswerkey\newpage\fi}
2929 \newcommand\qNewPage{\ifanswerkey\else\newpage\fi}

\OnBackOfPage In an effort to make maximum use of the paper, I sometimes ask the students to
solve the problem on the back of a page. The following command is an automated
instruction. Generally, we work on the back of the previous page, unless we are
on page 1, in this case we work on the back of page 1.

2930 \newcounter{backofpage}
2931 \newcommand\bopText{on the back of page~\boPage}
2932 \newcommand\bopCoverPageText{on the back of the cover page}
2933 \newcommand\OnBackOfPage[1] [\bopText] {%
2934   \refstepcounter{backofpage}\label{bop\thebackofpage}
2935   \begingroup
2936   \csarg\ifx{r@bop\thebackofpage}\relax
2937     \def\boPage{??}\else
2938     \edef\eqe@temp{\csname r@bop\thebackofpage\endcsname}%
2939     \ifx\hyper@anchor\@undefined
2940       \edef\boPage{\expandafter\@secondoftwo\eqe@temp}\else
2941       \edef\boPage{\expandafter\@secondoffive\eqe@temp}\fi
2942     \c@eq@count\boPage
2943     \advance\c@eq@count-1\relax

If on page 1, we work on the back of page 1, otherwise, we work on the back of
the previous page.

2944   \edef\boPage
2945   {%
2946     \ifx\eqex@coverpage\relax
2947       \ifnum\value{eq@count}=0
2948         1%
2949       \else
2950         \the\value{eq@count}%
2951       \fi

```



```

2952         \else
2953             \ifnum\value{eq@count}=0
2954                 -1%
2955             \else
2956                 \the\value{eq@count}%
2957             \fi
2958         \fi
2959     }%
2960 \fi
2961 \csarg\ifx{r@bop\thebackofpage}\relax#1\else
2962 \ifnum\boPage=-1\relax\bopCoverPageText\else#1\fi\fi
2963 \endgroup
2964 }

```

## 14 Vertical Space Filling Options

When the `nosolutions` or the `vspacewithsolns` is used, a vertical space is generated by the `solution` environment. Previously, this has just been a vertical white space, now, we provide the ability to fill the space with horizontal rules of different types. Below is the implementation of this.

### 14.1 General filler lines commands and controls

**\useFillerLines** When used, the vertical space is written with lines (rules, dashes, dots, grids ).

```

2965 \newcommand{\useFillerLines}{\ifx\vspaceFiller % dpsj8
2966 \vspaceFillerLines\else\@eqlinedfillertrue
2967 \let\vspaceFiller\vspaceFillerLines\fillTypeDefault
2968 \fi}

```

**\useFillerDefault** Resets the vertical space to the original white space.

```

2969 \newcommand{\useFillerDefault}{\@eqlinedfillerfalse
2970 \let\vspaceFiller\vspaceFillerDefault
2971 \let\eqWriteLine\hfill}

```

**\fillTypeHRule** Writes the line as a solid line (`\hrulefill`).

```

2972 \newcommand{\fillTypeHRule}{\let\eqWriteLine\eqWriteLineFill
2973 \let\makeVgrid\relax}

```

**\fillTypeDots** Writes the line as a dotted line (`\dotfill`).

```

2974 \newcommand{\fillTypeDots}{\let\eqWriteLine\eqWriteLineDots
2975 \let\makeVgrid\relax}

```

**\fillTypeDashLine** Writes the line as a dotted line (`\eqdashrulefill`).

```

2976 \newcommand{\fillTypeDashLine}{\let\eqWriteLine\eqWriteLineDashFill
2977 \let\makeVgrid\relax}

```

`\fillTypeBlankLine` Fills the line

```

2978 \newcommand{\fillTypeBlankLine}{\let\eqWriteLine\eqWriteLineBlankFill
2979 \let\makeVgrid\relax}

\fillTypeDefault Resets fill type back to the default, \hrulefill.
2980 \newcommand{\fillTypeDefault}{\let\eqWriteLine\eqWriteLineFill
2981 \let\makeVgrid\relax}

\fillTypeGrid Fills the space with a grid, horizontal and vertical lines.
2982 \newcommand{\fillTypeGrid}{\ifx\vspaceFiller\vspaceFillerLines
2983 \let\eqWriteLine\eqWriteLineFill\let\makeVgrid\eq@makeVgrid\fi}

\eqWriteLineColor The color of the rule to use.
2984 \newcommand{\eqWriteLineColor}[1]{\def\eq@WriteLineColor{#1}}
2985 \eqWriteLineColor{gray}

\eqWLSpacing The line spacing between the rules.
2986 \newlength\wlVspace
2987 \newcommand{\eqWLSpacing}[1]{\setlength\wlVspace{#1}}
2988 \eqWLSpacing{14.0pt}

```

**Fill Types.** We have three types of line fill: `\hrulefill`, `\dotfill`, and a custom rule `\eqdashrulefill`. The commands `\eqWriteLineFill`, `\eqWriteLineDots`, and `\eqWriteLineDashFill` implements these three types. They are `\let` to `\eqWriteLine`, which is used in `\vspaceFillerLines`.

```

2989 \newcommand{\eqWriteLineFill}{%
2990 \textcolor{\eq@WriteLineColor}{\leaders
2991 \hrule height \flboxrule\hfill}}

The first box is used in \eqdotrulefill, whereas the second two are used to
split off content using \vsplit. These are active when the options flexended and
answerkey are in force, and when \useFillerLines and \turnflanskeyOn have
been expanded.

2992 \newbox\eq@tempbox
2993 \newbox\eq@nskeyflsplit
2994 \newbox\eq@nskeyfltop
2995 \def\eqdotrulefill{\leavevmode
2996 \cleaders\hb@xt@ .44em{\copy\eq@tempbox\hss}\hfill} % dps16
2997 % \cleaders\hb@xt@ .44em{\unhcopy\eq@tempbox\hss}\hfill} % dps8
2998 \newcommand{\eqWriteLineDots}{%
2999 \textcolor{\eq@WriteLineColor}{\eqdotrulefill}}
3000 \def\eqdashrulefill{\leavevmode
3001 \cleaders\hb@xt@ .44em{\rule{.22em}{\flboxrule}\hss}\hfill\kern\z@}
3002 \newcommand{\eqWriteLineDashFill}{%
3003 \textcolor{\eq@WriteLineColor}{\eqdashrulefill}}
3004 \newcommand{\eqWriteLineBlankFill}{\hfill}
3005 \newcommand{\eqWriteLineBlankFill}{\vphantom{\hrulefill}}

```

Vertical counterparts to those above, excepting blank fill.

```

3006 \newcommand{\eqWriteLineVFill}{\leaders\vrule width\flfboxrule\vfill}
3007 \def\eqdotruleVfill{\cleaders\vbox to .44em{\vss
3008   \hbox to0pt{\hss.\hss}}\vfill}
3009 \newcommand{\eqWriteLineVDots}{\eqdotruleVfill}
3010 \def\eqdashruleVfill{\cleaders\vbox to .44em{\vss
3011   \hbox to0pt{\hss\rule{\flfboxrule}{.22em}\hss}}\vfill}
3012 \newcommand{\eqWriteLineDashVFill}{\eqdashruleVfill}

```

When the `solution` environment contains both an `nLines` and a dimension, `\usenLineDimen` by default, the dimension is used; however, if `\usenLineDimen` is specified, we force the use of the `nLines` specification. Switch back to the default using `\useVspaceDimen` `\useVspaceDimen`.

```

3013 \newif\if@equsedim \@equsedimtrue
3014 \newcommand{\useVspaceDimen}{\@equsedimtrue}
3015 \newcommand{\usenLineDimen}{\@equsedimfalse}
3016 \newif\if@eqalignfillllinestoleft \@eqalignfillllinestoleftfalse

\fillerLinesAlignDef Use \fillerLinesAlignDef and \fillerLinesOnLeftMargin to adjust the align-
\fillerLinesOnLeftMargin ment of the rule lines for this feature. The first is the default, the second one aligns
rule lines to the left margin.

3017 \newcommand{\fillerLinesOnLeftMargin}{\@eqalignfillllinestolefttrue}
3018 \newcommand{\fillerLinesAlignDef}{\@eqalignfillllinestoleftfalse}
3019 \newcommand{\priorPageBreakMsg}[1]{\def\priorP@geBre@kMsg{#1}}
3020 \let\priorP@geBre@kMsg\@empty
3021 \newcommand{\flPageBreakMsg}[1]{%
3022   \priorPageBreakMsg{\emitMessageNearBottom[\iacvspace]
3023     {\eqfititin{\Large\strut}#1}}}%
3024 }

```

The command `\makeVgrid` is an internal macro that is let to either `\relax`, as below, or to `\eqe@makeVgrid`. The latter creates a grid of fill lines.

```

3025 \let\makeVgrid\relax %dps1

```

**Key-values for fill lines** We define the `eqefillLines` xkeyval family, the keys are set through the command `\setFillLinesFmt`. We define nine keys `numbers`, `numbersep`, `color`, `topline`, `gridtype`, `fltype`, `align`, `outlineonly`, `bgonly`, and `bgcolor`. The first two are used to number the fill lines, the `color` is to color the fill lines, the last two concerning the grid type filler lines.

`numbers=(none|left|right)` This is a `numbers` is a choice key with possible values of `none`, `left` and `right`. It places a line numbering to the left or right of the line, or no line number is used if `none` is selected.

```

3026 \define@choicekey+{eqefillLines}{numbers}{\val\nr}%
3027 {none,left,right}[none]{\ifcase\nr\relax
3028   \let\eqe@numLinesL\relax\let\eqe@numLinesR\relax\or
3029   \let\eqe@numLinesL\eqe@numLinesL\let\eqe@numLinesR\relax\or
3030   \let\eqe@numLinesL\relax\let\eqe@numLinesR\eqe@numLinesR
3031   \fi

```

```

3032 }{\PackageWarning{aeb}{Bad choice for numbers, permissible values
3033   are none, left, and right. Try again}}
3034 \let\eqe@numLinesL\relax\let\eqe@numLinesR\relax
numbersep=<dimen> When numbers is either left or right, the separation between the line
and the number is numbersep. The default value is 2pt.
3035 \define@key{eqefillLines}{numbersep}[2pt]{%
3036   \setlength{\eqetmplengtha}{#1}%
3037   \edef\eqe@numbersep{\the\eqetmplengtha}}
3038 \def\eqe@numbersep{2pt}
color=<color-spec> The color key paints the lines the specified color, for example,
color=red!20. It simply defines \eq@WriteLineColor to be this color.
3039 \define@key{eqefillLines}{color}[]{\def\eq@WriteLineColor{#1}}
topline=<true|false> When \fillTypeGrid is in effect, topline=true writes an additional
line above the top most line, this line is not numbered.
3040 \define@boolkey{eqefillLines}{topline}[true]{}
3041 \KV@eqefillLines@toplinefalse
gridtype=<line|dash|dots> The gridtype key determines the line style for the grid, choices
are line (the default), dash, and dots. This key also sets the line style when the
outlineonly key is specified.
3042 \let\eqe@usedeffboxrule\eqe@NO % dps20
3043 \let\gridtypeselectd@empty % dps26
3044 \define@choicekey+{eqefillLines}{gridtype}[\val\nr]%
3045   {line,dash,dots}[line]{\edef\gridtypeselectd{\val}% dps26
3046   \ifcase\nr\relax
3047     \let\gridHLineFill\eqWriteLineFill
3048     \let\gridVLineFill\eqWriteLineVFill
3049   \or
3050     \let\gridHLineFill\eqWriteLineDashFill
3051     \let\gridVLineFill\eqWriteLineDashVFill
3052   \or
3053     \let\gridHLineFill\eqWriteLineDots
3054     \let\gridVLineFill\eqWriteLineVDots
3055   \let\eqe@usedeffboxrule\eqe@YES %dps20
3056   \fi
3057 }{\PackageWarning{aeb}{Bad choice for numbers, permissible values
3058   are line, dash, and dots. Try again}}
3059 \let\gridHLineFill\eqWriteLineFill
3060 \let\gridVLineFill\eqWriteLineVFill
fltype=<line|dash|dots|blank> The fltype key determines the type of line style for the
horizontal rendering. This key simply executes the various command versions.
3061 \define@choicekey+{eqefillLines}{fltype}[\val\nr]%
3062   {line,dash,dots,blank,grid}[line]{\ifcase\nr\relax
3063   \fillTypeHRule\or
3064   \fillTypeDashLine\or
3065   \fillTypeDots
3066   \let\eqe@usedeffboxrule\eqe@YES\or % dps20

```

```

3067 \fillTypeBlankLine\or
3068 \fillTypeGrid\fi
3069 }\PackageWarning{aeb}{Bad choice for numbers, permissible values
3070 are line, dash, dots, blank, and grid. Try again}}

align=<left|default> Another convenience key, align simply executes the command
versions.

3071 \define@choicekey{eqefillLines}{align}[\val\nr]%
3072 {default,left}[default]{\ifcase\nr\relax
3073 \fillerLinesAlignDef\or
3074 \fillerLinesOnLeftMargin\fi
3075 }\PackageWarning{aeb}{Bad choice for numbers, permissible values
3076 are line, dash, dots, blank, and grid. Try again}}
3077 \define@choicekey*{eqefillLines}{equalcells}[\val\nr]%
3078 {true,false}[true]{\ifcase\nr\relax
3079 \equalCellSizesOn\or
3080 \equalCellSizesOff\fi
3081 }

outlineonly*=<true|false> outlines the work area only, does not create horizontal or vertical
lines, other than the lines to outline the region. Obeys the gridtype key.

3082 \define@boolkey{eqefillLines}{outlineonly}[true]{ }
3083 \KV@eqefillLines@outlineonlyfalse
3084 \define@choicekey{eqefillLines}{outlineonly*}{true,false}[true]{% dps26
3085 \@nameuse{KV@eqefillLines@outlineonly#1}%
3086 \def\eqefillLines@outlineonlystar{#1}}%
3087 \let\eqefillLines@outlineonlystar\@empty

bgonly*=<true|false> colors the work area with a color of your choice, see bgcolor below.

3088 \define@boolkey{eqefillLines}{bgonly}[true]{ } % dps11
3089 \KV@eqefillLines@bgonlyfalse
3090 \define@choicekey{eqefillLines}{bgonly*}{true,false}[true]{% dps26
3091 \@nameuse{KV@eqefillLines@bgonly#1}%
3092 \def\eqefillLines@bgonlystar{#1}}%
3093 \let\eqefillLines@bgonlystar\@empty

bgcolor=<color-spec> The color to be used when bgcolor is in force. If no color has been
assigned, the color will be white.

3094 \define@key{eqefillLines}{bgcolor}[]{\def\eqe@BGColor{#1}} % dps11
3095 \let\eqe@BGColor\@empty

\fillLinesNumFmt{<cmds>} is used to specify the format for line numbers. Within the <cmds> argu-
ment, use #1 to indicate the placement of the number. For example, the code
\fillLinesNumFmt{\textbf{#1}} puts the line numbers in bold font.

3096 \def\fillLinesNumFmt#1{\def\eqe@fillLinesNumFmt##1{\def\flnum{##1}#1}}
3097 \fillLinesNumFmt{\flnum} % set to default values
3098 \define@key{eqefillLines}{numfmt}[]{\fillLinesNumFmt{#1}} % dps21
3099 \define@key{eqefillLines}{linegap}[14pt]{\def\fl@wlspace{#1}} % dps21
3100 \def\fl@wlspace{14pt}

\setFillLinesFmt{<KV-pairs>} Set the format of the fill lines, use the keys from the eqefillLines

```

defined and described above. Changes outside a group are global. An example of the usage is given here: `\setFillLinesFmt{numbers=left,color=red}`.

Other important notes: To produce horizontal lines only, use `fltype=line|dash|dots`, to produce blank space, use `fltype=blank` and to produce a grid `fltype=grid`.

```

3101 \newcommand{\setFillLinesFmt}[1]{\def\@rgi{#1}%
3102 \let\eqe@usedeffboxrule\eqe@NO %dps20
3103 \let\gridtypeseselected\@empty
3104 \ifx\@rgi\@empty
3105 \setkeys{eqefillLines}{numbers,numbersep,color,
3106 gridtype,fltype,align}%
3107 \else
3108 \let\is@outlineonly\eqe@NO\let\is@bgonly\eqe@NO
3109 \ifKV@eqefillLines@outlineonly\let\is@outlineonly\eqe@YES\fi
3110 \ifKV@eqefillLines@bgonly\let\is@bgonly\eqe@YES\fi
3111 \setkeys{eqefillLines}{#1}%
3112 \eqWLSpacing{\fl@wlspace}%
3113 \ifx\makeVgrid\relax
3114 \KV@eqefillLines@toplinefalse
3115 % \forceEqualCellsfalse % dps21
3116 \equalCellSizesOff % dpsj15
3117 \ifx\is@bgonly\eqe@YES % carryover from prev prob
3118 \ifKV@eqefillLines@bgonly
3119 \KV@eqefillLines@bgonlyfalse
3120 \else
3121 \PackageWarning{eqexam}{You must specify fltype=grid for
3122 bgonly to work.\MessageBreak
3123 Ignoring this key for now}%
3124 \KV@eqefillLines@bgonlyfalse
3125 \let\is@bgonly\eqe@NO
3126 \fi
3127 \else % not continuation
3128 \ifKV@eqefillLines@bgonly
3129 \PackageWarning{eqexam}{You must specify fltype=grid for
3130 bgonly to work.\MessageBreak
3131 Ignoring this key for now}%
3132 \KV@eqefillLines@bgonlyfalse
3133 \let\is@bgonly\eqe@NO
3134 \fi
3135 \fi
3136 % same for outline
3137 \ifx\is@outlineonly\eqe@YES % carryover from prev prob
3138 \ifKV@eqefillLines@outlineonly
3139 \KV@eqefillLines@outlineonlyfalse
3140 \else % dps18
3141 \PackageWarning{eqexam}{You must specify fltype=grid for
3142 outlineonly to work.\MessageBreak
3143 Ignoring this key for now}%
3144 \KV@eqefillLines@outlineonlyfalse

```

```

3145         \let\is@outlineonly\eqe@NO
3146     \fi
3147 \else % not continuation
3148     \ifKV@eqefillLines@outlineonly
3149         \PackageWarning{eqexam}{You must specify fltype=grid for
3150             outlineonly to work.\MessageBreak
3151             Ignoring this key for now}%
3152         \KV@eqefillLines@outlineonlyfalse
3153         \let\is@outlineonly\eqe@NO
3154     \fi
3155 \fi
3156 \let\gridHLineFill\eqWriteLine
3157 \let\gridVLineFill\eqWriteLine
3158 \else
3159     \ifx\gridtypesselected\@empty % dps26
3160         \let\gridHLineFill\eqWriteLineFill
3161         \let\gridVLineFill\eqWriteLineVFill
3162     \fi
3163 \fi
3164 \fi
3165 }

```

The next two definitions place line numbers on the left and right, respectively.

```

3166 \def\eqe@numLinesL{\ifKV@eqefillLines@topline\ifnum\@tempcntb>0 % dps5
3167     \smash{\llap{\tiny
3168         \eqe@fillLinesNumFmt{\the\@tempcntb}\hglue\eqe@numbersep\relax}}\fi
3169     \else\smash{\llap{\tiny
3170         \eqe@fillLinesNumFmt{\the\@tempcntb}\hglue\eqe@numbersep\relax}}\fi}
3171 \def\eqe@numLinesR{\smash{\rlap{\tiny
3172     \hglue\eqe@numbersep\relax\eqe@fillLinesNumFmt{\the\@tempcntb}}}}

```

The `\fillLinesLineWidth` is a special command used for adjusting the length of the horizontal lines, and is used only when `\fillTypeGrid` is in effect and when `\forceEqualCells` is true.

```

3173 \let\fillLinesLineWidth\@empty

```

A switch to signal that a grid has broken across pages. This is only set to true when the `topline=true` is specified through `\setFillLinesFmt`.

```

3174 \newif\ifgridpgbrk\gridpgbrkfalse % dps7

```

`\vspaceFillerLines`{*<dimen>*} This is an internal command, though we declare it publicly. It writes the horizontal lines (lines, dots, dashes) when the `solution` environment uses a *<dimen>* to specify space (as opposed to using the `nLines` key).

```

3175 \newif\ifwriteVertic@lFLines \writeVertic@lFLinestrue % dps16
3176 \def\fl@set@notContStrSkip{\def\@notContStrSkip{\vskip6pt}} % dpsj23
3177 \let\p@ssToFLs\relax
3178 \newif\iffl@firstpass % dps27
3179 \newcommand{\vspaceFillerLines}[1]{\begingroup\offinterlineskip %dps1
3180     \global\fl@firstpasstrue % dps27
3181     \fl@set@notContStrSkip % dpsj20

```

```

3182 \p@ssToFLs\if@eqlinedfiller
3183 \def\eqe@next{\vspaceFillerLines@i{#1}}\else
3184 \expandafter\def\expandafter\eqe@next
3185 \expandafter{\expandafter\endgroup\p@ssToFLs}\fi
3186 \global\let\p@ssToFLs\relax\eqe@next}
3187 \def\vspaceFillerLines@i#1{\advance\w1Vspace-\flfboxrule % dps25
3188 \ifx\eqe@usedeffboxrule\eqe@YES\flfboxrule=.4pt\fi % dps20
3189 \ifKV@eqefillLines@outlineonly\KV@eqefillLines@toplinetrue %dps10
3190 \KV@eqefillLines@bgonlyfalse % dps11
3191 % \forceEqualCellsfalse % dpsj15
3192 \equalCellSizesOff\let\makeVgrid\eqe@makeVgrid\fi %dps10
3193 \ifKV@eqefillLines@bgonly\KV@eqefillLines@toplinetrue
3194 \KV@eqefillLines@outlineonlyfalse\flfboxrule=.4pt % dps20
3195 % \forceEqualCellsfalse % dpsj15
3196 \equalCellSizesOff\let\makeVgrid\eqe@makeVgrid\fi %dps11
3197 \parindent0pt\relax\parskip0pt
3198 \@tempdima\w1Vspace\eqetmplengtha0pt
3199 \@tempcnta=0 \@tempcntb=1

\eqe@fillwidth is the width of the line, it is \linewidth plus a fudge factor,
which is determined dynamically by \eqe@wrtLineKernal.
3200 \def\eqe@fillwidth{\linewidth+\eqetmplengtha}%

If we are creating a grid (\fillTypeGrid), and \forceEqualCells is true, we
modify the length of the line to make equal cell sizes. \forceEq@lCells determines
the value of \fillLinesLineWidth.
3201 \eqe@wrtLineKernal
3202 \setbox\eqe@tempbox\hbox{\hb@xt@0pt{\hss.\hss}}\ht\eqe@tempbox.33pt
3203 \dp\eqe@tempbox0pt % dps16
3204 \ifx\makeVgrid\relax
3205 % \forceEqualCellsfalse % dpsj15
3206 \equalCellSizesOff
3207 \else % dps2
3208 \let\eqWriteLine\gridHLineFill
3209 \forceEq@lCells % dps2
3210 \ifx\flEqu@lLineWidth\@empty\else
3211 \linewidth=\flEqu@lLineWidth\relax\fi
3212 \ifKV@eqefillLines@bgonly\else
3213 \ifKV@eqefillLines@topline\bgroup\@tempcntb=0% dps5
3214 \makebox[0pt][l]{\eqe@x
3215 \makebox[\eqe@fillwidth]{\eqWriteLine}}\egroup
3216 \fi\fi % dps5
3217 \fi
3218 \settowidth{\eqetmplengthb}{\eqe@decPointPrb}\def\vfGo@l{#1}%
3219 \ifx\makeVgrid\relax\else
3220 \def\priorPNPAction{\xdef\fl@nRows{\the\@tempcnta}% dps27
3221 \if@eqalignfilllinestoleft\eqe@x\fi % dpsj12
3222 \makeVgrid\priorP@geBre@kMsg\global\fl@firstpassfalse}% dps4
3223 \def\postPNPAction{\global\gridpgbrktrue % dps27
3224 \setlength{\@tempdima}

```



```

3225      {(\vfGo@1+\w1Vspace)-\depthtodate}% dps25
3226      \edef\vfGo@1{\the\@tempdima}\@tempdima=\w1Vspace\relax
3227      \@tempcnta=0\relax}% dps18
3228      \fi
3229      \@whiledim\@tempdima<\vfGo@1\relax\do
3230      {\edef\depthtodate{\the\@tempdima}%
3231      \ifx\eqe@insert@more@content\relax\ifx\makeVgrid\relax
3232      \priorP@geBre@kMsg\fi\fi %dps15
3233      \eqe@insertContAnnot\vskip\w1Vspace\eqe@x

Draw the horizontal line: a rule, dotted line, dashed line
3234      \ifKV@eqefillLines@bgonly\vglue\flfboxrule\else % dpsj11 dpsj21
3235      \ifKV@eqefillLines@outlineonly\vglue\flfboxrule\else %dps10
3236      \makebox[0pt][l]{\eqe@numLinesL
3237      \makebox[\eqe@fillwidth]{\eqWriteLine}\eqe@numLinesR}\fi
3238      \fi
3239      \advance\@tempcnta1\relax\advance\@tempcntb1\relax
3240      \edef\depthtodate{\the\@tempdima}%
3241      \addtolength{\@tempdima}{\w1Vspace}%\par dps25

```

At the conclusion of the `\do` loop, we insert `\makeVgrid`, which is either `\relax` or is `\eqe@makeVgrid`, the latter is a complicated command to draw vertical lines across the horizontal ones, and to perform other tasks.

```

3242  } \xdef\fl@nRows{\the\@tempcnta}\makeVgrid % dps27
3243  \ifx\makeVgrid\relax\else %dps15

```

When there is insufficient space to accommodate the content, we place a small blank box in the lower left corner, put a information message in the log, and reset the color.

```

3244      \ifvoid\eqe@nskeyflsplit\else\PackageWarning{eqexam}
3245      {Some material from a solution environment does not\MessageBreak
3246      appear. Increase space allotted}{\normalcolor
3247      \llap{\smash{\rule{5pt}{5pt}}}}\aftergroup
3248      \reset@color\fi % dpsj24
3249      \fi

```

We conclude by adding some vertical space to better align the `workarea` environment. When a fill type is used, the space created by not be exactly *<dimen>* the dimension specified.

```

3250      \setlength{\@tempdima}{\vfGo@1-\depthtodate}\vskip\@tempdima
3251      % \vskip\flfboxrule\vskip3pt
3252      \dimen0=\sameVspace\relax
3253      \dimen2=\fboxrule\advance\dimen0by\dimen2
3254      \xdef\sameVspace{\the\dimen0 }\endgroup
3255      \global\flfrstsplitttrue
3256  }

```

`\eq@linesXPgs` Does the same as `\vspaceFillerLines` but is used instead of that command when the optional argument for `solution` specified a value for `nLines`. Code comments are the same as those just given for `\vspaceFillerLines`.

```

3257 \def\eq@linesXPgs{\@tempdimb\wlVspace % dpsj20
3258 \advance\@tempdimb-\flfboxrule
3259 \@tempdimb=\soln@keys@nLines\@tempdimb
3260 \advance\@tempdimb\flfboxrule
3261 \expandafter\vspaceFillerLines\expandafter{\the\@tempdimb}}

\eqe@wrtLineKernal, based on a case analysis, sets the register \eqetmplengha
and makes some horizontal glue adjustments as well.

3262 %\def\eq@linesXPgs{\@tempdimb\wlVspace
3263 % \@tempdimb=\soln@keys@nLines\@tempdimb
3264 % \expandafter\vspaceFillerLines\expandafter{\the\@tempdimb}}
3265 \def\eqe@wrtLineKernal{\let\eqe@x\relax\let\eqe@y\relax % dpsj9
3266 \ifx\solutionparshape\@empty % not lead-in
3267 \if@eqalignfilllinestoleft % align on left
3268 \setlength{\eqetmplengtha}{\labelwidth+\eqemargin}%
3269 \setlength{\eqetmplengtha}
3270 {\eqetmplengtha+\eqetmplengthb}%
3271 \ifwithinparts
3272 \ifx\istabularexer\eq@YES % tabular mode
3273 \setlength{\eqetmplengtha}{\prtsIndntSep}% dps27
3274 \eq@tmplengthA\eqetmplengtha
3275 \def\eqe@x{\hglue-\prtsIndntSep}% dps27
3276 \ifKV@eqefillLines@outlineonly\let\eqe@y\eqe@x\fi
3277 \else
3278 \settowidth{\eqetmplengtha}{\eqe@prtsepPrb}%
3279 \setlength{\eqetmplengtha}{\eqetmplengtha+\widthOfParts}% dps9
3280 \setlength{\eq@tmplengthA}{\eqetmplengtha}% dps9
3281 \def\eqe@x{\hspace*{-\eqetmplengtha}\hglue-\flfboxrule}% dps27
3282 \fi
3283 \else % not parts, still align left
3284 \def\eqe@x{\hspace*{-\eqemargin}}%
3285 \eqetmplengtha\eqemargin
3286 \eq@tmplengthA\eqemargin
3287 \fi
3288 \else % not lead-in, not align on left
3289 \ifwithinparts
3290 \setlength{\eqetmplengtha}{\labelwidth}%
3291 \setlength{\eqetmplengtha}
3292 {\eqetmplengtha+\eqetmplengthb}%
3293 \ifx\istabularexer\eq@YES % tabular mode
3294 \eqetmplengtha0pt
3295 \eq@tmplengthA\eqetmplengtha % dps27
3296 \else %dps2
3297 \def\eqe@x{\hglue-\flfboxrule}% dps4
3298 \eqetmplengtha0pt
3299 \eq@tmplengthA\eqetmplengtha
3300 \fi
3301 \else % not parts, not align left
3302 \eqetmplengtha0pt
3303 \eq@tmplengthA\eqetmplengtha

```

```

3304     \fi
3305     \fi
3306   \else % lead-in
3307     \if@eqalignnfilllinestoleft
3308       \setlength{\eq@tmplengtha}{\leadinIndentPrtSep}% dps27
3309       \eq@tmplengthA\eq@tmplengtha
3310     \else
3311       \eq@tmplengtha0pt
3312       \eq@tmplengthA\eq@tmplengtha
3313     \fi
3314   \fi
3315 }

\equalCellSizesOn set the switch \ifforceEqualCells to true, while \equalCellSizesOff sets the
\equalCellSizesOff same switch to false. Additionally, the latter command resets \fillLinesLine-
Width to its default value of \@empty. It is therefore important not to use the
switch directly, but to use these two convenience commands.

3316 \newif\ifforceEqualCells
3317 \def\equalCellSizesOn{\global\let\flEqu@lLineWidth\@empty
3318   \global\forceEqualCellstrue}
3319 \def\equalCellSizesOff{\global\let\flEqu@lLineWidth\@empty
3320   \global\forceEqualCellsfalse}
3321 \equalCellSizesOff

\forceEqu@lCells determines the value of \fillLinesLineWidth. It successively
adds \wlVspace until it exceeds \linewidth, it then subtracts off one \wlVspace
to get the value for \fillLinesLineWidth.

3322 \def\forceEqu@lCells{\bgroup
3323   \ifforceEqualCells
3324     \dimen6\wlVspace % reduced from \vspaceFillerLines
3325     \advance\dimen6\flfboxrule
3326     \dimen2=\z@
3327     \dimen4\linewidth
3328     \advance\dimen4by-\flfboxrule
3329     \if@eqalignnfilllinestoleft % dps8
3330       \advance\dimen4\eq@tmplengthA
3331     \fi
3332     \@tempcnta0
3333     \@whiledim \dimen2 < \dimen4 \do{%
3334       \advance\@tempcnta1
3335       \advance\dimen2by\dimen6
3336     }%
3337     \advance\@tempcnta-1
3338     \advance\dimen2-\dimen6
3339     \advance\dimen2\flfboxrule
3340     \advance\dimen2by-\eq@tmplengthA
3341     \xdef\flEqu@lLineWidth{\the\dimen2}%
3342   \fi
3343 \egroup}
3344 \let\eq@insert@more@content\relax

```

3345 \def\gobtodot#1.#2\@nil{\def\intPrt{#1}\def\fracPrt{#2}}

\eqe@makeVgrid A command that is executed when the document author expands \fillTypeGrid. The command itself draws vertical rule lines, which intersect with horizontal rule lines to form a grid. When \fillTypeGrid is active, \makeVgrid is \let to \eqe@makeVgrid. \makeVgrid appears twice in each of the macros \vspaceFillerLines and \eq@linesXPgs. The first instance of \makeVgrid draws the vertical line, assuming there is no page break. If there is a page break, the second instance of \makeVgrid continues to draw vertical lines with the correct height and number. To make the grid break across a page, the low level command \makeRoomForProb was modified; now prior and post actions (\priorPNPAction and \priorPNPAction) can be defined.

The \dbMrk is used to debug this portion of the code, view the debugging marks by setting \ifeqedb to true.

3346 \def\dbMrk#1{\textsf{\footnotesize#1}}

3347 \@ifundefined{ifeqedb}{\newif\ifeqedb \eqedbfalse}{}

\fillerCustomBg{<alt-bg>} Add a custom background feature, only available when the key bgonly is active. The argument <alt-bg> uses #1 and #2 to refer to the width and height

\resetFillerCustomBg of the work area. Use \resetFillerCustomBg to reset the method of coloring the background to its default method.

3348 \def\fillerCustomBg#1{\def\fillerBgIm@ge##1##2{#1}}

3349 \def\resetFillerCustomBg{\let\fillerBgIm@ge\@empty}

3350 \resetFillerCustomBg

\gridIndentAdj{<dimen>} is an adjustment in the horizontal positioning of the vertical lines of the grid. Under normal conditions, \gridIndentAdj is not needed. If the exam environment is embedded in environments that change some basic parameters, the alignment of the grid may not be right, in which case you can use \gridIndentAdj to shift the grid horizontally.

3351 \def\gridIndentAdj#1{\def\@rgi{#1}\ifx\@rgi\@empty

3352 \let\gridIndent@dj\relax\else % dps4

3353 \setlength{\@tempdima}{#1}\edef\gridIndent@dj{\the\@tempdima}\fi}

3354 \let\gridIndent@dj\relax

3355 \def\eqe@makeVgrid{\begingroup

3356 \@tempcnta=\fl@nRows\relax

3357 \advance\wlVspace\flfboxrule % dps25

3358 \setlength{\@tempdima}{\eqe@fillwidth}%

3359 \setlength{\@tempdimb}{\wlVspace}%

3360 \edef\eqe@tmp{\strip@pt\@tempdimb}%

3361 \expandafter\gobtodot\eqe@tmp.\@nil

3362 \divide\@tempdima by\intPrt\relax

3363 \dimen2=\@tempdima

3364 \dimen4=1pt

3365 \advance\dimen2 by 0.5\dimen4

3366 \divide\dimen2 by \dimen4

3367 \multiply\dimen2 by \dimen4

3368 \edef\eqe@tmp{\strip@pt\dimen2}%

3369 \expandafter\gobtodot\eqe@tmp.\@nil

```

3370 \advance\@tempcnta-1\relax % dps21
3371 \multiply\@tempdimb\@tempcnta
3372 \edef\eqe@HOfVrule{\the\@tempdimb}% dps7
3373 \ifKV@eqefillLines@topline % dps21
3374 \advance\@tempdimb\wlVspace
3375 \advance\@tempdimb\flfboxrule
3376 \edef\eqe@HOfVruleFrstRow{\the\@tempdimb}%
3377 \def\eqe@setVRule{\vbox
3378 to\eqe@HOfVruleFrstRow\relax{\gridVLineFill}}}% dps8
3379 \else
3380 \def\eqe@setVRule{\vbox
3381 to\eqe@HOfVrule\relax{\gridVLineFill}}}% dps8
3382 \fi
3383 \eqe@tempcnta=0
3384 \dimen2=\wlVspace
3385 \count4=\intPrt\relax
3386 \setlength{\dimen4}{\eqe@fillwidth}%
3387 \multiply\dimen2by\count4
3388 \@whiledim \dimen2 > \dimen4 \do{%
3389 \advance\count4by-1\relax
3390 \edef\intPrt{\the\count4 }%
3391 \dimen2\wlVspace
3392 \multiply\dimen2by\count4
3393 }%
3394 \dimen2\wlVspace
3395 \if@ealignfillllinestoleft % dps27
3396 \let\fl@oSmash\smash\let\fl@iSmash\relax
3397 \else
3398 \let\fl@iSmash\smash\let\fl@oSmash\relax
3399 \fi
3400 \fl@oSmash{%
3401 \makebox[0pt][l]{\makebox[\eqe@fillwidth][l]{\fl@iSmash}% dps27
3402 \ifx\gridIndent@dj\relax\else
3403 \hglue\gridIndent@dj\relax\fi
3404 \eqe@y\ifeqedb\rlap{\dbMrk{D}}\fi % dpsj9 \eqe@y (fudge)
3405 \rlap{\ifeqedb\rlap{\dbMrk{B}}}\fi
3406 \ifwriteVertic@lFLines % dps16
3407 \setlength{\@tempdima}{\eqe@fillwidth}%
3408 \ifKV@eqefillLines@bgonly % dps11
3409 \ifx\fillerBgIm@ge\empty
3410 \ifx\eqe@BGColor\empty\else
3411 \rlap{\textcolor{\eqe@BGColor}
3412 {\rule{\eqe@fillwidth}{\eqe@HOfVruleFrstRow}}}\fi
3413 \else
3414 \rlap{\fillerBgIm@ge{\eqe@fillwidth}{\eqe@HOfVruleFrstRow}}}\fi
3415 \else
3416 \dimen2=\wlVspace
3417 \dimen0=0pt
3418 \@whilenum\eqe@tempcnta<\count4\do{\rlap{\hglue\dimen0
3419 \ifKV@eqefillLines@outlineonly %dps10

```

```

3420         \ifnum\eqe@tempcnta=0 \else\let\gridVLineFill\vfill\fi\fi
3421         \textcolor{\eq@WriteLineColor}{\eqe@setVRule % dps27
3422         \ifeqedb\rlap{\advance\eqe@tempcnta by1\relax\scriptsize
3423         \raisebox{1pt}{\kern1pt\the\eqe@tempcnta}}\fi}}%
3424         \advance\dimen0by\dimen2
3425         \advance\eqe@tempcnta by1
3426     }% dps27
3427 \fi % dps11
3428 \advance\dimen2by\flfboxrule
3429 \ifKV@eqefillLines@bgonly\else % dps11
3430     \rlap{\hglue\@tempdima
3431     \textcolor{\eq@WriteLineColor}{\eqe@setVRule
3432     \ifeqedb\rlap{\thinspace\dbMrk{R}}\fi}}%
3433 \ifKV@eqefillLines@outlineonly %dps10
3434     \makebox[Opt][l]{\makebox[\eqe@fillwidth]{\eqWriteLine}}%dps10
3435 \fi
3436 \ifKV@eqefillLines@topline% dps7
3437     \ifgridpgbrk\@tempdima\eqe@H0fVruleFrstRow\relax % dps21
3438     \raise\@tempdima\hbox{\makebox[Opt][l]{%
3439     \makebox[\eqe@fillwidth]{\eqWriteLine}}}%
3440     \global\gridpgbrkfalse
3441 \fi
3442 \fi
3443 \fi
3444 \ifx\eqefillLines@outlineonlystar\@empty\else % dps26
3445     \global\let\eqefillLines@outlineonlystar\@empty
3446     \global\KV@eqefillLines@outlineonlyfalse
3447 \fi
3448 \ifx\eqefillLines@bgonlystar\@empty\else % dps26
3449     \global\let\eqefillLines@bgonlystar\@empty
3450     \global\KV@eqefillLines@bgonlyfalse
3451 \fi
3452 \fi % ifwriteVertic@lFLines
3453 \eqe@insert@more@content
3454 }% rlap
3455 }% smash (fl@iSmash)
3456 }% makeboxes and \fl@oSmash
3457 \endgroup
3458 }% \eqe@makeVgrid

```

In preparation for the `flexextended` option, we define several commands and one environment that emit a package warning if the `flexextended` option is not taken.

```

3459 \def\turnfl@nskeyMsg{\PackageWarning{eqexam}
3460 {You must first take the flexextended\MessageBreak
3461 option for this command to have any\MessageBreak effect}}
3462 \def\turnfl@nskeyOn{\turnfl@nskeyMsg}
3463 \def\priorw@Msg{\PackageWarningNoLine{eqexam}
3464 {The priorworkarea environment does nothing\MessageBreak
3465 without the flexextended option. All such content\MessageBreak in these
3466 environments are absorbed}}\global\let\priorw@Msg\@empty}

```

```

3467 \newenvironment{priorworkarea}{\priorw@Msg
3468     \setbox\eqe@nskeyflsplit\ vbox\bgroup}
3469     {\egroup\setbox\eqe@nskeyflsplit\box\voidb@x}
3470 \let\turnflanskeyOff\turnflanskeyOn
3471 \newif\iffirstsplit \firstsplittrue
3472 \newif\ifeqe@flosolns \eqe@flosolnsfalse
3473 \def\turnflosolnsOn{\eqe@flosolnstrue\turnflanskeyOn}
3474 \def\turnflosolnsOff{\eqe@flosolnsfalse\turnflanskeyOn}
3475 %\let\turnflosolnsOn\turnflanskeyOn
3476 %\let\turnflosolnsOff\turnflanskeyOn
3477 \newdimen\flfboxrule \flfboxrule=.4pt
3478 \def\tweakBreakPoint#1{\def\@rgi{#1}\ifx\@rgi\@empty
3479     \gdef\twe@kBre@kPoint{0pt}\else
3480     {\setlength{\@tempdima}{#1}%
3481     \xdef\twe@kBre@kPoint{\the\@tempdima}}\fi}
3482 \def\twe@kBre@kPoint{0pt}

```

When flexextended option is not loaded, we make a minimal definition of `\eq@b@ddCodeSpecial`. This redefinition is needed when `bgonly` is in effect.

```

3483 \def\eq@b@ddCodeSpecial#1{\if@eqlinedfiller
3484     \ifKV@eqefillLines@bgonly\leavevmode
3485     \vskip-\baselineskip\kern\lineskip\fi\fi#1}

```

`\flSeparateCutNames` For debug, or whatever reason, `\flSeparateCutNames` gives each flx cut file a distinct name.

```

3486 \def\flSeparateCutNames{\def\fl@CutName{flx\fl@Cnt-\jobname.cut}}
3487 </package>
3488 <*flexextended>

```

## 14.2 Utility commands used with the flexextended option

```

3489 \def\fls@vebaselinelineskip{% dps22
3490     \xdef\flbaselineskip{\the\baselineskip}}
3491 \def\fl@Cnt{0}
3492 \def\fl@CutName{flx-\jobname.cut}
3493 \def\flwriteexsol@fter{% dpsj23
3494     \ifx\exsolafter\@empty
3495         \immediate\write\verbatim@out{\string\ignorespaces}% dpsj23
3496     \else
3497         \immediate\write\verbatim@out{\string\exsolafter
3498         \string\space\string\ignorespaces}%
3499     \fi
3500 }
3501 \def\fl@getcontent#1{\begingroup
3502     {\count0=\fl@Cnt\relax
3503     \advance\count0by1\relax
3504     \xdef\fl@Cnt{\the\count0 }}%
3505 % \let\save@wlog\wlog\let\wlog\@gobble
3506 % \let\save@message\message\let\message\@gobble
3507 \global\let\verbatim@out\CommentStream

```

```

3508 \immediate\openout\verbatim@out=#1
3509 \ifwithinparts
3510   \ifx\istabularexer\eq@YES
3511     \immediate\write\verbatim@out{\string\hfuzz\string\hsize}%
3512     \immediate\write\verbatim@out{\string\minipage[t]{\string\hsize}}%
3513     \ifanswerkey
3514       \immediate\write\verbatim@out{\string\expandafter
3515         \string\noindent\string\eqSolnExCmds}% dps23
3516       \flwriteexsol@fter
3517     \else
3518       \immediate\write\verbatim@out{\string\expandafter
3519         \string\noindent\string\priorWorkAreaCmds
3520         \string\ignorespaces}% dps23
3521     \fi
3522   \else
3523     \ifanswerkey\flwriteexsol@fter\fi
3524   \fi
3525 \else
3526   \ifanswerkey\flwriteexsol@fter\fi
3527 \fi
3528 \verbatimwrite
3529 }
3530 \def\fl@vsplitandplace#1{%
3531   \iff@firstpass
3532   \global\setbox\eqe@nskeyflsplit\color@vbox
3533     \normalbaselines
3534     \predisplaypenalty=-50
3535     \postdisplaypenalty=-50
3536     \setlength{\hsize}{\eqe@fillwidth}%
3537     \expandafter\noindent % dpsj23
3538     \ifanswerkey\expandafter\eqSolnExCmds\else
3539       \expandafter\priorWorkAreaCmds\fi
3540     \ignorespaces\input{#1}\fls@vebaselinelineskip
3541     \color@endbox\global\fl@firstpassfalse\fi
3542   \@tempdima\eqe@H0fVrule\relax

```

I'm getting a black box when there is one line left on the second page and there is less than one line of text. The following three lines attempt to get that last line to appear when there is enough room for it.

```

3543 \advance\@tempdima\flfboxrule
3544 \advance\@tempdima\lineskip
3545 \iff@firstsplit\else\advance\@tempdima\wVspace\fi
3546 \advance\@tempdima\maxdepth
3547 \advance\@tempdima by\twe@kBre@kPoint\relax
3548 \eq@tmplengthB\@tempdima % target height
3549 \splittopskip\wVspace
3550 \splitmaxdepth\maxdepth
3551 \advance\splitmaxdepth by\twe@kBre@kPoint\relax
3552 \setbox\eq@pointbox=\copy\eqe@nskeyflsplit
3553 \setbox\eqe@nskeyfltop=\vsplit\eq@pointbox to \@tempdima

```



```

3554 \setbox\eqe@nskeyfltop=\vbox{\unvbox\eqe@nskeyfltop}%
3555 \ifvoid\eq@pointbox\else
3556   \setlength{\@tempdimb}{
3557     {\eq@tmplengthB-\dp\eqe@nskeyfltop}%
3558     \ifdim\@tempdimb<0pt
3559       \@tempdimb=-\@tempdimb
3560       \ifdim\@tempdimb>\maxdepth
3561         \advance\@tempdima-\flbaselineskip\relax
3562       \else
3563         \fi
3564     \else
3565       \setbox\eq@pointbox=\copy\eqe@nskeyflsplit
3566       \advance\@tempdima by\flbaselineskip\relax
3567       \setbox\eqe@nskeyfltop=\vsplit\eq@pointbox to \@tempdima
3568       \setbox\eqe@nskeyfltop=\vtop{\unvbox\eqe@nskeyfltop}%
3569       \setlength{\@tempdimb}{\eq@tmplengthB-\dp\eqe@nskeyfltop}%
3570       \ifdim\@tempdimb<0pt % too much
3571         \advance\@tempdima-\flbaselineskip\relax
3572       \else
3573         \fi
3574       \fi
3575     \fi
3576     \splittopskip\wlVspace
3577     \lineskip1pt
3578     \setbox\eq@pointbox=\copy\eqe@nskeyflsplit
3579     \global\setbox\eqe@nskeyfltop=\vsplit\eqe@nskeyflsplit to \@tempdima
3580     \global\setbox\eqe@nskeyfltop\vtop{\unvbox\eqe@nskeyfltop\vskip0pt}%
3581     \ifvoid\eqe@nskeyflsplit
3582       \ifeqedb\llap{Bot}\fi
3583       \ifeqedb\raise\eqe@H0fVrule\llap{Top}\fi
3584       \ifeqedb\llap{V\qquad}\fi
3585       \ifflrstsplit\global\flrstsplitfalse
3586       \ifeqedb\llap{FP\hskip.5in}\fi
3587       \rlap{\raisebox{\eqe@H0fVrule+\flfboxrule+\lineskip}{
3588         \vtop{\vsize=\eqe@H0fVrule\relax
3589           \unvbox\eqe@nskeyfltop\vfil}}}\else
3590       \ifeqedb\llap{SP\hskip.5in}\fi
3591       \rlap{\raisebox{\eqe@H0fVrule+\flfboxrule+\wlVspace+\lineskip}{
3592         \vtop{\vsize=\eqe@H0fVrule\relax
3593           \unvbox\eq@pointbox\vfil}}}%
3594       \global\setbox\eqe@nskeyflsplit\box\voidb@x
3595     \fi
3596   \else
3597     \ifeqedb\llap{Bot}\fi
3598     \ifeqedb\raise\eqe@H0fVrule\llap{Top}\fi
3599     \ifeqedb\llap{nV\qquad}\fi
3600     \ifflrstsplit\global\flrstsplitfalse
3601     \ifeqedb\llap{FP\hskip.5in}\fi
3602     \rlap{\raisebox{\eqe@H0fVrule+\flfboxrule+\lineskip}{
3603       \vtop{\vsize=\eqe@H0fVrule\relax

```

```

3604      \unvbox\eqe@nskeyfltop\vfil}}}\else
3605      \ifeqedb\llap{SP\hskip.5in}\fi
3606      \rlap{\raisebox{\eqe@H0fVrule+\flfboxrule+\wlVspace+\lineskip}
3607      {\vtop{\vsize=\eqe@H0fVrule\relax

```

To have the filler lines/grid with prior and solution content to space more than one page, we unbox `\eqe@anskeyfltop` but do not void it.

```

3608      \unvbox\eqe@nskeyfltop\vfil}}}%
3609 %      \unvbox\eq@pointbox\vfil}}}%
3610 %      \global\setbox\eqe@nskeyflsplit\box\voidb@x
3611 \fi
3612 \fi
3613 \setbox\eq@pointbox\box\voidb@x
3614 % \gdef\twe@kBre@kPoint{0pt}% dpsj21
3615 }

```

### 14.3 Filler lines with answerkey option

When using the `flexextended` option and the `\turnflanskeyOn` command, we always use the grid filler type. In the case of the basic filler types of line, dash, dots, and blank, the code threads passes to `\makeVgrid`, but in the case of the `answerkey` option, `\makeVgrid` writes no vertical lines. Hopefully, the spacing between the statement of the question and the solution are then consistent.

```

3616 \let\turnfl@nskeyOnOff\relax

```

`\turnflanskeyOn` The `\turnflanskeyOn` command turns *on* the feature of superimposing the solution over the filler lines. Does nothing unless the `answerkey` option is in force.

```

3617 \def\turnflnosolnsOn{\eqe@flnosolnstrue}
3618 \def\turnflnosolnsOff{\eqe@flnosolnsfalse}
3619 \def\turnflanskeyOn{%
3620   \def\turnfl@nskeyOnOff{%
3621     \ifanswerkey
3622       \ifx\makeVgrid\relax % already in grid mode, do nothing
3623       \ifwriteVertic@lFLines\KV@eqefilllLines@toplinefalse\fi % dps16
3624       \writeVertic@lFLinesfalse
3625       \let\makeVgrid\eqe@makeVgrid % dpsj5
3626     \fi % dps17
3627     \let\eqe@insert@more@content\eqe@insertSolns % dpsj5
3628     \eqe@flexextendedtrue % dps16
3629     \fillTypeGrid % dps16
3630     \global\let\eq@insertverticalspace\eq@YES
3631     \global\vspacewithsolnstrue
3632   \fi
3633 }%
3634 }

```

`\turnflanskeyOff` The `\turnflanskeyOff` command turns *off* the feature of superimposing the solution over the filler lines. Does nothing unless the `answerkey` option is in force.

```

3635 \def\turnflanskeyOff{%
3636   \def\turnfl@nskeyOnOff{%
3637     \ifanswerkey

```

```

3638     \writeVertic@lFLinestrue
3639     \eqe@fextendedfalse
3640     \let\eq@insertverticalspace\eq@NO
3641     \@eqlinedfillerfalse
3642     \global\let\eqe@insert@more@content\relax
3643     \global\let\eq@insertverticalspace\eq@YES
3644     \ifvspacewithsolns\else
3645     \global\vspacewithsolnsfalse\fi
3646     \global\displayworkareafalse
3647   \fi
3648 }%
3649 }
3650 \turnflanskeyOff

3651 \let\p@ssToSolns\relax % dps30
3652 \def\eqe@insertSolns{\clubpenalty=50
3653   \widowpenalty=50 \vbadness=10000
3654   \fl@vsplitandplace{s\fl@CutName}}
3655 \def\eq@b@ddCodeSpecial#1{\global\gridpgbrkfalse % dpsj15
3656   \global\let\eq@e@ddCodeSpecial\relax
3657   \gdef\eqe@flnexti{#1}% dpsj5
3658   \if@eqlinedfiller
3659     \ifKV@eqefilllLines@bgonly\leavevmode
3660       \vskip-\baselineskip\kern\lineskip\fi % dpsj11
3661     \ifKV@eqefilllLines@outlineonly\leavevmode
3662       \vskip-\baselineskip\kern\lineskip\fi
3663     \ifanswerkey %\turnfl@nskeyOnOff % dps16
3664       \ifeqe@fextended
3665         \ifcont@nnot
3666           \ifx\makeVgrid\relax
3667             \gdef\p@ssToFLs{\turnfl@nskeyOnOff
3668               \let\eqe@insert@more@content\eqe@insertSolns
3669               \writeVertic@lFLinesfalse
3670               \KV@eqefilllLines@toplinefalse
3671               \let\makeVgrid\eqe@makeVgrid
3672             }%
3673           \else
3674             \gdef\p@ssToFLs{\turnfl@nskeyOnOff
3675               \let\eqe@insert@more@content\eqe@insertSolns
3676               \writeVertic@lFLinestrue}%
3677           \fi
3678           \gdef\eqe@flnexti{\turnfl@nskeyOnOff
3679             \global\fl@firstpasstrue % dpsj5
3680             \p@ssToSolns
3681             \global\gridpgbrkfalse
3682             \eqe@setStartSolns

```

We are with a good two deep, but we need to pass `\vspacewithkeyOff`, so we save its current state and declare a global call to `\vspacewithkeyOff`. The state is restored by `\eq@e@ddCodeSpecial`.

```

3683         \xdef\kdvsp@SAVE{\ifkeepdeclaredvspacing
3684             true\else false\fi}%
3685         \gdef\kdvsp@Restore{\@nameuse
3686             {keepdeclaredvspacing\kdvsp@SAVE}}%
3687         \gdef\twe@kBre@kPoint{Opt}\global
3688         \let\kdvsp@Restore\relax}% dpsj21
3689     \expandafter\global\vspacewithkey0ff
3690     \vbadness=10000
3691     \fl@getcontent{s\fl@CutName}}}%
3692     \gdef\eq@e@ddCodeSpecial{\endverbatimwrite\endgroup
3693 %       \kdvsp@Restore % dpsj20
3694     \ifwithinparts\ifx\istabularexer\eq@YES
3695     \immediate\write\verbatim@out{\string\endminipage}\fi\fi
3696     \immediate\closeout\verbatim@out
3697     \global\let\eq@e@ddCodeSpecial\relax
3698 }%
3699 \else
3700     \PackageWarningNoLine{eqexam}{For solutions to appear
3701         using filler lines\MessageBreak
3702         (with flextexded and answerkey options),\MessageBreak
3703         you must first expand \string\turnContAnnotOn\MessageBreak
3704         prior to line \the\inputlineno.\MessageBreak
3705         Switching to \string\useFillerDefault}%
3706     \useFillerDefault
3707     \let\eq@insertverticalspace\eq@N0
3708 \fi
3709 \fi
3710 \fi
3711 \fi
3712 \eqe@flnexti
3713 }% dpsj4

```

## 14.4 Filler lines with nosolutions option

Just as we can supply filler lines for the `answerkey` option, we can do the same for the `nosolutions` option and superimpose content on the filler lines.

**priorworkarea** Place the `priorworkarea` environment *prior to* the solution environment. It has not arguments, but gets dimensions from the `solution` environment that follows.

```

3714 \def\eqe@priorw@content{%
3715     \clubpenalty=50 % dps15
3716     \widowpenalty=50
3717     \vbadness=10000
3718     \fl@vsplitandplace{p\fl@CutName}}%
3719 }%
3720 \renewenvironment{priorworkarea}{\par
3721     \ifx\solutionparshape\@empty\else % dps28
3722     \pushEnvir
3723     \everypar{}\if@e@alignfilllllinestoleft\else

```

```

3724         \parshape=1 \leadinIndent \linewidth\fi
3725     \popEnvir
3726 \fi
3727 \global\let\p@ssToFLs\relax
3728 \global\let\p@ssToSolns\relax
3729 \vspacewithkeyOff
3730 \ifeq@nosolutions\else\sloppy\fi
3731 \def\eqe@flnexti{\eqSavedComment}%
3732 \ifeqe@flnosolns\ifdisplayworkarea
3733     \def\eqe@flnexti{\fl@getContent{p\fl@CutName}}\fi\fi % dpsj21
3734 \gdef\eqe@flnextii{\endeqSavedComment}%dps11
3735 \ifeqe@flnosolns\ifdisplayworkarea
3736     \gdef\eqe@flnextii{\endverbatimwrite\endgroup
3737     \ifwithinparts\ifx\istabularexer\eq@YES
3738         \immediate\write\verbatim@out{\string\endminipage}\fi\fi
3739     \immediate\closeout\verbatim@out}\fi\fi
3740 \eqe@flnexti
3741 }{f%
3742 \eqe@flnextii
3743 \ifcont@nnot
3744     \ifeqe@flnosolns
3745         \ifdisplayworkarea
3746             \ifx\eq@insertverticalspace\eqe@YES
3747                 \ifx\makeVgrid\relax
3748                     \gdef\p@ssToFLs{%
3749                         \let\eqe@insert@more@content\eqe@priorw@content
3750                         \writeVertic@lFLinesfalse
3751                         \KV@eqefillLines@toplinefalse
3752                         \let\makeVgrid\eqe@makeVgrid}%
3753                 \else
3754                     \gdef\p@ssToFLs{%
3755                         \let\eqe@insert@more@content\eqe@priorw@content
3756                         \writeVertic@lFLinestrue}%
3757                 \fi
3758                 \gdef\kdvsp@Restore{\gdef\twe@kBre@kPoint{Opt}\global
3759                 \let\kdvsp@Restore\relax}% dpsj21
3760             \fi
3761         \fi
3762     \else
3763         \global\setbox\eqe@nskeyflsplit\box\voidb@x
3764         \global\setbox\eqe@nskeyfltop\box\voidb@x
3765     \fi
3766 \else
3767     \PackageWarningNoLine{eqexam}{For priorworkarea to appear
3768     using filler lines\MessageBreak
3769     (with flextextded and nosolutions options),\MessageBreak
3770     you must first expand \string\turnContAnnotOn\MessageBreak
3771     prior to line \the\inputlineno.\MessageBreak
3772     Switching to \string\useFillerDefault}%
3773     \gdef\p@ssToSolns{\useFillerDefault}%

```

```

3774 \fi
3775 }

3776 % end of segment
3777 </flexextended>
3778 <*package>

Set the eqexam page style, if not otherwise indicated.

3779 \ifeqfortextbook\else
3780     \if@bypasseqexamheading\else
3781 \pagestyle{eqExamheadings}\fi\fi

3782 </package>
3783 <*textbook>
3784 \ProvidesFile{eqtextb.def}
3785 [2016/01/18 v4.6 Cmds used by the fortextbook option (dps)]

```

## 15 Concerning the fortextbook option

What are my goals/desired features? Modern (U.S.) textbooks—at least the ones I’m familiar with—consist of some or all of the following resources:

- **Student Edition:** Answers to odd-numbered problems appear in the back of the text.

We need to have a scheme where odd-numbered problems, under suitable options, are compiled. **Goal:** It does not need to be restricted to odd-numbered, however, need to latex only those problems that meet the “include” criteria.

For chapter review problem sets, odd-number problems are have solutions in the back of the book.

For chapter quizzes, odd-numbered (optionally all) solutions are in the back of the book.

- **Instructor Edition:** Answers to all problems appear in the back of the book. Answers may also appear in the body of the text, in the margins of the text, or immediately after the statement of the problem. If the answer is too long, there is a cross-reference to the solution in the appendix.

Some publishers I’ve seen have wide margins where additional material can be inserted (historical sketches, instructor notes, pictures, etc.). In these margins, the answers to the problems can appear. Other publishers put answer immediately following the questions. The latter is easy to do; just have a macro, say `\ANS{$12.5$}`, which only expands when the “instructor” option is used.

- **Student Solution Manual:** Contains solutions to all odd-problems, as well as any review problems and chapter quizzes. Some publishers include all solutions to chapter quizzes.

A solution manual is a separate publication. This document would be created by latexing one or more of the auxiliary file (`.sol`) These files might have to be edited before the final compile. We include only the solutions that meet the include criteria (i.e., odd-numbered ones).

The current features of `eqexam` is what is needed here. The authors need only include solutions to each problem in a `solution` environment. Now, I realize that often times the authors create the solutions, but someone else, possibly a grade student or contractor, solves the problems. In the latter case, the authors would probably not like to turn over the source files to the one solving and typesetting the problems.

- **Instructor Solution Manual:** Contains solutions to all problems, review problems, chapter quizzes.

Similar comments for the instructor solution manual.

Some other thoughts by a contributor:

- Often there is a diagram or graphic within the problem – this has some figure caption and after the running counter of the figure it is named the NUMBER of the problem (cross-reference to the problem number)
- Often the probs with soln are setup in two-column style.
- Often the PROBLEM NUMBER has a special formatting (not only bold and black), maybe with a `colorframebox` around or some special formatting from the author
- Of course I have seen in some EXAMPLES that there is a wide margin to put in additional graphics etc. setup in two-side style – wide left margin on even pages, wide right margins on odd pages. Here as well are captions setup and cross-references.

## 15.1 Setting options with `\textbookOpts`

```
\textbookOpts We set up a command for setting the options for the fortextbook option.
marginans 3786 \define@boolkey{eqe@tbopts}[is]{instred}[true]{}
inlineans 3787 \define@boolkey{eqe@tbopts}[is]{studented}[true]{}
marginsonleft 3788 \ifisstudented
ssols Now let's try to filter out the even-numbered problems for the student edition.
lsols 3789 \tbfilterOutEvenNums
The above command is normally \let to \@gobble.
3790 \fi
3791 }
```

This code is executed in `\exambegdef`, the start up code of the `exam` environment. This enables problems with fill-ins, true/false, or multiple choice, to have the answer appear in the space provided.

```

3792 \def\tb@beginexam@code{%
3793     \ifisinstred\answerkeytrue\eq@proofingtrue\fi}
    \eqEXT{\thequestionnoi} and \endeqEXT\tok1\tok2 enclose each solution,
    \tbfilterOutEvenNums redefines \eqEXT to gobble everything, when the page
    number is even, through \endeqEXT and the two tokens it follows. This leaves
    only the odd-numbered problems.
3794 \newcommand{\tbfilterOutEvenNums}{%
    exerquiz changed \eqEXT to two variables, so we make the same change here
3795     \def\eqEXT##1##2{\ifodd##1\let\eqe@next\relax\else
3796     \def\eqe@next{\gobbletoEndEXT}\fi\eqe@next}%
3797 }
3798 \newcommand{\tballowAllNums}{%
    exerquiz changed \eqEXT to two variables, so we make the same change here
3799 %     \let\eqEXT\@gobble
3800     \let\eqEXT\@gobbletwo
3801     \let\endeqEXT\relax
3802 }
3803 \define@boolkey{eqe@tbopts}[is]{marginans}[true]{}
3804 \define@boolkey{eqe@tbopts}[is]{inlineans}[true]{}
3805 \define@boolkey{eqe@tbopts}[]{marginsonleft}[true]{}
    If margins are always on left, we turn off switching of margin notes as placed by
    \marginpar, and use \reversemarginpar to get them on the left.
3806     {\@mparswitchfalse\reversemarginpar}
3807 \define@boolkey{eqe@tbopts}[show]{ssols}[true]{}
3808 \define@boolkey{eqe@tbopts}[show]{lsols}[true]{}
3809     \ifshowlsols\let\tb@soln@choice\tb@showlsols\fi}
    The default settings are true for studented and false for instred.
3810 \newcommand{\textbookOpts}[1]{\setkeys{eqe@tbopts}{#1}%
    We do not allow both instred and studented to be true.
3811     \ifisinstred\global\isstudentedfalse\else
3812         \ifisstudented\global\isinstredfalse
3813     \fi\fi
    Added this part in in case \textbookOpts comes after \marparboxwidth.
3814     \ifdim\tbmarparboxwidth=1sp\else
3815     \expandafter\tbMakeFinalCalcs\fi
3816 }
    As mentioned above, the default settings are true for studented and false for
    instred.
3817 \isstudentedtrue
3818 \isinstredfalse
3819 \ismarginansfalse
3820 \isinlineansfalse

```



The command is available only in the preamble.

```
3821 \onlypreamble{\textbookOpts}
```

`\turnOffMarAnsOnAnsInline` These three command may not be useful in the creation of a textbook, but you  
`\turnOnMarAnsOffAnsInline` never know, I used them in my demo doc `fortextbook.tex` to turn off and on  
`\toggleInstrAns` the display of the answers (change margin to inline, change inline to margin, and  
toggle margin and inline).

```
3822 \newcommand{\turnOffMarAnsOnAnsInline}{%
3823   \global\ismarginansfalse\global\isinlineanstrue
3824   \insMidMarg{\global\ismarginansfalse
3825     \global\isinlineanstrue}%
3826 }
3827 \newcommand{\turnOnMarAnsOffAnsInline}{%
3828   \global\ismarginanstrue\global\isinlineansfalse
3829   \insMidMarg{\global\ismarginanstrue
3830     \global\isinlineansfalse}%
3831 }
3832 \newcommand{\toggleInstrAns}{%
3833   \ifisinstred\ifismarginans
3834     \global\ismarginansfalse\global\isinlineanstrue
3835     \insMidMarg{\global\ismarginansfalse
3836       \global\isinlineanstrue}%
3837   \else
3838     \global\ismarginanstrue\global\isinlineansfalse
3839     \insMidMarg{\global\ismarginanstrue
3840       \global\isinlineansfalse}%
3841   \fi\fi
3842 }
```

## 15.2 Macros to display answers/shortsolns

In this section, we develop some commands to display answers or short solutions. These would appear if `instred=true`, in-line, or in the margins.

`\ANS` Let us begin by creating a simple macro for saving an answer. The answer is displayed “in-line.” No verbatim-type text allowed, no unbalanced braces unless escaped. `\ANS` displays the answer if the `instred` option of the `eqe@tbopts` family, i.e., by executing

`\bGrpANS` Two macros used to group answers in the margins.

```
\eGrpANS 3843 \newif\ifWithinANSGrp\WithinANSGrpfalse
3844 \newif\ifftb@isANSListOpen\ftb@isANSListOpenfalse
3845 \newcommand{\bGrpANS}{%
3846   \if\probstar*\else
3847     \PackageError{eqexam}{Use of \string\bGrpANS\space
3848       only applies\MessageBreak to the problem* environment}{Please
3849       remove this \string\bGrpANS.}%
3850   \fi
3851   \ifWithinANSGrp
```

```

3852      \global\WithinANSGrpfalse
3853      \let\tb@next\relax
3854      \PackageError{eqexam}{\string\bGrpANS\space already open}
3855      {You issued an earlier \string\bGrpANS,
3856       but did not close it.}%
3857    \else
3858      \global\WithinANSGrptrue
3859      \global\ftb@isANSListOpenfalse
3860      \def\tb@next{\ANS}%
3861    \fi
3862    \tb@next
3863  }
3864  \newcommand{\eGrpANS}{%
3865    \if\probstar*\else
3866      \PackageError{eqexam}{Use of \string\eGrpANS\space
3867       only applies\MessageBreak to the problem* environment}{Please
3868       remove this \string\eGrpANS.}%
3869    \fi
3870    \ifWithinANSGrp
3871      \global\WithinANSGrpfalse
3872      \def\tb@next{\ANS}%
3873    \else
3874      \let\tb@next\relax
3875      \PackageError{eqexam}{\string\eGrpANS\space already closed}
3876      {You've issued two consecutive \string\eGrpANS\space
3877       commands,\MessageBreak either remove this one
3878       or the previous one.}%
3879    \fi
3880    \tb@next
3881  }

```

\ANS begin by checking to see if there is a star that follows the command, this is used for inline answers. If \* is present, we do not put the answer inline, but will put it in the margins if the option call for it.

```

3882 \newcommand{\ANS}{\@ifstar{\let\tb@istart=1\tb@ANS}
3883   {\let\tb@istart=0\tb@ANS}}

```

(10/13/2011) The following is the original definition of \tb@ANS before the creation of the commands \bGrpANS and \bGrpANS. We keep this to revert to this definition if this new feature causes problems.

```

\newcommand{\tb@ANS}[1]{%
  \ifinstred
    \ifisinlineans\if\tb@istart0\ANSFmt{\thequestionnoi}{#1}\fi\fi
    \ifismarginans
      \edef\eqe@prehold{\noexpand\par\kern0pt\noindent
        \if\probstar*%
          \noexpand\begin{eqeList}[\tb@wparts@len]{%
            \noexpand\eqedsplyOnlyFrst{\thequestionnoi}%
            {\thepartno}\noexpand\eqe@hspannerMrg

```

```

\noexpand\makebox[\noexpand\tbmrpartwidth]%
{\noexpand\tb@mrPartFmt{\thepartno}}}%
\else
\noexpand\begin{eqeList}%
{\noexpand\tb@mrDigitFmt{%
\theeqquestionnoi\eqe@decPointMrg}}%
\fi
}\expandafter\insMidMrg%
\expandafter{\eqe@prehold#1\end{eqeList}}%
\fi
}
}

\ftb@defineInsSpan is used when there is an optional argument for \ANS. It formats the range of parts,
for example, (a)–(c). This macro can be redefined, I suppose, to meet the needs
of the author.
3884 \def\ftb@defineInsSpan#1{\def\ftb@argi{#1}\ifx\ftb@argi\@empty
3885 \def\ftb@InsSpan{}\else\ftb@spanPrts{#1}%
3886 \def\ftb@InsSpan{\noexpand\hspace{-\labelsep}}%
3887 \noexpand\textcolor{MRGPARTcolor}{--}\noexpand
3888 \makebox[\noexpand\tbmrpartwidth]{\noexpand
3889 \tb@mrPartFmt{\ftb@EndSpanPrts}}\eqe@hspannerMrg}\fi
3890 }

\ftb@spanPrts calculates the letter of the end of the range. #1 is passed by \ANS (\tb@ANS,
actually). For example if we have \ANS[2]{...}, #1=2.
3891 \def\ftb@spanPrts#1{ {%
3892 \advance\value{partno}by#1\relax
3893 \xdef\ftb@EndSpanPrts{\thepartno}}%
3894 }

\ftb@EqeListPrtsFmt is the internal formatting used within the eqeList for the part letter.
3895 \def\ftb@EqeListPrtsFmt{\noexpand
3896 \makebox[\noexpand\tbmrpartwidth]{\noexpand
3897 \tb@mrPartFmt{\thepartno}}\nobreak
3898 }

\ftb@OpenEqeListPrts opens an eqeList environment, and displays the question number (optionally)
and the part number.
3899 \def\ftb@OpenEqeListPrts{\noexpand
3900 \begin{eqeList}[\tb@wparts@len]{\noexpand
3901 \eqedsplyOnlyFrst{\theeqquestionnoi}%
3902 {\thepartno}\noexpand\eqe@hspannerMrg\ftb@EqeListPrtsFmt}%
3903 }

\ftb@CloseEqeList closes the eqeList after inserting \qe@prehold and the content, #1.
3904 \def\ftb@CloseEqeList#1{\expandafter\insMidMrg%
3905 \expandafter{\eqe@prehold#1\end{eqeList}}%
3906 }

\grpANSDelimiter delimits the parts when \bGrpANS/\eGrpANS is used. May be redefined.
3907 \newcommand{\grpANSDelimiter}{\textcolor{MRGPARTcolor}{,}\space}

```

`\tb@ANS` does the main work of `\ANS`.

```

3908 \newcommand{\tb@ANS}[2][{}]{%
3909     \ifisinstred
3910         \ifisinlineans
3911             \if\tb@istart0\ANSFmt{\theeqquestionnoi}{#2}\fi
3912         \fi
3913     \ifismarginans
3914         \ftb@defineInsSpan{#1}%

```

We create the code that we will introduce into `\insMidMarg`, this will be introduced prior to #2.

```

3915         \edef\eqe@prehold{%
3916             \if\probstar%

```

If this question is one with parts...

```

3917             \ifftb@isANSListOpen

```

If the list is already open (`\ifftb@isANSListOpen`), we just add content to the `eqeList` environment.

```

3918                 \ftb@EqeListPrtsFmt\noexpand\eqe@hspannerMrg
3919             \else

```

If the list is not open, we start the `eqeList` environment in the usual way, this also includes the case where `\bGrpAns` is not used, which is normally the case.

```

3920                 \noexpand\par\kern0pt\noindent
3921                 \ftb@OpenEqeListPrts\ftb@InsSpan
3922             \fi
3923         \else

```

This is a question without parts.

```

3924                 \noexpand\begin{eqeList}%
3925                 {\noexpand\tb@mrgDigitFmt{%
3926                 \theeqquestionnoi\eqe@decPointMrg}}}%
3927             \fi
3928         }%

```

We have finished constructing `\eqe@prehold`. We next set `\ftb@isANSListOpen` to true, if `\WithinANSGrp` is true.

```

3929         \ifWithinANSGrp\global\ftb@isANSListOpenttrue\fi

```

If we are within an open group, we emit `\insMidMarg` with the `\eqe@prehold`, followed by #2, and a comma-space combo, but *we do not close* the `eqeList` environment.

```

3930         \ifWithinANSGrp
3931             \expandafter\insMidMarg\expandafter
3932             {\eqe@prehold#2\grpANSDelimiter}%
3933         \else

```

This is the normal case, we insert `\eqe@prehold`, #2, and close the `eqeList` environment.

```

3934             \expandafter\insMidMarg%

```

```

3935             \expandafter{\eqe@prehold#2\end{eqeList}}%
3936             \global\ftb@isANSListOpenfalse
3937         \fi
3938     \fi % \ifismarginans
3939 \fi %\ifisinstred
3940 }

```

End (10/13)

**eqeList** An environment used to format the answers in the margins, when **marginans** is in effect.

```

3941 \newenvironment{eqeList}[2][\tb@woparts@len]{\begin{list}{#2}{%
3942     \def\argi{#1}\setlength{\labelwidth}{#1}%
3943     \ifx\argi\tb@woparts@len
3944     \settowidth{\labelsep}{\eqe@prtsepMrg}\else
3945     \settowidth{\labelsep}{\eqe@hspannerMrg}\fi
3946     \setlength{\leftmargin}{\labelwidth+\labelsep}%
3947     \setlength{\parskip}{0pt}\setlength{\partopsep}{0pt}%
3948     \setlength{\topsep}{1pt}\setlength{\parsep}{0pt}%
3949     \setlength{\itemindent}{0pt}\setlength{\itemsep}{3pt}%
3950 }\item\relax}{\end{list}}

```

## Formatting Answers and Solutions

**\mrgDigitFmt** Format of the digit (and the decimal point) for the answers in the margins.

```

\mrgDigitFmt{\textbf{#1}}
\mrgPartFmt{\textbf{(\hfil#1\hfil)}}
\setMarIndents[\bfseries\normalsize\normalfont]{00}{(d)}

3951 \newcommand{\mrgDigitFmt}[1]{\def\tb@mrgDigitFmt##1{#1}}
3952 \mrgDigitFmt{#1}

```

**\mrgPartFmt** Format of the part (including possibly the parentheses), example give above.

```

3953 \definecolor{MRGPARTcolor}{named}{black}
3954 \newcommand{\mrgPartFmt}[1]{\def\tb@mrgPartFmt##1{#1}}
3955 \mrgPartFmt{\textcolor{MRGPARTcolor}{(\hfil#1\hfil)}}

```

**ANScolor** The default color of the answers that appear in the margins or inline.

```

3956 \definecolor{ANScolor}{rgb}{0,0,.8}

```

**\ANSFmt** The command that sets the format, may be redefined as needed. Used in the **\ANS** command above.

```

3957 \newcommand{\ANSFmt}[2]{\textcolor{ANScolor}{#2}}

```

We have two environments that we use in three different situations:

- **eqequations** environment: (1) Used to control the display of the **probset** environment within the body of the textbook; (2) used to control the display of the solutions “in the back of the book.”

- `eqlist` environment: Used for displaying answers in the margin of the book, when the appropriate options allow it.

We want to be able to manipulate some of the parameters of these three situation, independently of each other. There are several issues, setting what I have been calling the gutter width, and the display of the problem numbers.

We define four commands for each of the three situations described above. The names have a pattern to them, and similarly named commands have the same use.

The numbering of the problems has the pattern: `dd.□(a)□`. We provide convenience commands to give these internal macros values

```

\prbDecPt Basic parameters for the problems in the body of the text.
\prbPrtsep 3958 \def\eqe@decPointPrb{.} % decimal point of prob number
\prbNumPrtsep 3959 \def\eqe@prtsepPrb{\ } % prob with parts, space after part
3960 \def\eqe@hspannerPrb{\ } % space after prob number
3961 \providecommand{\prbDecPt}[1]{\def\eqe@decPointPrb{#1}}
3962 \providecommand{\prbPrtsep}[1]{\def\eqe@prtsepPrb{#1}}
3963 \providecommand{\exPrtsep}[1]{\def\eqe@prtsepPrb{#1}}
3964 \providecommand{\prbNumPrtsep}[1]{\def\eqe@hspannerPrb{#1}}

\solDecPt Basic parameters for the problems in the solution sets.
\solPrtsep 3965 \def\eqedecPointSoln{.} % decimal point of prob number
\solNumPrtsep 3966 \def\eqe@prtsepSoln{\ } % prob with parts, space after part
3967 \def\eqe@hspannerSoln{\ } % space after prob number
3968 \newcommand{\solDecPt}[1]{\def\eqedecPointSoln{#1}}
3969 \newcommand{\solPrtsep}[1]{\def\eqe@prtsepSoln{#1}}
3970 \providecommand{\solNumPrtsep}[1]{\def\eqe@hspannerSoln{#1}}

\mrgDecPt Basic parameters for the problems in the margins.
\mrgPrtsep 3971 \def\eqe@decPointMrg{.} % decimal point of prob number
\mrgNumPrtsep 3972 \def\eqe@prtsepMrg{\ } % prob with parts, space after part
3973 \def\eqe@hspannerMrg{\ } % space after prob number
3974 \newcommand{\mrgDecPt}[1]{\def\eqe@decPointMrg{#1}}
3975 \newcommand{\mrgPrtsep}[1]{\def\eqe@prtsepMrg{#1}}
3976 \newcommand{\mrgNumPrtsep}[1]{\def\eqe@hspannerMrg{#1}}

\setMarIndents Sets some dimensions used by the eqlist environment. \tb@woparts@len is
calculated and is used as the default gutter width in eqlist. \tb@wparts@len
is used for the gutter width for the gutter width when there is a problem with
parts. Finally, \tbmrgpartwidth in \ANS and is used for the width of a \makebox
that enclosed the part letter.
3977 \newcommand{\setMarIndents}[3][\normalsize\normalfont]{\%
3978 \settowidth{\@tempdima}{#1#2\eqe@decPointMrg}%
3979 \xdef\tb@woparts@len{\the\@tempdima}%
3980 \settowidth{\@tempdima}%
3981 {#1#2\eqe@decPointMrg\eqe@hspannerMrg#3}%
3982 \xdef\tb@wparts@len{\the\@tempdima}%
3983 \settowidth{\@tempdima}{#1#3}%

```

```

3984 \xdef\tbmrpartwidth{\the\@tempdima}%
3985 }}
3986 \setMarIndents{00}{(d)}

```

**\setSolnIndent** Used to set the some parameters used by **equequestions**, in the solutions file.

```

3987 \newdimen\solnGutter
3988 \newcommand{\setSolnIndent}[3][\normalsize\normalfont\bfseries]{%
3989   {\settowidth{\@tempdima}{#1#2\eqedecPointSoln\eqe@hspannerSoln}%
3990   \global\solnGutter\@tempdima
3991   \settowidth{\@tempdima}{#1#3}%
3992   \xdef\tbsolnpartwidth{\the\@tempdima}%
3993 }}
3994 \setSolnIndent{00}{(d)}

```

**\setSolnMargins** This command is written to the solution file, and expanded when that file is input back in. If **\solnGutter** is not **Opt**, we set the length of **\eqemargin** using the current value; otherwise, we use the value determined by **\setSolnIndent**, above.

```

3995 \renewcommand{\setSolnMargins}[1]{%
3996   \ifdim\solnGutter=Opt \setlength\eqemargin{#1}\else
3997     \setlength\eqemargin{\solnGutter}\fi\ignorespaces}
3998 %\newcommand{\defaultSolnIndent}{\gdef\solnGutter{Opt}}
3999 \newcommand{\defaultSolnIndent}{\global\solnGutter=Opt}
4000 \defaultSolnIndent

```

**\prbNumFmt** We redefine **\exlabelformat**, **\exsllabelformat**, and **\exsllabelformatwp**.  
**\solWoPrtsFmt** They are defined in such a way as to simply their modification through a series of  
**\solWPrtsFmt** simple formatting commands. The defaults are

```

\prbNumFmt{\textbf{#1}}
\solWoPrtsFmt{\textbf{#1}}
\solWPrtsFmt{\textbf{#1}}{(\hfil#2\hfil)}

4001 \renewcommand{\exlabelformat}{%
4002   \tbprbNumFmt{\theeqquestionnoi\eqe@decPointPrb}}

```

**\prbNumFmt** is the format for the number of the problems in the body of the text.  
The argument **#1** is a symbolic argument for the question number.

```

4003 \newcommand{\prbNumFmt}[1]{\def\tbprbNumFmt##1{#1\eqe@hspannerPrb}}
4004 \prbNumFmt{\textbf{#1}}

```

Redefine **\exsllabelformat**, and **\exsllabelformatwp**

```

4005 \renewcommand{\exsllabelformat}{\string\tbsolWoPrtsFmt{%
4006   \theeqquestionnoi\string\eqedecPointSoln}}
4007 \renewcommand{\exsllabelformatwp}{\string\tbsolWPrtsFmt%
4008   {\string\eqedsplyOnlyFrst{\theeqquestionnoi}{\thepartno}}%
4009   {\thepartno}}%
4010 }

```

**\solWoPrtsFmt** is the format for the number of the problems in the solution set.  
The argument **#1** is a symbolic argument for the question number. **\solWPrtsFmt**

is the format for a problem with parts in the solution file. #1 is symbolically the question number, and #2 is a symbolic for the part letter.

```

4011 \newcommand{\solWoPrtsFmt}[1]{\def\tbsolWoPrtsFmt##1{%
4012   \makebox[Opt][r]{#1\eqe@hspannerSoln}}
4013 \solWoPrtsFmt{\textbf{#1}}
4014 \newcommand{\solWPrtsFmt}[2]{\def\tbsolWPrtsFmt##1##2{%
4015   \makebox[Opt][r]{#1\eqe@hspannerSoln}%
4016   \makebox[\tbsolnpartwidth][l]{#2}\eqe@prtsepSoln%
4017 }}
4018 \solWPrtsFmt{\textbf{#1}}{(\hfil#2\hfil)}

```

An alternate definition for `\solWPrtsFmt`, used by `\hangSolWPrtsFmt`.

```

4019 \newcommand{\solWPrtsFmt@hang}[2]{%
4020   \def\tbsolWPrtsFmt##1##2{%
4021     \makebox[Opt][r]{#1\eqe@prtsepSoln%
4022     \makebox[\tbsolnpartwidth][l]{#2}\eqe@hspannerSoln}%
4023 }}

```

`\hangSolWPrtsFmt` The command takes two arguments, the same as `\solWPrtsFmt`. When this command is executed in the preamble, we get hanging indentation for problems with parts.

```

4024 \let\bpartsmrk\relax
4025 \let\epartsmrk\relax
4026 \newcommand{\hangSolWPrtsFmt}[2]{%

```

At the beginning and ending of a parts environment, we begin and end a special `eqepartsquestions` environment, designed to give the desired indentation.

```

4027   \def\prior@parts@hook{%
4028     \ifisleadin\else
4029       \writeT@SolnFile{^^J\protect\bpartsmrk}\fi}%
4030   \def\post@parts@hook{%
4031     \writeT@SolnFile{\protect\epartsmrk^^J}}%
4032   \def\bpartsmrk{\global\firstitemtrue\begin{eqepartsquestions}}%
4033   \def\epartsmrk{\end{eqepartsquestions}\global\firstitemfalse}%

```

We must also redefine `\solWPrtsFmt` by letting it to `\solWPrtsFmt@hang`, then executing it using the parameters passed.

```

4034   \let\solWPrtsFmt\solWPrtsFmt@hang
4035   \solWPrtsFmt{#1}{#2}%
4036 }
4037 %\@onlypreamble\hangSolWPrtsFmt

```

Define some switches, token registers, and boxes for managing the answers and marginal notes.

```

4038 \newif\ifexamenv \examenvfalse
4039 \newif\iffirstemit \firstemittrue
4040 \newtoks\txtbkt@ks \txtbkt@ks={}
4041 \newtoks\txtbkt@ksi \txtbkt@ksi={}
4042 \newbox\txtbkb@xb@t
4043 \newbox\txtbkb@xt@p

```



```

4044 \newbox\txbktb@xh@ld
4045 \let\tbTopMargin\relax
4046 \let\tbBotMargin\relax
4047 \long\def\tb@addtoTopMargin#1{\txbkt@ksi={#1}%
4048   \edef\eqe@tmphold{\the\txbkt@ksi\the\txbkt@ks}%
4049   \global\txbkt@ks=\expandafter{\eqe@tmphold}%
4050 }
4051 \newcommand{\tb@addtoMargin}[1]{%
4052   \edef\eqe@tmphold{\the\txbkt@ks}%
4053   \global\txbkt@ks=\expandafter{\eqe@tmphold#1}%
4054 }

```

As my first attempt, let's create two comment environments to be used within the solution environment.

```

\begin{solution}
\begin{ssol}
  <short solution/answer>
\end{ssol}
\begin{lsol}
  <long solution>
\end{lsol}
\end{solution}

```

The control of these environments are made through

```

lsol  Place full (or long) solutions in this environment.
\tb@showlsols 4055 \newcommand{\tb@showlsols}{\includecomment{lsol}\excludecomment{ssol}}

ssol  Place short solutions in this environment.
\tb@showssols 4056 \newcommand{\tb@showssols}{\includecomment{ssol}\excludecomment{lsol}}

```

The default is to show the short solutions.

```

4057 \let\tb@soln@choice\tb@showssols
4058 %\let\tb@sols@choice\tb@showssols

```

### 15.3 Marginal Matter.

There are three levels in the margins:

1. **Top level:** This is a command `\tbTopMargin` with may be redefined between pages. It should have the width of the `\parbox` that contains all the content of the margin, this width is `\tb@marparboxwidth` (`\oddsidemargin-\marginparsep`). The content *must* be unbreakable across pages. The content of `\tbTopMargin` will appear *on every page subsequent to its definition*.

```

\clearTopMargin  We can clear the top level using the following command

4059   \newcommand{\clearTopMargin}{\global\let\tbTopMargin\relax}

```

Clearing will take effect on the following page.

`\setTopMargin` As a convenience macro, we can create top margin content. Redefinitions will appear on the next page from where the definition was made.

```
4060 \newcommand{\setTopMargin}[1]{%
4061   \gdef\tbTopMargin{\color@begingroup\normalcolor
4062     #1\color@endgroup}%
4063   \gdef\tbSaveTopMargin{\color@begingroup\normalcolor
4064     #1\color@endgroup}}
4065 \let\tbTopMargin\@empty
4066 \let\tbSaveTopMargin\@empty
```

2. **Middle level:** The middle level is the most interesting. You write to it using `\insMidMarg`. Normally, this is text. If there is too much text, it will be split off and placed in the middle level of the next page. The command `\ANS` also writes to the middle level when the `instred` and `marginans` options are taken.

3. **Bottom level:** This is similar to the top level, but on the bottom. The command is named `\tbBotMargin` and follows the same rules as `\tbTopMargin`. Again, the content of `\tbBotMargin` will appear *on every page subsequent to its definition*.

`\clearBotMargin` We can clear the bottom level using the following command

```
4067 \newcommand{\clearBotMargin}{\global\let\tbBotMargin\relax}
```

Clearing will take effect on the following page.

`\setBotMargin` As a convenience macro, we can create bottom margin content. Redefinitions will appear on the next page from where the definition was made.

```
4068 \newcommand{\setBotMargin}[1]{\gdef\tbBotMargin{#1}%
4069   \gdef\tbSaveBotMargin{#1}}
4070 \let\tbBotMargin\@empty
4071 \let\tbSaveBotMargin\@empty

4072 \newcommand{\restoreLastTopMargin}{\expandafter\setTopMargin
4073   \expandafter{\tbSaveTopMargin}}
4074 \newcommand{\restoreLastBotMargin}{\expandafter\setBotMargin
4075   \expandafter{\tbSaveBotMargin}}
```

`\insMidMarg` `\insMidMarg` is a `\parbox` that will hold the material in the margin.

`\MarParBoxFmt` The formatting for the marginal `\parbox`

```
4076 \newcommand{\MarParBoxFmt}{\normalsfcodes
4077   \normalfont\normalsize\normalbaselines\parindent0pt
4078   \vbadness\@Mi \hbadness5000 \tolerance9999
4079   \parskip0pt\raggedright \spaceskip=0pt\xspaceskip=0pt
4080   \setlength{\linewidth}{\tbmarparboxwidth}%
4081 }
```

`\tbmarparboxwidth` The width of the margin box. Initial value of `1sp`, if the user does not reset the value, it is a package error.

```

4082 \newlength\tbmarparboxwidth
4083 \setlength\tbmarparboxwidth{1sp}

```

`MidMargcolor` The default color of text of the middle level

```

4084 \definecolor{MidMargcolor}{rgb}{0,0,.8}
4085 \newcommand{\midMargFmt}[1]{%
4086     \def\tb@midMargFmt{\normalfont\normalsize\normalcolor#1}}
4087 \midMargFmt{\color{MidMargcolor}}

```

`\eqe@MarParBox` This is the actual `\parbox` that holds the marginal material. I have two versions of this box, the first one has height `\textheight`, the second one has height `\textheight+\footskip`.

```

4088 %\def\eqe@MarParBox#1{\parbox[b][\textheight][t]{%
4089 %    {\tbmarparboxwidth}{\color{MidMargcolor}#1}}
4090 \def\eqe@MarParBox#1{\lower\footskip\hbox{%
4091     \leavevmode\parbox[b][\textheight+\footskip][t]{%
4092         {\tbmarparboxwidth}{\tb@midMargFmt#1}}}%

```

Finally, we get to the `\insMidMarg`, this is used to write to the middle level.

```

4093 \newcommand{\insMidMarg}[1]{%
4094     \let\eqe@margininsert\@empty
4095     \expandafter\tb@addtoMargin\expandafter{\eqe@margininsert#1}%
4096 }

```

`\tbPreMarginHeader` Executed prior to the marginal heading

`\tbPostMarginHeader` Executed after the marginal heading

`HEADERcolor` Default color of a marginal header

`\cngMargHeadColorTo` Changes the marginal header to a named color

`\resetMargHeadColor` Reset the marginal header color to the default, `HEADERcolor`

`\tbMarginHeaderFmt` Formatting for a marginal header. Format the marginal header, the default is `HEADERcolor` in bold

```

4097 \newcommand{\tbPreMarginHeader}{\par\penalty0 \kern3pt}
4098 \newcommand{\tbPostMarginHeader}{\par\nobreak}
4099 \definecolor{HEADERcolor}{named}{black}
4100 \newcommand{\cngMargHeadColorTo}[1]{\insMidMarg{\gdef\tb@MHC{#1}}}
4101 \newcommand{\resetMargHeadColor}{\insMidMarg{\gdef\tb@MHC{HEADERcolor}}}
4102 \resetMargHeadColor
4103 \newcommand{\tbMarginHeaderFmt}[1]{\textcolor{\tb@MHC}{\textbf{#1}}}

```

`\insMargHead` Used to insert a general marginal heading into the middle level. The optional parameter allows you to set a mark.

`\insProbHead` Used to insert a marginal heading for a problem set into the middle level. The optional parameter allows you to insert a mark, the default mark is `#1 \tbcontinued`.

```

4104 \newcommand{\insMargHead}[2][]{\% dps
4105   \insMidMarg{\tb@marginHeader{#1}{#2}}
4106 \newcommand{\insProbHead}[2][]{\%
4107   \def\tb@argi{#1}\ifx\tb@argi\@empty
4108     \protected@xdef\currProbHead{#2 \tbcontinued}\else
4109     \protected@xdef\currProbHead{#1}\fi
4110   \ifisinstred\ifismarginans
4111     \insMidMarg{\tb@marginProbHeader{#1}{#2}}\fi\fi
4112 }
4113 \newcommand{\tb@marginProbHeader}[2]{\%
4114   \def\tb@argi{#1}\ifx\tb@argi\@empty
4115     \tb@marginHeader{#2 \tbcontinued}{#2}\else
4116     \tb@marginHeader{#1}{#2}\fi
4117 }
4118 \newcommand{\tb@marginHeader}[2]{\tbPreMarginHeader
4119   \tbMarginHeaderFmt{#2}\def\tb@argi{#1}\ifx\tb@argi\@empty
4120   \mark{#2}\else\mark{#1}\fi\tbPostMarginHeader
4121 }
```

`\tbcontinued` The continue annot that appear when a problem set flows over to the next page.

```

4122 \newcommand{\tbcontinued}{(cont.)}
```

`\tbplaceMargins` Redefine this macro to set the locations of the margins we are writing to.

```

4123 \newcommand{\marparboxwidth}[1]{\%
4124   \setlength\tbmarparboxwidth{#1}\%
4125   \setlength{\marginparwidth}{\tbmarparboxwidth}\%
4126   \tbMakeFinalCalcs
4127 }
4128 \@onlypreamble\marparboxwidth
4129 \newcommand{\chkmarginboxwidth}{\%
4130   \ifdim\tbmarparboxwidth=1sp \PackageError{eqexam}\%
4131   {You have not set the value of\MessageBreak
4132     \string\marparboxwidth}\%
4133   {Define the \string\marparboxwidth\space command}\fi
4134 }
```

`\ifmarginsonleft` A Boolean switch, if true, all margins are on the left; otherwise, they alternate

```

4135 \newif\ifmarginsonleft \marginsonleftfalse
```

`\tbSetupForMargins` We compute `\oddsidemargin`, `\evensidemargin`, and `\textwidth`

```

4136 \newcommand{\tbSetupForMargins}{\%
4137   \ifmarginsonleft
4138     \setlength{\oddsidemargin}{\tbmarparboxwidth+\marginparsep}\%
4139     \setlength{\evensidemargin}{\oddsidemargin}\%
4140     \setlength{\textwidth}{\paperwidth-2in-\oddsidemargin}\%
4141   \else
```

```

4142      \setlength{\oddsidemargin}{0pt}%
4143      \setlength{\evensidemargin}{\tbmarparboxwidth+\marginparsep}%
4144      \setlength{\textwidth}{%
4145          \paperwidth-2in-\oddsidemargin-\evensidemargin}%
4146      \fi
4147 }

```

**\tbplaceMargins** We calculate the coordinates of the lower left hand corner of the margin **\parbox** depending on the value of **\ifmarginsonleft**.

```

4148 \newcommand{\tbplaceMargins}{%
4149     \setlength{\@tempdima}{%
4150         \paperheight-1in-\topmargin-\headheight-\headsep-\textheight}%
4151     \xdef\@evenlly{\strip@pt\@tempdima}%
4152     \xdef\@oddlly{\@evenlly}%
4153     \setlength{\@tempdima}{1in}%
4154     \xdef\@evenllx{\strip@pt\@tempdima}%
4155     \ifmarginsonleft\else
4156         \setlength{\@tempdima}{1in+\textwidth+\marginparsep}\fi
4157     \xdef\@oddllx{\strip@pt\@tempdima}%
4158 }

```

**\tbMakeFinalCalcs** Executed by **\marparboxwidth**

```

4159 \newcommand{\tbMakeFinalCalcs}{%
4160     \tbSetupForMargins
4161     \tbplaceMargins
4162 }

```

**\tbminskipbtnlayers** **\tbminskipbtnlayers** is the minimum skip between layers (top, middle, bottom)  
Executed by **\marparboxwidth**

```

4163 \newlength\tbminskipbtnlayers
4164 \setlength{\tbminskipbtnlayers}{6pt}

4165 \newif\ifiscarryover \iscarryoverfalse

```

**carryoverFmt** is a work-around for the color problem experienced with carry over text: Suppose there is a change of color of the text on the previous page, the carry over text will naturally be colored the default color, **MidMargcolor**. To continue the text with the same color as the one the previous page, we enclose the text in the **carryoverFmt** environment.

```

4166 \newenvironment{carryOverFmt}[1]{#1\let\tb@carryoverFmt\@empty
4167     \c@rryoverFmt{#1}}{}

```

This command is called by the **carryOverFmt** environment. it takes its argument, which is a change in color or style, and defines **\tb@carryoverFmt**, which will be executed on the next page.

```

4168 \def\c@rryoverFmt#1{%
4169     \ifx\tb@carryoverFmt\@empty
4170         \global\let\tb@carryoverFmt\@empty
4171     \xdef\tb@co@page{\thepage}%

```

```

4172      \gdef\tb@carryoverFmt{\ifnum\thepage>\tb@co@page\relax
4173      #1\global\let\tb@carryoverFmt\@empty\fi}%
4174      \fi
4175 }
4176 \let\tb@carryoverFmt\@empty

```

\tb@insertCarryOver takes its argument, that is always \unvbox\txtbkb@xb@t}, and if there is any carryover content, will insert its argument followed by a copy, \tb@rest@reMarginFmt of the default margin format. This seems to work for recovering from a change of text or style over a page boundary.

```

4177 \let\tb@rest@reMarginFmt\relax
4178 \def\tb@insertCarryOver#1{%
4179     \let\tb@rest@reMarginFmt\relax
4180     \ifiscarryover\ifx\tb@carryoverFmt\@empty\else
4181         \let\tb@rest@reMarginFmt\tb@midMargFmt
4182         \tb@carryoverFmt\fi\fi
4183     #1 \tb@rest@reMarginFmt
4184 }

```

\eqe@tb@shipout We define the shipout to the margins.

**Bug:** When I use `graphicxsp`, embed the picture (such as a logo), and use that picture as the `\setTopMargin`, the shipout routine is executed twice for each page. I haven't figured out what causes this, but here is a work around. We record the most recent page number, if it equals the page number of the last iteration of `\eqe@tb@shipout`, we do nothing; otherwise, execute the shipout code.

```

4185 \newif\iftb@shipoutPermitted \tb@shipoutPermittedtrue
4186 \newcommand{\turnOnFTBShipout}{\global\tb@shipoutPermittedtrue}
4187 \newcommand{\turnOffFTBShipout}{\global\tb@shipoutPermittedfalse}

```

\turnOnFTBShipout \turnOnFTBShipout turns on the shipout, the default, and \turnOffFTBShipout turns it off.

```

4188 \newcommand{\eqe@tb@shipout}{\iftb@shipoutPermitted
4189     \ifnum\arabic{page}=\tb@lastpageshipped
4190     \let\tb@so@next\relax\else
4191     \xdef\tb@lastpageshipped{\arabic{page}}%
4192     \def\tb@so@next{\eqe@tb@shipout}\fi
4193     \expandafter\tb@so@next\fi
4194 }
4195 \def\tb@lastpageshipped{-100}

```

Here is the actual shipout code for writing to the margins.

```

4196 \newcommand{\eqe@tb@shipout}{%
4197     \fboxsep=Opt\setlength{\unitlength}{1pt}%
4198     \global\setbox\txtbkb@xb@t=\vbox\bgroup
4199         \color@begingroup
4200         \hsize=\tb@marparboxwidth
4201         \vsize=\textheight
4202         \MarParBoxFmt
4203         \csname tbTopMargin\endcsname

```

```

4204         \vskip\tbminskipbtnlayers
4205 \set@typeset@protect
4206         \the\txtbkt@ks
4207         \color@endgroup\vfil
4208     \egroup
4209     \global\setbox\txtbkb@xt@p=\vsplit\txtbkb@xb@t to\textheight
4210     \ifvoid\txtbkb@xb@t\global\iscarryoverfalse
4211     \else\global\iscarryovertrue\fi

```

We have three levels the top (`\tbTopMargin`), the bottom (`\tbBotMargin`), and the middle (`\txtbkt@ks`). `\tbTopMargin` is no problem but `\tbBotMargin` requires some special attention.

```

4212     \ifx\tbBotMargin\relax\else

```

If `\tbBotMargin` is not `\relax`, we begin by putting `\tbBotMargin` into a `\vbox` under the same assumptions, and get its height.

```

4213         \bgroup\setbox2=\vbox{%
4214             \color@begingroup\normalcolor
4215             \hsize=\tbmarparboxwidth\kern0pt
4216             \MarParBoxFmt\csname tbBotMargin\endcsname
4217             \color@endgroup
4218             \kern0pt
4219         }%

```

We reduce `\textheight` by the height of `\tbBotMargin`

```

4220         \dimen0=\textheight
4221         \advance\dimen0-\ht\txtbkb@xh@ld
4222         \advance\dimen0-\tbminskipbtnlayers

```

We split off the top material by this amount, the new bottom is in `\txtbkb@xt@p` the new top is in `\box0`

```

4223         \setbox0=\vsplit\txtbkb@xt@p to \dimen0

```

The new bottom (which will overflow to the next page) is the content we clipped off bottom of `\txtbkb@xt@p` and the original overflow material still in `\txtbkb@xb@t`.

```

4224         \global\setbox\txtbkb@xb@t=\vbox{%
4225             \unvbox\txtbkb@xt@p\unvbox\txtbkb@xb@t}%

```

We then patch everything together the new top is in `\txtbkb@xt@p` the new top is in `\@tempboxa` followed by `\tbBotMargin` (in `\box\txtbkb@xh@ld`).

```

4226         \global\setbox\txtbkb@xt@p=\vbox{\unvbox0
4227             \vfill\vskip\tbminskipbtnlayers
4228             \vfil\unvbox2\relax}\egroup
4229     \fi
4230     \ifodd\value{page}%
4231         \put(\@oddllx,\@oddlly){%
4232             \eqe@MarParBox{\unvbox\txtbkb@xt@p}}\else
4233         \put(\@evenllx,\@evenlly){%
4234             \eqe@MarParBox{\unvbox\txtbkb@xt@p}}\fi

```

We see if there is any carry over, if yes, we insert into `\txtbkt@ks` for use on the next page, along with a heading, if any.

```
4235 \global\txtbkt@ks={}\ifvoid\txtbkb@xb@t\else
```

We test whether these is a `\splitbotmark`, if yes, then we will insert it at the top of the next page with formatting.

```
4236 \if!\splitbotmark!\global\let\tb@sbm@exp\relax\else
4237 \xdef\tb@sbm@exp{\noexpand\tbPreMarginHeader
4238 \noexpand\tbMarginHeaderFmt{\splitbotmark}}%
4239 \noexpand\tbPostMarginHeader
4240 \noexpand\par\kern3pt}%
4241 \fi
```

Here is the content that will be carried over to the next page, we insert a `\splitbotmark` if it is non-empty (`\tb@tmp@exp`).

```
4242 \global\txtbkt@ks=\expandafter{\tb@sbm@exp
4243 \tb@insertCarryOver{\unvbox\txtbkb@xb@t}}%
4244 \fi
4245 }
```

`\insertpageifcarryover` This macro is use to generate a blank page if there is carry over from the previous page. It is place just after the exercises, and before a new chapter of section. The optional argument allows you to insert something into the new page, if one is automatically created. The default is `\null`.

```
4246 \newcommand{\insertpageifcarryover}[1][\null]{%
```

We begin by starting a new page, the shipout routine of previous page will be initialized and can then get an accurate result for `\ifiscarryover`.

```
4247 \newpage
```

If there is carryover, we create a new page by inserting a content into the page. If there is no carry over, we do now insert any content, and the page will not be created.

```
4248 \ifiscarryover\def\eqeifnext{\csname iftrue\endcsname}%
4249 \PackageInfo{eqexam}{Carry over of content in margin
4250 from page \thepage.\MessageBreak Creating a blank page}\else
4251 \def\eqeifnext{\csname iffalse\endcsname}\fi\eqeifnext#1\fi}
```

`\setFullWidthHeader` Makes the running header full width.

```
4252 \newcommand{\setFullWidthHeader}{%
4253 \setlength{\@tempdima}{%
4254 \evensidemargin+\tbmarparboxwidth+\marginparsep}%
4255 \edef\@headoffset{\the\@tempdima}%
4256 \def\@evenhead{\makebox[0pt]{\makebox[0pt][l]
4257 {\thepage}\hspace{\@headoffset}}\hfil\slshape\leftmark}%
4258 \ifmarginsonleft
4259 \def\@oddhead{\makebox[0pt]{\makebox[0pt][l]
4260 {\slshape\rightmark}\hspace{\@headoffset}}\hfil\thepage}%
4261 \else
4262 \def\@oddhead{\{\slshape\rightmark}\hfil\makebox[0pt]
```



```

4263         {\hspace{\@headoffset}\makebox[0pt][r]{\thepage}}}%
4264     \fi
4265 }

```

## 15.4 In support of solutions at end of document and chapter

A feature that may not be used much is to have solutions at the end of each chapter.

`\chaptersolutions` If `\tb@EndOfChapterExercises` is executed, and `\chaptersolutions` is placed between chapters, we can generate solutions at the end of the chapters, instead of at the end of the book. `\chaptersolutions` is `\let` to `\relax` unless `\tb@EndOfChapterExercises` is executed. In this case `\chaptersolutions` inputs the the .sol file, then then opens it

```

4266 \newif\ifchapterexercises \chapterexercisesfalse
4267 \let\chaptersolutions\relax
4268 \def\tb@EndOfChapterExercises{%
4269     \let\include@solutions@chapter\include@solutions
4270     \def\includeexersolutions{%
4271         \include@solutions@chapter
4272         \global\let\include@solutions\relax
4273     }%

```

`\chaptersolutions` is redefined from `\relax`. Input current solutions, close stream, open stream.

```

4274     \def\chaptersolutions{%
4275         \includeexersolutions
4276         \immediate\closeout\ex@solns
4277         \newwrite \ex@solns \global\let\quiz@solns\ex@solns
4278         \immediate\openout \ex@solns \jobname.sol
4279         \ifvspacewithsolns\writeAllAnsAtEnd\fi
4280     }%
4281 }

```

```

4282 \def\writeallsolutions{\let\chaptersolutions\relax}

```

`\exercisesAtEndOfChapter` If you want solutions at the end of each chapter, you'll have to execute this command in the preamble. See `\initChapAfterSolns` for an example of usage.

`afterChapSolns` This comment environment is a convenience for placing content between chapters.

```

4283 \excludecomment{afterChapSolns}
4284 %\includecomment{solnsAtEnd}

```

(2014/05/08) There is a problem with nested comment environments when `solnsAtEnd` contains within it the `\includeexersolutions` command, especially when `multicols` is used. The fix seems to redefine things so that `solnsAtEnd` writes to a different CUT file

```

4285 \def\NewCommentCutFile{\def\CommentCutFile{solnsAtEnd.cut}}
4286 \def\RestoreCommentCutFile{\def\CommentCutFile{comment.cut}}
4287 \@ifundefined{BeforeIncludedComment}{%

```

```

4288 \long\def\solnsAtEndcomment
4289 #1{\message{Special comment '#1'}%
4290   \csarg\def{#1}{\endgroup \message{Processing '#1' comment.}%
4291     \NewCommentCutFile\SetUpCutFile
4292     % #2 before SetUp, so we can do renaming.
4293     \message{Comment '#1' writing to \CommentCutFile.}%
4294     \ProcessComment{#1}}%
4295   \csarg\def{After#1Comment}{\immediate\closeout\CommentStream
4296     \RestoreCommentCutFile\input{solnsAtEnd.cut}\relax}%
4297   \CommentEndDef{#1}}
4298 }\long\def\solnsAtEndcomment
4299 #1{\message{Special comment '#1'}%
4300   \csarg\def{After#1Comment}{\immediate\closeout\CommentStream
4301     \RestoreCommentCutFile\input{solnsAtEnd.cut}\relax}%
4302   \csarg\def{#1}{\NewCommentCutFile\BeforeIncludedComment\relax
4303     \ProcessComment{#1}}%
4304   \CommentEndDef{#1}}
4305 }
4306 \solnsAtEndcomment{solnsAtEnd}
4307 \newcommand{\exercisesAtEndOfChapter}{%
4308   \ifeq@nosolutions\else
4309     \typeout{^^J!!!!Executing in chapter solutions!!!!^^J}
4310     \chapterexercisetrue\tb@EndOfChapterExercises
4311     \ifchapterexercisetrue
4312       \csarg\let{solnsAtEnd}\@gobble
4313       \excludacomment{solnsAtEnd}%
4314       \csarg\let{AftersolnsAtEndComment}\relax
4315       \includecomment{afterChapSolns}\else
4316       \excludacomment{afterChapSolns}\fi
4317     \fi
4318 }
4319 \@onlypreamble\exercisesAtEndOfChapter

```

## 15.5 Modifying and restoring the Layout

The book may need a wide page format and use multi-columns to display homework sets, or solutions at the end if the book.

`\setFullWidthLayout` A command to set the page layout for the solutions in the back of the book. Typically, we do away with the wide margins. We also save the current values of the parameters we are changing so we can restore them later.

```

4320 \newcommand{\setFullWidthLayout}{%
4321   \saveBasicLayoutParams
4322   \setlength{\oddsidemargin}{0in}%
4323   \setlength{\evensidemargin}{\oddsidemargin}%
4324   \setlength{\textwidth}{\paperwidth-2in}%
4325   \setlength{\linewidth}{\paperwidth-2in}%
4326   \setlength{\columnseprule}{0pt}%
4327   \def\@evenhead{\thepage\hfil\slshape\leftmark}%

```

```

4328     \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
4329 }

```

`fullwidthtext` When `\setFullWidthLayout` is in effect, we have the problem of writing text. Originally, I used a `\parbox` with width of `\linewidth`, but this has it problems when breaking across pages. We have instead an environment for writing; the list environment obeys the current `\linewidth`, which is set to `\paperwidth-2in`, this latter value may not always be correct (especially when the margins are smaller than 2in).

```

4330 \newenvironment{fullwidthtext}{%
4331 \begin{list}{}{%
4332   \setlength{\labelwidth}{0pt}\setlength{\labelsep}{0pt}%
4333   \setlength{\itemindent}{0pt}\setlength{\itemsep}{0pt}%
4334   \setlength{\topsep}{0pt}\setlength{\parsep}{0pt}%
4335   \setlength{\listparindent}{\parindent}%
4336   \setlength{\leftmargin}{0pt}\setlength{\rightmargin}{0pt}
4337 }\item\relax}\end{list}}

```

`\restorePageLayout` Restore the last saved page parameters.

```

4338 \newcommand{\restorePageLayout}{\newpage
4339   \setlength{\oddsidemargin}{\tb@osms}
4340   \setlength{\evensidemargin}{\tb@esms}
4341   \setlength{\textwidth}{\tb@tws}
4342   \setlength{\linewidth}{\tb@lws}
4343   \setlength{\columnseprule}{\tb@csr}
4344 }

```

Used by `\setFullWidthLayout` just before the page layout parameters are changed.

```

4345 \newcommand{\saveBasicLayoutParams}{%
4346   \xdef\tb@osms{\the\oddsidemargin}%
4347   \xdef\tb@esms{\the\evensidemargin}%
4348   \xdef\tb@tws{\the\textwidth}%
4349   \xdef\tb@lws{\the\linewidth}%
4350   \xdef\tb@csr{\the\columnseprule}%
4351 }

```

`\initChapAfterSolns` Initializes the environment when solutions appear after each chapter. Example of usage, taken from `fortextbook.ltx`,

```

\begin{afterChapSolns}
\initChapAfterSolns
\section{Solutions to Chapter Exercises}
\begin{fullwidthtext}
We present short solutions to the problems.
We present short solutions to the problems.
We present short solutions to the problems.
We present short solutions to the problems.
\end{fullwidthtext}
\bigskip

```

```

\begin{multicols}{2}\forceNoColor
\chaptersolutions
\end{multicols}
\restoreFromChapAfterSolns
\end{afterChapSolns}

4352 \newcommand{\initChapAfterSolns}{\newpage
4353     \clearTopMargin\clearBotMargin
4354     \setFullWidthLayout
4355 }

```

`\restoreFromChapAfterSolns` Restores the saved parameters at the end of the chapter solutions, see above for an example.

```

4356 \newcommand{\restoreFromChapAfterSolns}{\newpage
4357     \restorePageLayout\setFullWidthHeader
4358 }

4359 % End of textbook segment
4360 </textbook>
4361 <*package>

```

## 15.6 We shipout in support of fortetbook

We shipout `\eqe@tb@shipout` to be placed in the margins on every page.

```

4362 \ifeqfortetbook
4363 \AtBeginDocument{\tb@soln@choice
4364     \ifeqwritetomargins\chkmarginboxwidth
4365     \AddToShipoutPicture{\eqe@tb@shipout}\fi}
4366 \fi
4367 </package>
4368 <*textbook>

```

## 15.7 Modify eqequestions environment

We adjust the `eqequestions` environment to minimize spacing between problems.

```

4369 \eqequestopsep{0pt}
4370 \eqequesparsep{0pt}
4371 \eqequesitemsep{0pt}
4372 \eqequeslistparindent{0pt}
4373 \renewenvironment{eqequestions}{%
4374     \begin{list}{}{
4375         \ifwithinsoldoc\let\solnItemMngt\eqeSolnItemMngt\fi
4376         \setlength{\labelwidth}{\eqemargin}%
4377         \setlength{\parsep}{\eqeques@parsep}%
4378         \setlength{\itemsep}{\eqeques@itemsep}
4379         \setlength{\topsep}{\eqeques@topsep}%
4380         \setlength{\itemindent}{0pt}%
4381         \setlength{\listparindent}{\eqeques@listparindent}%
4382         \ifwithinsoldoc\settowidth{\labelsep}{\eqe@hspannerSoln}\else

```

```

4383 \settowidth{\labelsep}{\eqe@hspannerPrb}\fi
4384 \setlength{\leftmargin}{\labelwidth}%
4385 }\ifwithinsoldoc\global\firstitemtrue\fi\item\relax}{\end{list}}

```

**eqepartsquestions** This environment is used in the SOL file with problems with parts to hang indent the solutions with parts.

```

4386 \newcommand{\eqepquestopsep}[1]{\def\eqepques@topsep{#1}}
4387 \newcommand{\eqepquesparsep}[1]{\def\eqepques@parsep{#1}}
4388 \newcommand{\eqepquesitemsep}[1]{\def\eqepques@itemsep{#1}}
4389 \eqepquestopsep{\eqepques@itemsep}
4390 \eqepquesparsep{\eqepques@parsep}
4391 \eqepquesitemsep{\eqepques@itemsep}
4392 \newenvironment{eqepartsquestions}{%
4393 \begin{list}{}{%
4394 \settowidth{\labelwidth}{\eqe@prtsepSoln\hspace{\tbsolnpartwidth}}
4395 \setlength{\parsep}{\eqepques@parsep}%
4396 \setlength{\itemsep}{\eqepques@itemsep}%
4397 \setlength{\topsep}{\eqepques@topsep}%
4398 \setlength{\itemindent}{0pt}%
4399 \settowidth{\labelsep}{\eqe@hspannerSoln}
4400 \setlength{\leftmargin}{\labelwidth}%
4401 }\item\relax}{\end{list}}

```

## 15.8 Modifications for solutions page

**\gobbletoEndEXT** is a command to gobble all content from the current position **\eqEXT** down to **\endeqEXT**. In the solutions file ends with **\par\medskip**, which we gobble up too. We define **\eqExtArg** to **\thequestionno** so we can use the problem number to filter out the even-problems.

```

4402 \long\def\gobbletoEndEXT#1\endeqEXT{\@gobbletwo}
4403 \def\eqExtArg{\thequestionnoi}

```

When creating a book, we can manually create a chapter and insert the solutions, or we can automatically have it done. The manual method is the default, emit

**\autoInsSolns** **\autoInsSolns** in the preamble to have the solutions inserted automatically. Use **\InputExrSolnsLevel** to tune the section level (**eqexam.def**).

```

4404 \if\load@exerquiz\eqe@NO\DoNotFitItIn\fi
4405 \let\fillInFormatDefault\@empty
4406 \def\fbInsSolnsStyle{\def\exerSolnsHeadnToc{}}
4407 \def\autoInsSolns{\let\fbInsSolnsStyle\relax}
4408 \AtBeginDocument{\fbInsSolnsStyle}
4409 \renewcommand{\exerSolnInput}{%
4410 \global\let\webnewpage\relax
4411 \ifsolutionsonly\else\immediate\closeout\ex@solns\fi
4412 \ifeq@nosolutions\else\newpage % 2012-03-14
4413 \iftherearesolutions\eq@solutionshook\eqsolutionshook
4414 \ifsolutionsonly\else\newpage\eq@solutionshook
4415 \eqsolutionshook\fi
4416 \ifx\webnewpage\relax

```

```

4417         \gdef\webnewpage{\global\let\webnewpage\newpage}%
4418     \fi
4419     \priorexsectitle\exerSolnsHeadnToc\priorexslinput
4420     \InputIfFileExists{\jobname.sol}{\PackageWarning{exerquiz}
4421     {!!! Solutions to exercises not found}}%
4422 \fi
4423 \fi
4424 }

```

`\eqedsplyOnlyFrst` The default listing of a problem with multiple parts is to typeset <num>. (<part>). Here, we do not typeset the number after the first time.

```

4425 \setcounter{partno}{1}\edef\firstPartLtr{\thepartno}
4426
4427     ??? 6/2/11 When part (a) is hidden we need to generate the questions number
4428     for the the first non-hidden part. Created \iffirstProbNumShown to help but it
4429     not working yet.
4430 \newif\iffirstProbNumShown\firstProbNumShownfalse
4431 \def\tb@insertDecPoint{\ifwithinsoldoc\eqedecPointSoln\else
4432     \eqe@decPointMrg\fi}
4433 \newcommand{\eqedsplyOnlyFrst}[2]{\def\thisPart{#2}%
4434     \ifx\thisPart\firstPartLtr\global\firstProbNumShowntrue
4435     \tb@mrgDigitFmt{#1}\tb@insertDecPoint\else
4436     \iffirstProbNumShown\tb@GenProbNum{#1}\else
4437     \global\firstProbNumShowntrue\tb@mrgDigitFmt{#1}%
4438     \tb@insertDecPoint\fi\fi\global\eqeGenProbNumfalse
4439 }

```

`\displayProbNumOnce` If a part is carried over to the next page, it may be necessary to manually force the display of the first digit.

```

4440 \insMidMarg{\displayProbNumOnce}
4441
4442 \newif\ifeqeGenProbNum \eqeGenProbNumfalse
4443 \newcommand{\displayProbNumOnce}{\global\eqeGenProbNumtrue} %
4444 %\def\tb@GenProbNum#1{\ifeqeGenProbNum#1\eqe@decPointMrg\else
4445 \def\tb@GenProbNum#1{\ifeqeGenProbNum#1\tb@insertDecPoint\else
4446     \phantom{#1\tb@insertDecPoint}\fi}%
4447
4448 \def\squ@priorhook{\medskip}
4449
4450     Adjustments of spacing between problems \eqexerskip, and the check for enough
4451     room for the next problem.
4452 \def\default@fvsize@skip{.1}
4453
4454     The skip prior to the beginning of an exercise
4455 \aboveexskip{Opt}
4456
4457     The skip after the end of an exercise
4458 \belowexskip{Opt}

```

The skip in the solutions file following an exercise OR a part of an exercise The text of this command should be a single token, that's why I've enclosed it in braces. (There is a `\@gobbletwo` that gobbles it up for the `studented` option.)

```
4445 \renewcommand\belowexsolnskip{}}
```

We remove the `\mark` from this definition, see original definition in `eqexam.def`

```
4446 \renewcommand\exerSolnHeader[3]{%
4447   \ifeqforpaper\else\webnewpage\fi%\par
4448   \noindent\@ifundefined{hypertarget}
4449   {#3}{\hypertarget{#2}{#3}\relax}\solnhspace
4450 }
```

This causes the `eqexam` environment to write the user friendly name of the exam even if there is only one exam.

```
4451 \def\nNumberOfP@rts{0}
```

## 15.9 Some Convenience/Formatting Commands

`\preExamSolnHead` These are redefinitions of commands defined in `eqexam`, They control the vertical spacing before and after a heading in the solutions at the end of the book, as well  
`\examSolnHeadFmt` as the formatting.  
`\postExamSolnHead`

```
4452 \renewcommand{\preExamSolnHead}{\medbreak\noindent}
4453 \renewcommand{\examSolnHeadFmt}[1]{\textbf{#1}}
4454 \renewcommand{\postExamSolnHead}{\smallskip}
```

`\wrtChapSolnHead` Writes a chapter heading to the solution file, usage,

```
\wrtChapSolnHead{The New {\eqexam}}
```

```
4455 \newcommand{\wrtChapSolnHead}[1]{%
4456   \writeT@SolnFile{%
4457     \protect\preChapSolnHead
4458     \protect\chapHeadSolnFmt{\protect\ftbFmtChapter{\thechapter}#1}%
4459     \protect\postChapSolnHead
4460 }}
```

`\preChapSolnHead` Same as above, except for chapter headings.

```
\chapHeadSolnFmt 4461 \newcommand{\preChapSolnHead}{\bigbreak\noindent}
\postChapSolnHead 4462 \newcommand{\chapHeadSolnFmt}[1]{\large\textbf{#1}}
4463 \newcommand{\postChapSolnHead}{\medbreak}
```

`\ftbFmtChapter` This command may (optionally) insert the chapter number into the chapter title passed to `\wrtChapSolnHead`. The default is to pass the chapter name (“Chapter”) and chapter number. If you say `\let\ftbFmtChapter\@gobble`, the chapter name and number will not appear. You can redefine this command as desired.

```
4464 \newcommand{\ftbFmtChapter}[1]{\chaptername\space#1.\space\ignorespaces}
```

In the solution manual, all these chapter commands may be redefined like so

```
\let\preChapSolnHead\relax
\let\chapHeadSolnFmt\chapter
```

```
\let\ftbFmtChapter@gobble
\let\postChapSolnHead\relax
```

In fact, let's make this into a command.

**\convertChapHeadToChapters** In the solutions manual, the chapter headings will become chapters of the manual, rather than just a bold faced heading.

```
4465 \newcommand{\convertChapHeadToChapters}{%
4466     \let\preChapSolnHead\relax
4467     \let\chapHeadSolnFmt\chapter
4468     \let\ftbFmtChapter@gobble
4469     \let\postChapSolnHead\relax
4470 }
```

**\probset** A simple command to announce the problem set.

```
\subsection*{\probset{\thesection}}
```

See also the definition for the **probset** environment below.

```
4471 \newcommand{\probset}[1]{Problem Set #1}
```

**\annotPage** Use to annotation the page number onto a solution heading, for example,

```
\begin{exam}[\thesection. Another Section\annotPage]{\autoExamName}
```

or using the **probset** environment defined below

```
\begin{probset}{\thesection} Setting the page layout\annotPage}
```

```
4472 \newcommand{\annotPage}{\protect\annotThePage{\thepage}}
```

```
4473 \newcommand{\annotThePage}[1]{\space(page\protect~#1)}
```

## 15.10 The probset and example environments

We define two environments based. The first (**probset**) is based on the **exam** environment; the second (**example**) is based on the **exercise** environment.

**probset** A convenience environment, it is the **exam** environment, renamed, with different arguments. **#1** is the heading that will appear in the margins, and **#2** is the heading to appear in the back of the book.

```
4474 \def\noProbHeader{NPH}
4475 \newenvironment{probset}[2][\probset{\thesection}]{%
4476     \exam[#2]{\autoExamName}\ifx#1\noProbHeader\else
4477     \protected@edef\ftb@tmp@exp{\noexpand\insProbHead{#1}}%
4478     \ftb@tmp@exp\fi}{\endexam}
```

```
4479 \newcounter{exampleno}[section]
```

```
4480 \renewcommand{\theexampleno}{\arabic{section}.\arabic{exampleno}}
```

**Note:** The counter is **exampleno** and is designed to show the section number and example number, and to reset at each section. To change the definition of



`\theexampleno` to reflect the chapter number followed by the example number, and resetting at the beginning of each new chapter, the following code is needed.

```
\usepackage{remreset}
\@removefromreset{exampleno}{section}
\renewcommand{\theexampleno}{\arabic{chapter}.\arabic{exampleno}}
\@addtoreset{exampleno}{chapter}
```

`\examplenoname` is the label placed on the example.

```
4481 \newcommand{\examplenoname}{Example}
```

**example** A simple example environment, based on the `exercise` environment.

```
4482 \newenvironment{example}{%\medskip
4483   \belowexskip{\medskipamount}\aboveexskip{\medskipamount}%
4484   \makeRoomForProb{\@nbaselineskip\baselineskip}{}%
4485   \renewcommand\exlabel{Example}%
4486   \renewcommand\exlabelformat{\textbf{\exlabel~\theexampleno.}}%
4487   \let\eq@fititin\eqfititin
4488   \renewcommand\exrtnlabelformat{$\square$}%
4489   \def\eqexheader@wrapper{\eqexheader}%
4490   \SolutionsAfter
4491   \begin{exercise}[exampleno]}\end{exercise}}
```

**example\*** An example environment with parts.

```
4492 \newenvironment{example*}{%\medskip
4493   \belowexskip{\medskipamount}\aboveexskip{\medskipamount}%
4494   \makeRoomForProb{\@nbaselineskip\baselineskip}{}%
4495   \renewcommand\exlabel{Example}%
4496   \renewcommand\exlabelformat{\textbf{\exlabel~\theexampleno.}}%
4497   \let\eq@fititin\eqfititin
4498   \renewcommand\exrtnlabelformat{$\square$}%
4499   \def\eqexheader@wrapper{\eqexheader}%
4500   \SolutionsAfter
4501   \begin{exercise*}[exampleno]}\end{exercise*}}
```

We set some parameters, to values better suited for the option.

```
4502 \setDefaultfvssizeskip{.1}
4503 \nbaselineskip{4}
```

## 15.11 Commands in support of Solution Manuals

Generally, the solution manual source file should have the same packages as the source file for the book itself, perhaps with a few exceptions, but definitely the `eqexam` package is required with its `fortextbook` option.

At this time, we provide only two commands, these are `\ftbInputBookAux` and `\ftbInputSolnFiles`.

**\ftbInputBookAux** This command is used to input the auxiliary files of the master source file. It takes one argument, the name of the master source file (`myBook.ltx` or `myBook.tex`). If the extension is not present, it is assumed to be `.tex`.

```

4504 \newcommand{\ftbInputBookAux}[1]{%
4505     \filename@parse{#1}\@ifundefined{filename@ext}%
4506         {\def\filename@ext{tex}}{}%
4507     \xdef\tbBaseName{\filename@base}%
4508     \xdef\tbSourceFile{\filename@base.\filename@ext}%

```

In the next 4 lines, we save `\@writefile` and `\@setckpt`, and `\let` them to `\@gobbletwo`. We restore their definitions after we input the aux files. We include the aux files of the source file in case there are some cross references in the solution files, or the body of the text would like to refer back to the original book. (Seems unlikely.)

```

4509     \let\save@writefile\@writefile
4510     \let\@writefile\@gobbletwo
4511     \let\save@setckpt\@setckpt
4512     \let\@setckpt\@gobbletwo
4513     \makeatletter
4514     \InputIfFileExists{\tbBaseName.aux}{%
4515         \PackageInfo{eqexam}
4516             {Inputting auxiliary files of\MessageBreak\tbSourceFile}%
4517         }{%
4518         \PackageError{eqexam}
4519             {Auxiliary files for \tbSourceFile\space were not found}
4520             {Compile the source file \tbSourceFile\space
4521               three times\MessageBreak%
4522               to create the required auxiliary files.}%
4523         }%
4524     \makeatother

```

The solution files really shouldn't have a label, but if we do we'll save the L<sup>A</sup>T<sub>E</sub>X definition of `\label`, and `\let` it two `\@gobble`. Within the body of the solutions, the command `\ftblabel` may be used to cross reference, if needed.

```

4525     \global\let\ftblabel\label
4526     \let\label\@gobble
4527     \let\@writefile\save@writefile
4528     \let\@setckpt\save@setckpt
4529 }
4530 \@onlypreamble\ftbInputBookAux

```

`\restorelabel` These two are used to restore the usual definition of `\label`, and to cancel it out  
`\gobblelabel` by letting it to `\@gobble`.

```

4531 \newcommand{\restorelabel}{\global\let\label\ftblabel}
4532 \newcommand{\gobblelabel}{\let\label\@gobble}

```

`\ftbInputSolnFiles` In the body of the text, place `\ftbInputSolnFiles` in vertical mode. This will input the `.sol` file of the master source document. The optional argument is the name of the solution file. The default name is `\tbBaseName.sol`, where `\tbBaseName` was defined in `\ftbInputBookAux`. If no extension is specified, an extension of `.sol` is assumed. The original `.sol` may have changed its name, if

someone renamed it (to keep it from being overwritten). The solution file may be editing (by hand) as needed.

```

4533 \newcommand{\ftbInputSolnFiles}[1][\tbBaseName.sol]{%
4534   \filename@parse{#1}\@ifundefined{filename@ext}%
4535     {\def\filename@ext{sol}}{}%
4536   \xdef\tbBaseName{\filename@base}%
4537   \xdef\tbSourceFile{\filename@base.\filename@ext}%
4538   \InputIfFileExists{\tbBaseName.sol}{%
4539     \PackageInfo{eqexam}
4540     {Inputting solutions file \tbBaseName.sol\MessageBreak}%
4541   }{%
4542     \PackageError{eqexam}
4543       {Solutions file for \tbSourceFile\space was not found}%
4544       {Compile the source files three times}%
4545   }%
4546 }

4547 % End of textbook segment
4548 \</textbook>
4549 \<ftbsty>

```

## 16 fortexbook Style File

One person said it would be nice to separate `eqexam` from the `fortexbook` option, and have `fortexbook` as a separate style (package). Rather than spending tens of hours separating them I create a simple “wrapper” package, which simply calls `eqexam` with the `fortextook` option along with all the recommended options.

**Usage:**

```

\documentclass[twoside,letterpaper]{book}
\usepackage[fleqn]{amsmath}
\usepackage{fortexbook}
...

```

Below is the style.

```

4550 \NeedsTeXFormat{LaTeX2e}
4551 \ProvidesPackage{fortexbook}
4552 [2012/03/14 v1.0 A fortexbook Package (dps)]
4553 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{eqexam}}
4554 \ProcessOptions
4555 \RequirePackage[%
4556   ftbsolns,fortexbook,usecustomdesign,
4557   forcolorpaper,noseparationrule,usexkv
4558 ]{eqexam}

```

In support of this style, I’ve also defined `\NoSolutions` to compile the document without creating the solutions at the end of the file (this reduces the amount of IO when compiling). I’ve also defined a special option `nocustomdesign` which cancels out the `usecustomdesign` option.

```

4559 % End of ftbsty segment
4560 </ftbsty>
4561 <*package>

```

---

**Input eqtextb.def.** Back in the main package, we choose this point to input the `fortextbook` code (`eqtextb.def`) if the `fortextbook` option is taken.

```

4562 \edef\ftbInputEqTextb{\ifeqfortextbook\noexpand
4563   \InputIfFileExists{eqtextb.def}{\fi}
4564 \ftbInputEqTextb

```

---

## 17 xkeyval Extensions

We load this material if `xkeyval` exists, and if the document author has specified the `usexkv` option.

2014/12/19 Now, the `usexkeys` is on by default.

```

4565 %\IfFileExists{xkeyval}{%
4566 %   \if\eq@usexkeys\eqe@YES\RequirePackage{xkeyval}\else
4567 %   \endinput\fi}{\endinput}

```

We redefine selected commands if the user has specified the `usexkv` option.

### New options for `\fillin`

- underline** Underline the fillin
- u,b** Legacy parameters, underlines (u) or leaves a blank space (b)
- boxed** Boxes in the response region
- boxpretext** When boxed is use, use this to insert text in front of the answer, for example, `x=`
- boxsize** When boxed is taken, use `boxsize` to set the size of the box; permissible choices are `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`
- align** Align the answer within the response region, permissible values are `l`, `c`, `r`.
- color** The color of the response (named `color`)
- format** Special formatting for the answer, the default is `\bfseries`
- enclosesoln** This Boolean key only takes effect when the `boxed` key is used, and when either the `nosolutions` or the `vspacewithsolns` option is taken. When these conditions are met, a box is created around the solution (the third parameter of `\fillin`); the solution is enclosed in a `\phantom` so it is not seen, but the dimensions of the solution are used. This key allows you to create a box of arbitrary dimension.

- fitwidth** The `fitwidth` option uses the natural width of the answer to create the fillin when the `answerkey` option is in effect; otherwise it uses the second parameter #2.
- parbox** The `parbox` parameter may be used to create a multiline `\fillin` box. The value of `parbox` is the same as the first three parameters of the L<sup>A</sup>T<sub>E</sub>X command `\parbox`, e.g., `parbox={[t][.5in][t]}`. The value needs to be enclosed in braces.
- hiddenbox** When the `boxed` option is used, this option resets the `\fbox` parameters to `0pt`, making the box “hidden.”

Below are the `xkeyval` definitions of the keys recognized by `\fillin`. Add some logic to the `underline` key, now it is equivalent to the `b` key.

```

4568 \define@boolkey{eqFillin}{underline}[true]{}
4569 \define@key{eqFillin}{u}[]{\KV@eqFillin@underlinetrue}
4570 \define@key{eqFillin}{b}[]{\KV@eqFillin@underlinefalse}
4571 \define@boolkey{eqFillin}{boxed}[true]{}
4572 \define@key{eqFillin}{boxpretext}[]{\def\eq@fillintext{#1}}
4573 \let\eq@fillintext\@empty

If the user just says parbox,... the value of \eq@fillinparbox is \relax. If
parbox does not appear in the option list, \eq@fillinparbox has a default value
of \@empty. In this way, we can distinguish between parbox with the empty value,
and parbox not present at all.

4574 \define@key{eqFillin}{parbox}[\relax]{\def\eq@fillinparbox{#1}}
4575 \let\eq@fillinparbox\@empty
4576 \define@key{eqFillin}{hiddenbox}[]{%
4577     \def\eq@fillinhiddenbox{%
4578         \setlength{\fboxrule}{0pt}\setlength{\fboxsep}{0pt}}
4579 \let\eq@fillinhiddenbox\@empty
4580 \define@boolkey{eqFillin}{enclosesoln}[true]{}
4581 \define@choicekey+{eqFillin}{boxsize}{tiny,scriptsize,footnotesize,%
4582     small,normalsize,large,Large,LARGE,huge,Huge}[normalsize]{%
4583     \def\eq@eqFillin@boxsize{\text{\csname#1\endcsname\strut}}}%
4584 }{\PackageWarning{eqexam}{Bad choice for boxsize, permissible values
4585     are tiny, scriptsize, footnotesize, small, normalsize,
4586     large, Large, LARGE, huge and Huge. Try again}}
4587 \def\eq@eqFillin@boxsize{\text{\normalsize\strut}}

4588 \define@key{eqFillin}{fboxsep}[3pt]{\def\eq@fillin@fboxsep{#1}}
4589 \def\eq@fillin@fboxsep{3pt}
4590 \define@choicekey+{eqFillin}{fontsize}{tiny,scriptsize,footnotesize,%
4591     small,normalsize,large,Large,LARGE,huge,Huge}[normalsize]{%
4592     \def\eq@eqFillin@fontsize{\csname#1\endcsname}%
4593 }{\PackageWarning{eqexam}{Bad choice for boxsize, permissible values
4594     are tiny, scriptsize, footnotesize, small, normalsize,
4595     large, Large, LARGE, huge and Huge. Try again}}
4596 \def\eq@eqFillin@fontsize{\ifmmode\else\normalsize\fi}
4597 \define@key{eqFillin}{color}[\eq@fillinColor]{\edef\eq@fillin@color{#1}}

4598 \define@choicekey+{eqFillin}{align}[\val\nr]%
4599     {l,r,c}[\eq@eqFillin@align@default]{%

```

```

4600 \def\eq@eqFillin@align{#1}%
4601 \ifcase\nr\relax
4602     \def\eq@align@hfill{}\or
4603     \def\eq@align@hfill{\hfill}\or
4604     \def\eq@align@hfill{\hfil}\fi
4605 }{%
4606 \PackageWarning{eqexam}{Bad choice for align, permissible values
4607 are l, r, and c. Try again}}
4608 \let\eq@align@hfill\relax

```

**defaultalign** is used to change the values of the default macros `\eq@eqFillin@align@default` and `\eq@align@hfill@default` for the align key together.

```

4609 \define@choicekey+{eqFillin}{defaultalign}[\val\nr]{l,r,c}[c]{%
4610     \def\eq@eqFillin@align@default{#1}%
4611     \ifcase\nr\relax
4612         \def\eq@align@hfill@default{}\or
4613         \def\eq@align@hfill@default{\hfill}\or
4614         \def\eq@align@hfill@default{\hfil}\fi
4615 }{%
4616 \PackageWarning{eqexam}{Bad choice for defaultalign, permissible
4617 values are l, r, and c. Try again}}
4618 \setkeys{eqFillin}{defaultalign=c}

```

**\fillInFormatDefault** is the default fill-in format

```

4619 \renewcommand{\fillInFormatDefault}{\normalfont}
4620 \define@key{eqFillin}{format}[\fillInFormatDefault]{%
4621     \def\eq@fillin@format{#1}}
4622 \def\eq@fillin@format{\bfseries}
4623 \def\eq@fbox@corr#1{#1-2\fboxsep-2\fboxrule}
4624 \define@boolkey{eqFillin}{fitwidth}[true]{ } %

```

**boxcmd** The `boxcmd` key is used to define a boxing command. The default is either `\boxed` or `\fbox`. You can say `boxcmd={\fboxcolor{blue}{yellow}}` to obtain a box with a blue frame and yellow background.

```

4625 \@ifundefined{boxed}{%
4626     \def\eq@fillin@defaultbox{\fbox}%
4627     \def\eq@fillin@boxcmd{\fbox}%
4628 }{%
4629     \def\eq@fillin@defaultbox{\boxed}%
4630     \def\eq@fillin@boxcmd{\boxed}%
4631 }

```

**ulcmd** The `ulcmd` key is used to define a underlining command. The default is either `\underline` or `\underbar`. You can say `ulcmd=\underline` to obtained an underline using `\underline`.

```

4632 \define@key{eqFillin}{boxcmd}{%
4633     [\eq@fillin@defaultbox]{\def\eq@fillin@boxcmd{#1}}
4634 \@ifundefined{underbar}{%
4635     \def\eq@fillin@defaultul{underline}%
4636     \def\eq@fillin@ulcmd{\underline}%

```

```

4637     }{%
4638         \def\eq@fillin@defaultul{underbar}%
4639         \def\eq@fillin@ulcmd{\underbar}%
4640     }
(2017/01/28) Added custom underline option, the author must define \ulcustom
and specify ulcmd=custom.
4641 \edef\ulcustom{\expandafter\noexpand\eq@fillin@ulcmd}
4642 \edef\temp@exp{\noexpand
4643     \define@choicekey+{eqFillin}{ulcmd}
4644         {underbar,underline,custom}[\eq@fillin@defaultul]}
4645 \temp@exp{\def\@rgi{#1}\def\eq@custom{custom}\ifx\@rgi\eq@custom
4646     \def\eq@fillin@ulcmd{\@nameuse{ulcustom}}\else
4647     \def\eq@fillin@ulcmd{\@nameuse{#1}}\fi
4648 }{\PackageWarning{aeb}{Bad choice for ulcmd, permissible values
4649     are underbar and underline. Try again}}

```

**lift** (2013/02/16) **lift** lifts (actually lowers) the underline by the amount specified.  
**autolift** **autolift** measures the depth of the content and lifts (actually lowers) the under-  
**addtoautolift** line by that amount. **addtoautolift** works with **autolift** to add the specified  
amount to the amount of lift as calculated by **autolift**.

```

4650 \define@key{eqFillin}{lift}[-1sp]{\def\eq@fillin@lift{#1}}
4651 \define@boolkey{eqFillin}{autolift}[true]{} %
4652 \define@key{eqFillin}{addtoautolift}[0pt]{%
4653     \def\eq@fillin@addtoautolift{#1}}

```

The macro `\eqe@getiii0pts` is based on early parsing code of `\parbox`. It picks up three optional parameters and saves their values under the commands `\eqe@opts@argi`, `\eqe@opts@argii`, `\eqe@opts@argiii`. We are interested in `\eqe@opts@argiii`, which specifies the depth of the `\parbox`. If the `boxed` option is taken, we reduce the value of `\eqe@opts@argiii` by `2\fbboxsep+2\fbboxrule` so that the height will be exactly as specified. The macro `\eqe@getiii0pts` is used with the `parbox` option of `\fillin`. The macro `\eqe@getiii0pts` has syntax:

`\eqe@getiii0pts[pos][height][inner-pos]`

```

4654 \def\eqe@getiii0pts{%
4655     \@ifnextchar[%]
4656     \i@eqe@getiii0pts
4657     {\iii@eqe@getiii0pts{c}{\relax}[s]}}
Get pos
4658 \def\i@eqe@getiii0pts[#1]{%
4659     \@ifnextchar[%]
4660     {\ii@eqe@getiii0pts{#1}}%
4661     {\iii@eqe@getiii0pts{#1}{\relax}[s]}}
Get height
4662 \def\ii@eqe@getiii0pts#1[#2]{%
4663     \@ifnextchar[%]
4664     {\iii@eqe@getiii0pts{#1}{#2}}%
4665     {\iii@eqe@getiii0pts{#1}{#2}[#1]}}

```

Get *inner-pos*

```
4666 \def\iii@eqe@getiiiOpts#1#2[#3]{%
4667   \def\eqe@opts@argi{#1}%
4668   \def\eqe@opts@argii{#2}%
4669   \def\eqe@opts@argiii{#3}}
```

## Redefine the `\fillin` command

`\fillin` Re-worked `\fillin` to have `xkeyval` in the optional first parameter. The syntax is illustrated below.

```
\fillin[
  underline=true|false,u,b,boxed=true|false,boxpretext=<text>,
  align=l|r|c,boxsize=\tiny|..\normalsize|\large|...\Huge,
  color=<namedcolor>,format=<\bfseries|\ttfamily|\Large|whatever>
]{<width>}{<ans>}
```

`\setfillinDefaults` `\setfillinDefaults` sets any default options for `\fillin` the document author wants.

```
4670 \newcommand{\setfillinDefaults}[1]{\def\eqe@argi{#1}\ifx\eqe@argi\@empty
4671   \let\eqe@setfillinDefaults\@empty\else
4672   \def\eqe@setfillinDefaults{#1}\fi}
4673 \let\eqe@setfillinDefaults\@empty
```

`\fillineol` The `\fillineol` command is used to extend the `\fillin` box or line to the end of the line.

```
\fillineol*{<phrase>}[<opts>]{<ans>}
```

Normally, `\fillineol` is placed at the beginning of a line. The command measures the length of `{<phrase>}`, subtracts this from `\linewidth` and uses this value as the width of the underlying `\fillin`. When the star-form is used, the `collectbox` package is required; in the star-form, `<phrase>` can contain verbatim text.

```
4674 \AtBeginDocument{\@ifpackageloaded{collectbox}
4675   {\let\eqe@cb\eqe@YES}{\let\eqe@cb\eqe@NO}}
4676 \def\fillineolTooLongMsg{\PackageWarning{eqexam}{%
4677   The 'phrase' you are measuring is longer than\MessageBreak
4678   \string\linewidth. Changing length to Opt in hopes you\MessageBreak
4679   can fix things}}
4680 \def\fillineolNoCBMsg{\PackageWarning{eqexam}{%
4681   For the \string\fillineol*\space form, the
4682   collectbox package\MessageBreak
4683   is required, but not loaded at this time.\MessageBreak
4684   Switching to \string\fillineol\space in hopes you
4685   load collectbox\MessageBreak
4686   next time or you remove the star-option if not\MessageBreak
4687   really needed}}
4688 \newcommand\fillineol{\@ifstar{\ifx\eqe@cb\eqe@YES
4689   \let\eqe@next\cbfillineol}\else
4690   \def\eqe@next{\fillineolNoCBMsg\fillineol@i}\fi}
```



```

4691 \eqe@next}{\fillineol@i}}
4692 \newcommand{\fillineol@i}[1]{\bgroup\setbox\z@\hbox{#1}\fillineol@ii}
4693 \newcommand{\fillineol@ii}[2][]{\%
4694 \setlength\eqetmplengthb{\linewidth-\wd\z@}%
4695 \ifdim\eqetmplengthb<0pt\eqetmplengthb0pt\fillineolTooLongMsg\fi
4696 \unhbox\z@\fillin[#1]{\eqetmplengthb}{#2}\egroup}

```

Allow the third argument to have verbatim text as well.

```

4697 \newcommand\cbfillineol{\bgroup\collectboxto{\@tempboxa}
4698 {\setlength{\eqetmplengthb}{\linewidth-\wd\@tempboxa}%
4699 \ifdim\eqetmplengthb<0pt\eqetmplengthb0pt\fillineolTooLongMsg\fi
4700 \unhbox\@tempboxa\cbfillineol@ia}}
4701 \newcommand\cbfillineol@ia[1][]{\def\@rgi{#1}\cbfillineol@ib}
4702 \newcommand\cbfillineol@ib{\collectboxto{\@tempboxa}%
4703 {\cbfillineol@i[\@rgi]}}
4704 \newcommand\cbfillineol@i[1][]{\fillin[#1]{\eqetmplengthb}%
4705 {\unhbox\@tempboxa}\egroup}

```

`\fillin` We finally begin the `\fillin` command.

```

4706 \renewcommand{\fillin}[3][]{\begingroup
\ifsp@expand is defined in spdef package. This is a version if \ifsp that expands
correctly in an \edef.

```

```

4707 \expandafter\let\expandafter\ifsp\csname ifsp@expand\endcsname
Get the keys indicated by the document author.
4708 \setkeys{eqFillin}{boxsize,underline=true,boxed=false,%
4709 boxpretext,color,format,enclosesoln=false,fitwidth=false,lift,%
4710 autolift=false,addtoautolift}%
4711 \ifx\eqe@setfillinDefaults\@empty\else
4712 \protected@edef\eq@temp@exp{\noexpand
4713 \setkeys{eqFillin}{\eqe@setfillinDefaults}}%
4714 \eq@temp@exp
4715 \fi
4716 \protected@edef\eq@temp@exp{\noexpand\setkeys{eqFillin}{#1}}%
4717 \eq@temp@exp
4718 \fboxsep\eq@fillin@fboxsep\relax

```

If boxed, we turn underlining off

```

4719 \ifKV@eqFillin@boxed\KV@eqFillin@underlinefalse\fi
Put \eq@fillin@lift to -2sp as a signal not to use the lift value in the case
the author specified autofill and lift.

```

```

4720 \ifKV@eqFillin@autolift\def\eq@fillin@lift{-2sp}\fi

```

Get the second parameter.

```

4721 \edef\eqe@argii{#2}\ifx\eqe@argii\@empty\else
4722 \ifdim\eqe@argii=0pt\let\eqe@argii\@empty\fi\fi

```

We reset `\fboxrule` and `\fboxsep` as needed.

```

4723 \eq@fillinhiddenbox

```

If the document author uses the `hiddenbox` option, this option assumes the `boxed` option as well so we'll set `\KV@eqFillin@boxedtrue` to signal the `boxed` option.

```
4724 \ifx\eq@fillinhiddenbox\@empty\else
4725 \KV@eqFillin@boxedtrue\fi
```

If the `parbox` option is taken, we define the third parameter to be wrapped in a `\parbox`.

```
4726 \ifx\eq@fillinparbox\@empty\def\eqe@argiii{#3}\else
If parbox is specified, we make align=1 the default.
4727 \ifx\eqe@align@hfill\relax
4728 \def\eq@eqFillin@align{1}\def\eqe@align@hfill{}\fi
```

If `parbox` is specified, we get its three optional parameters so we can manipulate the width parameter.

```
4729 \expandafter\eqe@getiiOpts\eq@fillinparbox\relax
```

Now, if this is to be `boxed`, we reduce the height of the box (`\boxed` increases the height by `2\fbboxrule+2\fbboxrule`)

```
4730 \ifKV@eqFillin@boxed
```

`\eqe@opts@argii` has a value of `\relax` if the document author did not specify a height for the box.

```
4731 \expandafter\ifx\eqe@opts@argii\relax\else
4732 \edef\eqe@opts@argii{\expandafter
4733 \eqe@fbbox@corr\expandafter{\eqe@opts@argii}}\fi
4734 \fi
```

We need to feed `\parbox` the parameters it expects, so, if the height parameter is not given, we just pass the first argument; otherwise, we pass all three parameters.

```
4735 \edef\eqe@parboxOptArgs{[\eqe@opts@argi]%
4736 \expandafter\ifx\eqe@opts@argii\relax\else
4737 [\eqe@opts@argii][\eqe@opts@argiii]\fi}%
```

Now we build the third parameter, `\eqe@argiii`.

```
4738 \def\eqe@argiii{\expandafter\parbox\eqe@parboxOptArgs{\eqe@bw}%
```

We insert `\eqe@align@hfill`, which is synchronized to the value of the `align` key to move the `\parbox` contents to left aligned, centered, or right aligned. `\eqe@align@hfill` will only be effective if `#3` is enclosed in a narrower box.

```
4739 {\eqe@align@hfill\ifKV@eqFillin@boxed\eq@fillintext\fi#3}}%
4740 \fi
```

If `\eqe@align@hfill` is still equal to `\relax`, give it the default value.

```
4741 \ifx\eqe@align@hfill\relax
4742 \def\eq@eqFillin@align{c}%
4743 \edef\eqe@align@hfill{\eqe@align@hfill@default}\fi
4744 \ifmmode\let\@eqmath\ensuremath\else\let\@eqmath\text\fi
```

We re-calculate the width of the formatted box

```
4745 \ifx\eq@fillinparbox\@empty
4746 \ifx\eqe@argii\@empty
```

If no `parbox` option and if the second argument is empty, we set width based on the natural width of #3

```

4747         \settowidth{\eqetmplengthb}{\@eqmath{\eq@eqFillin@fontsize
4748         \eq@fillin@format\ifKV@eqFillin@boxed\eq@fillintext\fi
4749         \eqe@argiii}}%
4750         \ifKV@eqFillin@boxed
4751             \setlength{\eqetmplengthb}{%
4752             \eqetmplengthb+2\fbboxsep+2\fbboxrule}%
4753         \fi
4754     \else

```

If #2 is nonempty, we use this value.

```

4755         \setlength{\eqetmplengthb}{#2}%
4756     \fi
4757 \else

```

`parbox` option with empty second argument, use `\linewidth`. for width

```

4758     \ifx\eqe@argii\@empty
4759         \setlength{\eqetmplengthb}{\linewidth}%
4760         \PackageWarning{eqexam}{Parameter \#2
4761         is empty with parbox option,\MessageBreak
4762         using \string\linewidth\space for width}%
4763     }%
4764 \else

```

`parbox` option with second argument, use #2 for width

```

4765         \setlength{\eqetmplengthb}{#2}%
4766     \fi
4767 \fi

```

Return `\ifsp` to its default definition.

```

4768 \expandafter\let\expandafter\ifsp\csname ifsp@default\endcsname

```

Save the final calculated width as `\eqe@bw`.

```

4769 \edef\eqe@bw{\the\eqetmplengthb}%

```

Set the underline option, ...

```

4770 \ifKV@eqFillin@underline\let\@fillinFmt\eq@fillin@ulcmd
4771 \else\let\@fillinFmt\relax\fi

```

however, if `parbox` is specified, we remove the underlining, if any.

```

4772 \ifx\eq@fillinparbox\@empty\else
4773     \ifKV@eqFillin@underline\let\@fillinFmt\relax
4774 %     \ifx\@fillinFmt\underbar\let\@fillinFmt\relax
4775     \PackageInfo{eqexam}{Removing underline option, not permissible
4776     \MessageBreak with parbox option}%
4777 \fi\fi

```

**Build the `\fillin` box.** After the preliminaries, we create the requested answer field. We begin by building the answer field for the case of `\ifeq@proofing` is true (which occurs when the `answerkey` is used).

```

4778 \ifeq@proofing
4779 \ifKV@eqFillin@fitwidth

If the fitwidth option is taken, we measure the width of the box. Ignored when
the parbox option is used.

4780 \settowidth{\eqetmplengthb}{\@eqmath{\eq@fillin@format
4781 \ifx\eq@fillin@parbox\@empty\ifKV@eqFillin@boxed
4782 \eq@fillintext\fi\fi\eqe@argiii}}%

If boxed, we increase the width by 2\fbboxsep+2\fbboxrule; when content is
\boxed, the dimensions are reduced.

4783 \ifKV@eqFillin@boxed
4784 \setlength{\eqetmplengthb}{%
4785 \eqetmplengthb+2\fbboxsep+2\fbboxrule}%
4786 \fi
4787 \edef\eqe@bw{\the\eqetmplengthb}%
4788 \fi

We build the fill-in field for the case of boxed.

4789 \ifKV@eqFillin@boxed
4790 \ifmmode\let\@eqmath\ensuremath\else\let\@eqmath\text\fi
4791 % \mbox{\eq@fillin@format\ensuremath{\boxed{%
4792 \setbox\z@\hbox{\@eqmath{\eq@fillin@boxcmd{%
4793 % \eq@fillin@format\ensuremath{\eq@fillin@boxcmd{%
4794 \eq@eqFillin@boxsize
4795 \@fillinFmt{\eq@eqFillin@fontsize % dpsf0214
4796 \ifKV@eqFillin@boxed
4797 \edef\eqe@bw{\eqe@fbbox@corr{\eqe@bw}}%
4798 \fi

When the boxed option is taken, we adjust the width of the \makebox to get the
desired width #2.

4799 \makebox[\eqe@bw][\eq@eqFillin@align]{\strut
4800 \@eqmath{\eq@fillin@format\color{\eq@fillin@color}}%
4801 \ifx\eq@fillin@parbox\@empty
4802 \eq@fillintext\fi\eqe@argiii}}%
4803 }% end \@fillinFmt
4804 }% end \mbox
4805 \setlength{\@tempdima}{\ht0+\dp0}%
4806 \xdef\fillinTotalHeight{\the\@tempdima}%
4807 \mbox{\unhbox\z@}%
4808 \else

The content is not to be boxed.

4809 \ifmmode\let\@eqmath\ensuremath\else\let\@eqmath\relax\fi
4810 \setbox\z@\hbox{\eq@eqFillin@fontsize % dpsf0214

A value of -1sp signals that lift has not been specified, nor has autofill been
specified. So we use the usual code for underlining.

4811 \ifdim\eq@fillin@lift=-1sp\relax
4812 \@fillinFmt{\makebox[\eqe@bw][\eq@eqFillin@align]{\strut
4813 \color{\eq@fillin@color}}%

```

```

4814         \eqmath{\eq@fillin@format\eqe@argiii}}}%
4815     \else
(2014/02/16) The autolift or lift keys are specified. If autofill, put con-
tents in a box. Set \@tempdima to \dp2+\eq@fillin@addtoautolift, the latter
normally has a value of 0pt unless specified on the option list.
4816         \ifKV@eqFillin@autolift
4817             \setbox2=\hbox{\eqmath{\eq@fillin@format\eqe@argiii}}%
4818             \setlength{\@tempdima}{\dp2+\eq@fillin@addtoautolift}%
4819             \edef\eq@fillin@lift{\the\@tempdima}\fi
(2014/02/16) We lower by an amount of \eq@fillin@lift, but raise the context
by that same amount.
4820         \raisebox{-\eq@fillin@lift}{\@fillinFmt{%
4821             \makebox[\eqe@bw][\eq@eqFillin@align]{\strut
4822             \color{\eq@fillin@color}%
4823             \raisebox{\eq@fillin@lift}{\eqmath{\eq@fillin@format
4824             \eqe@argiii}}}}}%
4825         \fi
4826         \setlength{\@tempdima}{\ht0+\dp0}%
4827         \xdef\fillinTotalHeight{\the\@tempdima}%
4828         \mbox{\unhbox\z0}%
4829     \fi
4830 \else
We begin the case of not \ifeq@proofing, that is, the document author is not
compiling with the answerkey option.
4831     \ifKV@eqFillin@boxed
4832 %         \mbox{\eq@fillin@format\ensuremath{\boxed{%
4833 %         \eq@fillin@format\ensuremath{\eq@fillin@boxcmd{%
4834     \setbox\z0\hbox{\eq@eqFillin@fontsize % dpsf0214
4835         \eqmath{\eq@fillin@format
4836         \eq@fillin@boxcmd{\eq@eqFillin@boxsize\ifeq@nosolutions
4837         \rlap{\eqmath{\eq@eqFillin@fontsize % dpsf0214
4838         \eq@fillintext}}}\fi
4839     \ifvspacewithsolns % dpsf0214 \eq@eqFillin@fontsize
4840     \rlap{\eqmath{\eq@eqFillin@fontsize\eq@fillintext}}}\fi
4841     \@fillinFmt{%
We do a similar thing if proofing is not active (nosolutions is taken).
4842         \makebox[\eqe@fbox@corr{\eqe@bw}]{%
4843             \ifKV@eqFillin@enclosesoln\phantom
4844             {\setlength\eqetmplengthb{\eqe@bw-2\fboxsep}%
4845             \edef\eqe@bw{\the\eqetmplengthb}%
4846             \eqmath{\eqe@argiii}}\else
4847             \strut\hfill\fi
4848         }%
4849     }%end \@fillinFmt
4850 }%end \hbox
4851 \setlength{\@tempdima}{\ht0+\dp0}%
4852 \xdef\fillinTotalHeight{\the\@tempdima}%

```

```

4853         \mbox{\unhbox\z@}%
4854     \else
        This is the case where the field is not boxed.
4855         \setbox\z@\hbox{\eq@eqFillin@fontsize % dpsf0214
4856         \ifdim\eq@fillin@lift=-1sp\relax
4857             \@fillinFmt{\makebox[\eq@bw]{%
4858 %                 \phantom{\@eqmath{\eq@argiii}}
4859                 \strut\hfil}}%
4860         \else
4861             \ifKV@eqFillin@autolift
4862                 \setbox2=\hbox{\@eqmath{\eq@fillin@format
4863                 \eq@argiii}}%
4864                 \setlength{\@tempdima}%
4865                 {\dp2+\eq@fillin@addtoautolift}%
4866                 \edef\eq@fillin@lift{\the\@tempdima}\fi
4867                 \raisebox{-\eq@fillin@lift}{\@fillinFmt{%
4868                 \makebox[\eq@bw]{\raisebox{\eq@fillin@lift}
4869                 {\phantom{\@eqmath{\eq@fillin@format
4870                 \eq@argiii}}\strut\hfil}}}}%
4871             \fi
4872             }\setlength{\@tempdima}{\ht0+\dp0}%
4873             \xdef\fillinTotalHeight{\the\@tempdima}%
4874             \mbox{\unhbox\z@}%
4875     \fi

```

**Online Code.** If the quiz environment is defined, and the user has asked for online option we build a text field.

```

4876     \@ifundefined{@quiz}{}{%
4877         \if\eq@online\eq@YES\relax
4878             \ifeq@nosolutions
4879                 \ifeq@solutionsafter\else
4880                     \ifx\eq@insertverticalspace\eq@YES\relax
        OK, we get this far if we choose online (or higher) and if nosolutions (which
        includes the vspacewithsolns option). We require \eq@insertverticalspace
        to be y. This last value is the default (\SpaceToWork).
4881                     \stepcounter{@cntfillin}%
4882                     \edef\fieldName{%
4883                         \if\probstar*eqexam.\curr@quiz.fillin.%
4884                         \theeqquestionnoi.part\thepartno.%
4885                         fi\the@cntfillin%
4886                     \else
4887                         eqexam.\curr@quiz.fillin.%
4888                         \theeqquestionnoi.fi\the@cntfillin%
4889                     \fi
4890                 }\ifx\eq@fillinparbox\@empty

```

If the parbox option is not taken, we build a text field with height 11bp

```

4891         \raisebox{-1bp}{\makebox[0pt][r]{%

```

```

4892             \textField[\BC{}\presets{\eqe@optsFillIn}]
4893             {\fieldName}{#2}{\fillinTotalHeight}}\else
    If the user has taken the parbox option, then the text field becomes a multiline
    field, with height equal to the requested height.
4894             \setlength{\@tempdima}{%
4895             {\eqe@opts@argii+2\fbboxrule+2\fbboxsep}%
4896             \raisebox{-1bp}{\makebox[0pt][r]{%
4897             \textField[\BC{}\presets{\eqe@optsFillIn}
4898             \Ff{\FfMultiline}]{\fieldName}{#2}
4899             {\fillinTotalHeight}}}\fi
4900             \fi
4901         \fi
4902     \fi
4903 \fi
4904 }%
    (2018/02/02) Removed \space\ignorespaces from \fillin
4905 \fi\endgroup}

```

**\TF** The **\TF** command depends on **\fillin**, so we make the appropriate changes.

```

4906 \renewcommand\TF[2][\defaultTFwidth]{%
4907     \def\eqe@next{\fillin[underline]{#1}{#2}}%
4908     \ifdim\eq@extralabelsep=0pt\relax\else
4909         \if\probstar*\relax\if\exerwparts@cols0
4910             \def\eqe@next{\makebox[0pt][r]{%
4911             \fillin[underline]{#1}{#2}\ignorespaces}%
4912         \fi\fi\fi\eqe@next
4913 }

```

This marks the end of the eqexam package. dps

```

4914 </package>

```

## 18 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	
\#	657, 661, 4760
\%	1702, 1878, 1987, 1992
\@Ans@sq@f@Defaults	1013, 1075
\@EmailCourseName	912, 938, 939
\@EmailExamName	914, 940, 941
\@EmailInstr	910, 936, 937
\@EmailSubject	916, 942, 943
\@Mi	4078
\@PTs	2571, 2572
\@ServerRetnMsg	918, 944, 945
\@SubmitButtonLabel	920, 928
\@actionsAtPageBreak	2139, 2180
\@altTitle	1326, 1329, 1330
\@argi	2777
\@auxout	1238, 1653, 1690, 1692, 1705, 1719, 1722, 1727, 1738, 1742, 1746, 2095, 2111, 2123, 2298, 2300, 2305, 2545, 2675, 2680, 2689, 2704, 2848
\@bypasseqexamheadingfalse	204
\@bypasseqexamheadingtrue	205
\@checkSpacing	80, 2147, 2435, 2436, 2460
\@checkSpacingi	2477
\@ckhide	2763
\@date	1325, 1413
\@eqCommentsColor	529, 2079
\@eqCommentsColorBody	531, 2083
\@eqalignfilllinestoleftfalse	3016, 3018
\@eqalignfilllinestolefttrue	3017
\@eqeuseclassmaketitlefalse	202
\@eqeuseclassmaketitletrue	203
\@eqlinedfillerfalse	2012, 2969, 3641
\@eqlinedfillertrue	2966
\@eqmath	4744, 4747, 4780, 4790, 4792, 4800, 4809, 4814, 4817, 4823, 4835, 4837, 4840, 4846, 4858, 4862, 4869
\@equsedimfalse	3015
\@equsedimtrue	3013, 3014
\@evenfoot	468
\@evenhead	466, 4256, 4327
\@evenllx	4154, 4233
\@evenlly	4151, 4152, 4233
\@examAnsKeyLabel	866, 868
\@examEmailLabel	891, 895, 897
\@examNameLabel	868, 875, 878
\@examSIDLabel	882, 885, 887
\@fillinFmt	2349– 2351, 2353, 2356, 4770, 4771, 4773, 4774, 4795, 4803, 4812, 4820, 4841, 4849, 4857, 4867
\@fvsizekip	2182, 2214, 2216, 2217, 2234, 2456
\@gatherTitleText	613, 615, 629, 656, 660
\@getsecondOf	1261, 1262, 1267
\@gobblet@end	2585, 2589
\@gobbloptone	2921
\@headoffset	4255, 4257, 4260, 4263
\@ifl@ter	1110
\@insertPointsBoxPDF	1578, 1594
\@insertTotalsBoxPDF	1585, 1595
\@instructionsColor	527, 2039
\@isitstar	2530, 2531, 2534, 2639–2641, 2645, 2647, 2820, 2832, 2865
\@linkcolor	541, 980, 982, 983, 1044, 1046, 1047, 1141, 1203
\@listii	2748, 2749
\@listii@SAVE	2748, 2749
\@lti	2530, 2532
\@ltii	2531, 2533
\@markerTotalFmt	1708, 1715
\@marktotalvalue	1537, 1545, 2535, 2575, 2715, 2822, 2842, 2862
\@minus	2037, 2077, 2084
\@mparswitchfalse	3806
\@nbaselineskip	2494, 2495, 2526, 2631, 4484, 4494
\@nnotContStrSkip	2904, 2916, 3176
\@nobreakfalse	2425, 2448
\@nolinkcolor	543, 980, 982, 1044, 1046
\@oddfoot	467, 468
\@oddhead	463, 466, 4259, 4262, 4328
\@oddlx	4157, 4231
\@oddly	4152, 4231
\@onlypreamble	511, 1564, 3821, 4037, 4128, 4319, 4530
\@pdfcreator	375
\@pdfviewparams	390
\@plus	2037, 2077, 2084
\@proofingsymbol	1145, 1243
\@proofingsymbolColor	525, 1145
\@reportpoints	153, 483, 484, 486, 2147, 2430, 2573
\@reporttotals	490, 495, 496, 500
\@rgi	568, 1768, 1772, 2512, 2513, 2518, 2519, 2876, 2881, 3101, 3104, 3351, 3478, 4645, 4701, 4703
\@secondoffive	1262, 2941



\@secondoftwo	1261, 2940	allowcirc4mc (option)	7
\@setckpt	4511, 4512, 4528	\allowcircmcfalse	113
\@spacetobreak	2139–2141	\allowcircmctrue	114
\@sssSolution	1819	\AllowFitItIn	507
\@sssolution	1817, 2014	allowrandomize (option)	8
\@startsection	2024, 2036, 2076	\allowZeroTotals	498, 499
\@subject	823, 824	\Alph	624, 626, 712, 728, 732, 734, 737, 740, 792, 796, 798, 801, 948, 950
\@tableadinitemtrue	2772	\alt@SetSolnMargins	302, 303, 307, 309
\@tempboxa	4697, 4698, 4700, 4702, 4705	\altSetSolnMargins	14, 302
\@templocalversionfalse	671, 808	\altTitle	1322
\@templocalversiontrue	770	\amtSpaceLeftOnPage	80, 2455
\@undefined	2056, 2939	\aNewPage	2928
\@web@title	814, 815	\annotContStr	2897
\@whiledim	3229, 3333, 3388	\annotPage	4472
\@writefile	4509, 4510, 4527	\annotThePage	4472, 4473
\{	1878	\ANS	3843
\}	1884	\Ans@choice	976, 1011, 1040, 1073, 1242
\_	426, 428, 430, 434, 835, 875, 885, 895, 1522, 2029, 2072, 2187, 2312, 2342, 2391, 3959, 3960, 3966, 3967, 3972, 3973	\Ans@ck@sq@f	1099
		\Ans@ck@sq@l	1098
		\Ans@proofing	984, 1017, 1048, 1079, 1205, 1211, 1240
		\Ans@r@l@Defaults	952, 978, 1042
		\Ans@sq@f	1097
		\Ans@sq@f@Actions	1013, 1075
		\Ans@sq@f@driver	997, 1000, 1059, 1062, 1207
		\Ans@sq@l	1096
		\Ans@sq@l@driver	1203
		\ANScolor	3956
		\ANSFmt	3911, 3957
		\AnswerKey	82, 84
		answerkey (option)	6
		\answerkeyfalse	70, 338, 512
		\answerkeyifSave	342
		\answerkeytrue	84, 185, 280, 3793
		\answers	286
		\answers@sq	286
		\applyleadinfix	2609, 2613
		\applyparfixes	85, 2611
		\applyparfixesp	2509, 2613
		\ARG	329, 331
		\arg@i	17
		\argi	1641, 1645, 1649, 1651, 2698, 2703, 2783, 2796, 2815, 2834, 2838, 2841, 3942, 3943
		\argii	1641, 1642, 1646, 1648, 1652, 2783, 2787, 2818
		\AtBeginDocument	175, 367, 564, 567, 680, 1117, 1283, 1536, 1687, 4363, 4408, 4674
		\AtEndDocument	368, 1689
		\AtEndOfPackage	20, 29, 33–39, 44, 46, 53, 54, 69, 101, 112, 135, 183, 192, 262, 265, 746
		\author	811
<b>A</b>			
\A	928, 977, 1011, 1041, 1073		
\AA	1589		
\AACalculate	1589		
\aboveexskip	2500, 4443, 4483, 4493		
\abovesqskip	2220		
\acp@mpah	2852, 2866		
\acp@mpph	2853, 2867		
\acvspace	2897		
\addpenalty	2425, 2448		
\addtoautolift	151		
\addtocounter	663, 1543, 1650, 1694, 1695, 2167, 2176, 2177, 2534, 2543, 2575, 2576, 2699, 2794, 2802, 2821, 2841, 2856, 2862, 2895		
\AddToShipoutPicture	4365		
\aeb@sectioncolor	549		
\aebc@end	361, 362, 677, 696, 792		
\aebshowgraylettersfalse	190		
\aebshowgrayletterstrue	192		
afterChapSolns (environment)	4283		
\afterCommentSkip	2071, 2078, 2080, 2081		
\afterexamsepcode	2162, 2170, 2265		
\aftergroup	1894, 2144, 2265, 2429, 2557, 2712, 3247		
\afterInstrSkip	2028, 2038		
\afterlabelhskip	2527, 2637		
\align	4568		
align (key)	101		

\authorColor .....	525	\cancelleadinfix .....	2610, 2617, 2618
\auto .....	2583, 2590, 2649, 2679	\cancelparfixes .....	85, 2614, 2619
\autocalcparts .....	93, 2652, 2854	\cancelparfixesp .....	2510, 2511, 2616, 2618
\autoExamName .....	2218, 2238, 4476	carryoverFmt (environment) .....	4166
\autoInsSolns .....	141, 4407	\cbfillineol .....	4689, 4697
\autolift .....	151	\cbfillineol@i .....	4703, 4704
<b>B</b>			
\baselineskip .....	1797, 2526, 2606, 2631, 3485, 3490, 3660, 3662, 4484, 4494	\cbfillineol@ia .....	4700, 4701
\Bbox .....	1139, 1164, 1167, 1168, 1172–1174, 1181, 1183, 1184, 1187, 1188, 1209	\cbfillineol@ib .....	4701, 4702
\BC .....	905, 954, 1092, 1582, 1588, 2371, 4892, 4897	\centering .....	1425, 1431, 1437
\beforeCommentSkip .....	2070, 2077, 2084	\centerWidget .....	1131, 1161, 1208
\BeforeIncludedComment .....	4302	cfg (option) .....	5
\beforeInstrSkip .....	2027, 2037, 2044	\cfooteqe .....	1301
\belowexskip .....	2500, 4444, 4483, 4493	\chapHeadSolnFmt .....	4458, 4461
\belowexsolnskip .....	2504, 4445	\chapter .....	4467
bgcolor (key) .....	101	\chapterexercisefalse .....	4266
bgonly* (key) .....	101	\chapterexercisetrue .....	4310
\bGrpANS .....	2669, 2670, 3843	\chaptername .....	4464
\bIFFalseWrtSolns .....	2228, 2233	\chaptersolutions .....	4266, 4282
\bigbreak .....	4461	\chead .....	1290
\bigskip .....	1388	\cheadeqe .....	1284
\bigskipamount .....	1432, 1475, 1502	\cheadSol .....	1308
\boPage .....	2931, 2937, 2940–2942, 2944, 2962	\check@Box .....	1039, 1071
\bopCoverPageText .....	2932, 2962	\chkmarginboxwidth .....	4129, 4364
\bopText .....	2931, 2933	\chnGToNoSolns .....	512
\botmark .....	1644	\circ@Glyph .....	1157, 1160, 1162
\box .....	3469, 3594, 3610, 3613, 3763, 3764	\circProofingForCirc .....	1161, 1178
\boxcmd .....	4625	\ckbox@Color .....	569, 570, 1137
\boxed .....	4568, 4629, 4630, 4791, 4832	\ckboxColor .....	22, 559, 568, 570
\boxpretext .....	4568	\ckcir@Color .....	572, 573, 1158, 1160
\boxsize .....	4568	\ckcirColor .....	22, 560, 571, 573
\bpartsmrk .....	2723, 4024, 4029, 4032	\cleaders .....	2996, 2997, 3001, 3007, 3010
\bslash\textsl_{\texttt{Mrk1}}End .....	59	\clearBotMargin .....	130, 4067, 4353
\bslash\textsl_{\texttt{Mrk1}}nQs .....	59	\clearpage .....	1694
\bslash\textsl_{\texttt{Mrk1}}Start .....	59	\clearTopMargin .....	129, 4059, 4353
\btwnExamSkip .....	316, 325	\clubpenalty .....	3652, 3715
\btwnExamSkipAmt .....	315	\cngMargHeadColorTo .....	4097
\bWebCustomize .....	247	\col@number .....	2428, 2442
<b>C</b>			
\c@eq@count .....	2942, 2943	\collectboxto .....	4697, 4702
\c@partno .....	2894	\color .....	570, 573, 1424, 1430, 1436, 2083, 2354, 4087, 4089, 4568, 4800, 4813, 4822
\c@rryoverFmt .....	4167, 4168	color (key) .....	100
\CA .....	928	\color@begingroup .....	4061, 4063, 4199, 4214
\calcFromMarkers .....	1708	\color@endbox .....	3541
\calcQsBtwnMarkers .....	1717	\color@endgroup .....	4062, 4064, 4207, 4217
\calcTextField .....	1582, 1588	\color@vbox .....	3532
		\columnbreak .....	2428, 2442
		\columnseprule .....	413, 4326, 4343, 4350
		\comment .....	407
		\CommentCutFile .....	4285, 4286, 4293
		\CommentEndDef .....	4297, 4304

\CommentStream	3507, 4295, 4300	2680, 2690, 2704, 2738, 2773, 2794, 2848, 2857,
\cont@notfalse	2920, 2922	2936, 2961, 4290, 4295, 4300, 4302, 4312, 4314
\cont@nottrue	2924	\CT@arc@ 266
\convertChapHeadToChapters	4465	\CT@cell@color 265
\copy	2996, 3552, 3565, 3578	\ctrl@d@exp@exercise 2550, 2552
\copyrightyears	832	\curr@quiz 968, 970, 1003, 1006, 1032, 1035,
coverpage (option)	6	1065, 1068, 1086, 1089, 1580, 1587, 1590, 2058,
\coverpage@subject	1362–1365	2060, 2063, 2067, 2239, 2364, 2368, 4883, 4887
\coverpageCID	1377, 1385	\CurrentOption 75, 79, 226, 4553
\coverpagesubject	1362	\currhideopt 1779, 1798, 1863, 1868, 1902, 1905, 2762
\coverpageSubjectFmt	1362	\currProbHead 4108, 4109
coverpagesumry (option)	6	\customNaming 48, 1395
\coverpageTitleFmt	1362	
\coverpageUniversityFmt	1362	
\cp@@CID	1375, 1376, 1380, 1383, 1411	
\cp@CID	1380, 1411	
\cp@EnclNameAndID	1400, 1401, 1409	
\cp@HghtFrstLn	1402–1404	
\cp@IsetNaming	1457, 1478	
\cp@setHghtFrstLn	1404, 1410	
\cp@SetNameAndIDWidth	1398, 1406–1409	
\cp@SetSumryWidth	1451, 1465, 1466, 1494, 1495	
\cp@sumryWdth	1465–1467, 1494–1496	
\cp@Usefbox		
	1453, 1455, 1463, 1464, 1487, 1492, 1493, 1532	
\cpCID	1375	
\cpCID@argi	1377, 1379, 1381, 1383, 1408	
\cpEnclNameAndID	1400	
\cpNameAndID	1405, 1440	
\cpNofbox	1445	
\cprulelength	1456, 1469, 1498	
\cpSetCIDWidth	1378, 1384	
\cpSetHghtFrstLn	1402	
\cpSetNameAndIDWidth	1396, 1399	
\cpSetSumryWidth	1445	
\cpSumrybypages	51, 62, 1491	
\cpSumrybyparts	50, 61, 1462	
\cpSumryGrade	1445, 1485, 1530	
\cpSumryHeader	1445, 1475, 1502	
\cpSumryPage	1445, 1518	
\cpSumryPts	1445, 1480, 1484, 1521, 1523, 1524, 1529	
\cpSumryTotal	1445, 1482, 1527	
\cpUsefbox	1445	
\csarg	328, 330, 675, 677, 689, 690, 694, 696, 728, 792, 1252, 1253, 1264, 1271, 1471, 1500, 1514, 1521, 1653, 1690, 1692, 1705, 1719, 1721, 1722, 1725, 1727, 1737, 1738, 1741, 1743, 1745, 1746, 2095, 2100, 2102, 2104, 2107, 2111, 2123, 2163, 2246, 2251, 2254, 2274, 2285, 2286, 2290, 2295, 2299, 2300, 2305, 2546, 2620, 2674, 2675,	
		D
		\D@Num 2587, 2588, 2691, 2693, 2697
		\date 849
		\day 861
		\dbMrk 3346, 3404, 3405, 3432
		\DeclareOption 4553
		\DeclareOptionX 5, 7, 9,
		11, 12, 14, 28, 33–39, 44, 45, 47, 49, 52, 55, 56,
		68, 80, 95, 100, 101, 112, 114, 118, 120, 121,
		123, 126, 127, 131, 133, 135, 157, 159, 161,
		163, 168, 173, 174, 176, 181, 187, 191, 201,
		203, 205, 207, 210, 212, 214, 217, 220, 222, 225
		\default@fvsize skip 2213, 2215, 2216, 4442
		\default@nbaselineskip 81, 2492, 2495
		\defaultalign 150
		\defaultInstructions 2025
		\defaultSolnIndent 3998–4000
		\defaultTFwidth 2380, 2381, 4906
		\define@boolkey 3040, 3082, 3088, 3786, 3787, 3803–
		3805, 3807, 3808, 4568, 4571, 4580, 4624, 4651
		\define@choicekey 58, 3026, 3044, 3061, 3071,
		3077, 3084, 3090, 4581, 4590, 4598, 4609, 4643
		\definecolor 3953, 3956, 4084, 4099
		\depthtodate 3225, 3230, 3240, 3250
		\dimen 3252–3254, 3324–3328, 3330, 3333, 3335,
		3338–3341, 3363–3368, 3384, 3386–3388, 3391,
		3392, 3394, 3416–3418, 3424, 3428, 4220–4223
		\ding 285, 1198, 1201, 2634
		\displ@yPointsfalse 519
		\displ@yPointstrue 517, 518
		\displayPointsOff 20, 519
		\displayPointsOn 20, 518
		\displayProbNumOnce 4436
		\displayworkareafalse 85, 180, 186, 194, 196, 282, 3646
		\displayworkareaifSave 344
		\displayworkareaOff 196
		\displayworkareaOn 195

<code>\displayworkareatru</code>	98, 171, 195, 339, 510, 514	<code>\endminipage</code>	2016, 2017, 2019, 3695, 3738
<code>\divide</code>	765, 784, 2289, 3362, 3366	<code>\endpanel</code>	1882
<code>\dlcomment</code>	407, 409	<code>\endparts</code>	1540
<code>\Do</code>	2584, 2588	<code>\endproblem</code>	2553
<code>\documentclass</code>	272	<code>\endproblem*</code>	87
<code>\DoNotFitItIn</code>	508, 4404	<code>\endshortquiz</code>	2264
<code>\DoNotRecordThisExamfalse</code>	2226, 2269	<code>\endshortTitleText</code>	592, 646, 661
<code>\DoNotRecordThisExamtrue</code>	2227	<code>\endsolution</code>	1818, 1820
<code>\DoNum</code>	2399, 2401, 2405, 2632	<code>\endsplitsolution</code>	1823
<code>\dpPtBox</code>	1634, 2134, 2468, 2483	<code>\endtitleMarker</code>	612, 617, 632, 655
<code>\duedate</code>	849, 850, 851	<code>\endverbatimwrite</code>	1883, 3692, 3736
<code>\DV</code>	936, 938, 940, 942, 944	<code>\ensuremath</code>	4744, 4790, 4791, 4793, 4809, 4832, 4833
<code>dvipdfm</code> (option)	11	environments:	
<code>dvipdfmx</code> (option)	11	<code>afterChapSolns</code>	4283
<code>dvips</code> (option)	11	<code>carryoverFmt</code>	4166
<code>dvipsone</code> (option)	11	<code>eq@numparts</code>	19
		<code>eqComments</code>	2070
		<code>eqeList</code>	3941
		<code>eqepartsquestions</code>	4386
		<code>eqpointsofar</code>	19
		<code>eqpointsthispage</code>	19
		<code>exam</code>	2213
		<code>example</code>	4482
		<code>example*</code>	4492
		<code>fullwidthtext</code>	4330
		<code>instructions</code>	2027
		<code>lsol</code>	4055
		<code>priorworkarea</code>	3714
		<code>problem</code>	2497
		<code>problem*</code>	2594
		<code>probset</code>	4474
		<code>splitsolution</code>	63, 1803
		<code>ssol</code>	4056
		<code>workarea</code>	1762
		<code>\eoeTotalOff</code>	503
		<code>\eoeTotalOn</code>	504
		<code>\epartsmrk</code>	4025, 4031, 4033
		<code>\eq@@emitMessageNearBottom</code>	2446, 2447
		<code>\eq@@writeexheaderlist</code>	2766
		<code>\eq@argi</code>	2034, 2040, 2042
		<code>\eq@b@ddCodeSpecial</code>	3483, 3655
		<code>\eq@cfoot</code>	1303, 1307
		<code>\eq@chead</code>	1286, 1295
		<code>\eq@cheadSol</code>	1310, 1315
		<code>\eq@ckglobalhide</code>	2765
		<code>\eq@ColorPackage</code>	158, 198, 200,
			201, 209, 211, 213, 216, 219, 221, 224, 226, 264
		<code>\eq@currProbStartPage</code>	2452, 2906, 2907, 2919
		<code>\eq@custom</code>	4645
		<code>\eq@default@Instructions</code>	2025, 2031

## E

<code>\eachLabel</code>	1555
<code>\ef@Bbox</code>	1167, 1172, 1183, 1187
<code>\egroup</code>	880, 889,
	899, 1156, 1379, 1383, 1398, 1451, 1765, 2016,
	2017, 2152, 2183, 2429, 2452, 2455, 2476, 2491,
	2701, 3215, 3343, 3469, 4208, 4228, 4696, 4705
<code>\eGrpANS</code>	3843
<code>\eIFFalseWrtSolns</code>	2229, 2268
<code>\email</code>	811
<code>email</code> (option)	7
<code>\EmailCourseName</code>	33, 912, 913
<code>\EmailExamName</code>	33, 914, 915
<code>\EmailSubject</code>	34, 916, 917
<code>\emitMessageNearBottom</code>	2441, 3022
<code>\encloseProblemsWith</code>	521
<code>\enclosesoln</code>	4568
<code>\end@ssSolution</code>	1820
<code>\end@sssolution</code>	1818, 2018
<code>\endanswers</code>	287
<code>\endanswers@sq</code>	287
<code>\endcomment</code>	408
<code>\enddlcomment</code>	408, 410
<code>\endeqEXt</code>	3801, 4402
<code>\endeqSavedComment</code>	410, 3734
<code>\endexam</code>	4478
<code>\endexamtotal@color</code>	545, 2148
<code>\endexercise@parts@tabular</code>	1540
<code>\endgraf</code>	1971, 2424
<code>\endinput</code>	247, 4567
<code>\endlongTitleText</code>	592, 632, 642, 655, 658
<code>\endmanswers</code>	289
<code>\endmanswers@sq</code>	289

<code>\eq@e@ddCodeSpecial</code> .....	3656, 3692, 3697	<code>\eq@listType</code> .....	961, 989, 1024, 1051
<code>\eq@Email</code> .....	895, 1412	<code>\eq@lw@f</code> .....	992, 1054
<code>\eq@emitMessageNearBottom</code> .....	2443–2445	<code>\eq@lw@l</code> .....	963, 1027
<code>\eq@eqFillin@align</code> 4600, 4728, 4742, 4799, 4812, 4821		<code>\eq@max@selected</code> .....	130, 131
<code>\eq@eqFillin@align@default</code> .....	4599, 4610	<code>\eq@NO</code> .....	3640, 3707
<code>\eq@eqFillin@boxsize</code> .....	4583, 4587, 4794, 4836	<code>\eq@nolinkfalse</code> .....	238, 339, 2762
<code>\eq@eqFillin@fontsize</code> .....	4592, 4596,	<code>\eq@nolinkifSave</code> .....	343
4747, 4795, 4810, 4834, 4837, 4839, 4840, 4855		<code>\eq@nolinktrue</code> .....	170, 509, 2767, 2768
<code>\eq@ExamLastPage</code> .....	43, 1246, 1311	<code>\eq@nosolutionsfalse</code> .....	236, 339
<code>\eq@ExamName</code> .....	874, 1289, 1327, 1410	<code>\eq@nosolutionsifSave</code> .....	344
<code>\eq@extralabelsep</code> .....	2383, 2393, 4908	<code>\eq@nosolutionstrue</code> .....	170, 509, 514
<code>\eq@fillin@addtoautolift</code> .....	4653, 4818, 4865	<code>\eq@nosummarytotals</code> .....	
<code>\eq@fillin@boxcmd</code> .....		.. 55, 148, 491, 501, 502, 2032, 2033, 2040, 2042	
..... 4627, 4630, 4633, 4792, 4793, 4833, 4836		<code>\eq@nototals</code> .....	147
<code>\eq@fillin@color</code> .....	4597, 4800, 4813, 4822	<code>\eq@nSelectedVersion</code> ..	713, 716, 762, 764, 766, 779
<code>\eq@fillin@defaultbox</code> .....	4626, 4629, 4633	<code>eq@numparts</code> (environment) .....	19
<code>\eq@fillin@defaultul</code> .....	4635, 4638, 4644	<code>\eq@nVersions</code> ..	588, 591, 618, 628, 709, 716, 721, 743
<code>\eq@fillin@fboxsep</code> .....	4588, 4589, 4718	<code>\eq@online</code> .....	119, 138, 139, 347, 392, 902,
<code>\eq@fillin@format</code> .....		973, 995, 1038, 1057, 1102, 1163, 1180, 2358, 4877	
.. 4621, 4622, 4748, 4780, 4791, 4793, 4800,		<code>\eq@parttotals</code> .....	46, 149, 503, 504, 2132, 2138
4814, 4817, 4823, 4832, 4833, 4835, 4862, 4869		<code>\eq@pointbox</code> .....	
<code>\eq@fillin@lift</code> .....	4650,	... 479, 1628, 1630, 1632, 1634, 1635, 3552,	
4720, 4811, 4819, 4820, 4823, 4856, 4866–4868		3553, 3555, 3565, 3567, 3578, 3593, 3609, 3613	
<code>\eq@fillin@ulcmd</code> 4636, 4639, 4641, 4646, 4647, 4770		<code>\eq@pointboxtotalheight</code> 480, 1630–1633, 2133–	
<code>\eq@fillin@color</code> .....	547, 4597	2136, 2143, 2465, 2467, 2468, 2470, 2482–2485	
<code>\eq@fillin@hiddenbox</code> .....	4577, 4579, 4723, 4724	<code>\eq@proofingfalse</code> .....	189, 237, 338, 513
<code>\eq@fillin@parbox</code> .....	4574,	<code>\eq@proofingifSave</code> .....	342
4575, 4726, 4729, 4745, 4772, 4781, 4801, 4890		<code>\eq@proofingtrue</code> .....	85, 187, 188, 280, 3793
<code>\eq@fillintext</code> .....		<code>\eq@protect</code> .....	977, 1011, 1041, 1073
. 4572, 4573, 4739, 4748, 4782, 4802, 4838, 4840		<code>\eq@Radio@driver</code> .....	988, 1012, 1074
<code>\eq@fititin</code> .....	507, 508, 1143, 4487, 4497	<code>\eq@RadioCheck@driver</code> .....	988
<code>\eq@globalshowsolutionsfalse</code> .....	235	<code>\eq@ren@selected</code> .....	130, 133
<code>\eq@globalshowsolutionstrue</code> .....	173, 175	<code>\eq@renditionOptions</code> .....	129, 132, 134, 747
<code>\eq@hidesolutionfalse</code> .....	234, 2762	<code>\eq@replaceToken</code> .....	668
<code>\eq@hspanner</code> .....	985, 1018, 1049, 1080	<code>\eq@rfoot</code> .....	1305, 1307
<code>\eq@initializeContAnnot</code> .....	2736	<code>\eq@rgi</code> .....	571
<code>\eq@initLoop</code> .....	2888, 2892, 2893	<code>\eq@rhead</code> .....	1288, 1295
<code>\eq@insertContAnnot</code> .....	2923, 2925, 2927, 3233	<code>\eq@rheadSol</code> .....	1312, 1315
<code>\eq@insertverticalspace</code> .....		<code>\eq@selectedItem</code> .....	2887, 2891
..... 69, 145, 146, 151, 1757, 1795,		<code>\eq@selectedVersion</code> .....	593,
2011, 2361, 3630, 3640, 3643, 3707, 3746, 4880		602, 666, 667, 705, 706, 712, 719, 2556, 2711	
<code>\eq@item</code> .....	2761, 2763	<code>\eq@selectedVersion@save</code> .....	
<code>\eq@item@latex</code> .....	2913	..... 593, 594, 602, 603, 643, 645, 647, 649	
<code>\eq@l@check@driver</code> .....	977, 1041	<code>\eq@setWidgetProps</code> .....	977, 1012, 1041, 1074
<code>\eq@l@l</code> .....	964, 1028	<code>\eq@shortTitlesFix</code> .....	619, 650
<code>\eq@leadin@item</code> .....	2761, 2765	<code>\eq@shortVersionAtext</code> .....	577, 580
<code>\eq@lfoot</code> .....	1301, 1307	<code>\eq@shortVersionBtext</code> .....	577, 581
<code>\eq@lhead</code> .....	1284, 1295	<code>\eq@SID</code> .....	885, 1411
<code>\eq@lheadSol</code> .....	1308, 1315	<code>\eq@solutionsafterfalse</code> .....	229, 232, 338, 513
<code>\eq@linesXPgs</code> .....	3257, 3262	<code>\eq@solutionsafterifSave</code> .....	343

<code>\eq@solutionsaftertrue</code> .....	85, 179, 231, 281	<code>\eqe@align@hfill</code> .....	
<code>\eq@solutionshook</code> .....	1316, 4413, 4414	<code>\eqe@align@hfill@default</code> .....	4602–4604, 4608, 4727, 4728, 4739, 4741, 4743
<code>\eq@sqlabel</code> .....	2343	<code>\eqe@allow</code> .....	1666, 1668, 1670, 1678
<code>\eq@sqsllabel</code> .....	1213, 1214, 2342	<code>\eqe@argi</code> ...	1718, 1730, 1740, 1917, 2051, 2052,
<code>\eq@sqslrtnlabel</code> .....	2326, 2330, 2332		2075, 2080, 2081, 2249, 2303, 2304, 2590, 4670
<code>\eq@sqslsecrunhead</code> .....	2341	<code>\eqe@argii</code>	616, 617, 2590, 2591, 4721, 4722, 4746, 4758
<code>\eq@sqslsectitle</code> .....	2340	<code>\eqe@argiii</code> .....	4726, 4738, 4749, 4782,
<code>\eq@temp@exp</code> .....	4712, 4714, 4716, 4717		4802, 4814, 4817, 4824, 4846, 4858, 4863, 4870
<code>\eq@tmp</code> .....	2244, 2245	<code>\eqe@auto@chk@drivers</code> .....	115, 118, 120, 121, 123
<code>\eq@tmpbox</code> .....	964, 993, 1028, 1055, 1127	<code>\eqe@Bbox</code> .....	39, 1209
<code>\eq@tmpdima</code> .....	964, 976, 983, 984,	<code>\eqe@BboxCirc</code> .....	1160, 1164, 1174, 1181
	993, 1028, 1040, 1047, 1048, 1055, 1128, 1204,	<code>\eqe@BboxRect</code> ...	1137, 1139, 1168, 1173, 1184, 1188
	1205, 1890, 1891, 1932, 1934–1936, 2740–2743	<code>\eqe@BGColor</code> .....	3094, 3095, 3410, 3411
<code>\eq@tmplength</code> .....	963, 966, 992, 994,	<code>\eqe@bParts</code> .....	2778–2780
	1027, 1030, 1054, 1056, 2744–2747, 2908, 2909	<code>\eqe@bw</code> .....	4738, 4769, 4787, 4797,
<code>\eq@tmplengthA</code> .....	477, 875, 877, 885, 886,		4799, 4812, 4821, 4842, 4844, 4845, 4857, 4868
	895, 896, 1380, 1382, 2391–2393, 3274, 3280,	<code>\eqe@calc@percent</code> .....	2279, 2283
	3286, 3295, 3299, 3303, 3309, 3312, 3330, 3340	<code>\eqe@cb</code> .....	4675, 4688
<code>\eq@tmplengthB</code> .....	478, 3548, 3557, 3569	<code>\eqe@chkZeroTotals</code> .....	1666, 1669, 1677
<code>\eq@usexkeys</code> .....	127, 128, 4566	<code>\eqe@cirDiam</code> .....	1153, 1158
<code>\eq@VersionAtext</code> .....	576, 578	<code>\eqe@cirRadius</code> .....	1155, 1157
<code>\eq@VersionBtext</code> .....	576, 579	<code>\eqe@color@opt</code> .....	198, 200
<code>\eq@WriteLineColor</code> .....		<code>\eqe@commentChkMsg</code> .....	363, 367
	2984, 2990, 2999, 3003, 3039, 3421, 3431	<code>\eqe@contTitleText</code> .....	598, 607, 610
<code>\eq@YES</code> ....	3272, 3293, 3510, 3630, 3643, 3694, 3737	<code>\eqe@coverPageNaming</code> .....	1392–1394, 1458
<code>\eqbothmargins</code> .....	487, 1638	<code>\eqe@cprulelength</code> .....	
<code>\eqcenterWidget</code> .....	1130, 1133–1135, 1162, 1209		1470, 1478, 1483, 1486, 1499, 1519, 1528, 1531
<code>eqComments (environment)</code> .....	2070	<code>\eqe@csarg</code> .....	241, 242
<code>\eqCommentsColor</code> .....	525	<code>\eqe@decPointMrg</code> .....	
<code>\eqCommentsColorBody</code> .....	531, 532, 558		3926, 3971, 3974, 3978, 3981, 4428, 4438
<code>\eqcustomdesignfalse</code> .....	6, 276	<code>\eqe@decPointPrb</code> .....	425,
<code>\eqcustomdesigntrue</code> .....	5		426, 432, 2314, 2318, 2321, 3218, 3958, 3961, 4002
<code>\eqdashrulefill</code> .....	3000, 3003	<code>\eqe@defNumRefii</code> .....	43, 1260, 1278, 1279
<code>\eqdashruleVfill</code> .....	3010, 3012	<code>\eqe@DoNum</code> .....	2399, 2401, 2632
<code>\eqdotrulefill</code> .....	2995, 2999	<code>\eqe@DoWarning</code> .....	2397, 2399, 2400, 2411, 2412
<code>\eqdotruleVfill</code> .....	3007, 3009	<code>\eqe@dpsepPrb</code> .....	426
<code>\eqDriverName</code> .....	164,	<code>\eqe@drivernum</code> .....	
	165, 208, 210, 212, 215, 218, 220, 223, 228, 389		208, 210, 212, 215, 217, 220, 223, 227, 370
<code>\eqDvipstone</code> .....	387, 389	<code>\eqe@emnb</code> .....	2442, 2444, 2451
<code>\eqe@holdTemp</code> .....	685, 686, 757–759, 772	<code>\eqe@eqOpts</code> .....	166, 167, 380
<code>\eqe@leadinitem</code> .....	2722, 2734	<code>\eqe@fbox@corr</code> .....	4623, 4733, 4797, 4842
<code>\eqe@numLinesL</code> .....	3029, 3166	<code>\eqe@fillLinesNumFmt</code> .....	3096, 3168, 3170, 3172
<code>\eqe@numLinesR</code> .....	3030, 3171	<code>\eqe@fillwidth</code> .....	3200, 3215, 3237, 3358,
<code>\eqe@OffVersion</code> .....	683, 684, 702, 763		3386, 3401, 3407, 3412, 3414, 3434, 3439, 3536
<code>\eqe@OnVersion</code> .....	681, 682, 701, 756, 758	<code>\eqe@fextendedfalse</code> .....	106, 111, 2013, 3639
<code>\eqe@SumryVert</code> .....	1388, 1390, 1463, 1492	<code>\eqe@fextendedtrue</code> .....	102, 3628
<code>\eqe@absorbTokensLong</code> .....	633, 642	<code>\eqe@flnexti</code> ....	3657, 3678, 3712, 3731, 3733, 3740
<code>\eqe@absorbTokensShort</code> .....	635, 646	<code>\eqe@flnextii</code> .....	3734, 3736, 3742
<code>\eqe@adjForSepRule</code> .....	2132, 2151, 2155	<code>\eqe@flnsofnsfalse</code> .....	3472, 3474, 3618
<code>\eqe@afterexamsepcode</code> .....	2137, 2170		

<code>\eqe@flnosolnstrue</code> .....	3473, 3617	2652, 2657, 2658, 2722, 2725, 2730, 3183, 3184,
<code>\eqe@Four</code> .....	42, 2655	3186, 3795, 3796, 4689–4691, 4907, 4910, 4912
<code>\eqe@fpmrk</code> .....	2058, 2562, 2872, 2873	<code>\eqe@nextitem</code> .....
<code>\eqe@getiiiopts</code> .....	4654, 4729	2877, 2878, 2882, 2883, 2896
<code>\eqe@gobbletoend</code> .....	2497, 2531, 2640	<code>\eqe@NO</code> .....
<code>\eqe@gobnxtpar</code> .....	2609, 2610, 2666	40, 46, 69, 81, 96, 138,
<code>\eqe@grabarg</code> .....	2498, 2530, 2639	140–142, 145, 147, 148, 169, 177, 182, 489, 497,
<code>\eqe@h</code> .....	43, 2513	501, 503, 506, 752, 995, 1057, 1667, 1668, 2011,
<code>\eqe@HOfVrule</code> ....	3372, 3381, 3542, 3583, 3587,	2223, 2225, 2505, 2521, 2585, 2586, 2603, 2628,
3588, 3591, 3592, 3598, 3602, 3603, 3606, 3607		2733, 2734, 2758, 2784, 2788, 2888, 2893, 3042,
<code>\eqe@HOfVruleFrstRow</code> ..	3376, 3378, 3412, 3414, 3437	3102, 3108, 3125, 3133, 3145, 3153, 4404, 4675
<code>\eqe@holdTemp</code> .....	686, 799	<code>\eqe@nocustomdesign</code> .....
<code>\eqe@hspannerMrg</code> .....		7, 8, 276
.....	3889, 3902, 3918, 3945, 3973, 3976, 3981	<code>\eqe@nopointsfalse</code> .....
<code>\eqe@hspannerPrb</code> .....	429, 430, 432,	488
1624, 2187, 2210, 2314, 3960, 3964, 4003, 4383		<code>\eqe@nopointstrue</code> .....
<code>\eqe@hspannerSoln</code> 434, 435, 2209, 2318, 2321, 2322,		490
3967, 3970, 3989, 4012, 4015, 4022, 4382, 4399		<code>\eqe@nOutOfNum</code> .....
<code>\eqe@idinfohl</code> .....	870, 878, 887, 1383	2412, 2414, 2633
<code>\eqe@initializeMultiVersions</code> .....	672, 680, 700	<code>\eqe@nskeyflsplit</code> .....
<code>\eqe@innermarkpts</code> .....	1537, 1541	2993, 3244, 3468, 3469, 3532,
<code>\eqe@insert@more@content</code> .....	3231,	3552, 3565, 3578, 3579, 3581, 3594, 3610, 3763
3344, 3453, 3627, 3642, 3668, 3675, 3749, 3755		<code>\eqe@nskeyfltop</code> .....
<code>\eqe@insertContAnnot</code> .....	2905, 2925	2994, 3553, 3554, 3557,
<code>\eqe@insertLogo</code> .....	1351, 1353, 1357	3567–3569, 3579, 3580, 3589, 3604, 3608, 3764
<code>\eqe@insertSolns</code> .....	3627, 3652, 3668, 3675	<code>\eqe@numbersep</code> .....
<code>\eqe@isle@dinnext</code> .....	2605, 2606	3037, 3038, 3168, 3170, 3172
<code>\eqe@isPts</code> .....	2432,	<code>\eqe@numLinesL</code> .....
2435, 2499, 2520, 2525, 2526, 2627, 2630, 2631		3028–3030, 3034, 3236
<code>\eqe@IW</code> .....	1896, 1987, 1992, 2016	<code>\eqe@numLinesR</code> .....
<code>\eqe@IW0</code> .....	277, 295,	3028–3030, 3034, 3237
297, 323, 1225, 1233, 1238, 1653, 1690, 1692,		<code>\eqe@numRefii</code> .....
1705, 1719, 1722, 1727, 1738, 1742, 1746, 1864,		44, 1270, 1273, 1274
1870, 1879, 1884, 1900, 2095, 2111, 2123, 2298,		<code>\eqe@offVersion</code> .....
2300, 2305, 2545, 2675, 2680, 2689, 2704, 2848		644, 648, 684, 738
<code>\eqe@l</code> .....	1856, 1858	<code>\eqe@One</code> .....
<code>\eqe@lcir</code> .....	1150, 1152, 1159	41, 2147, 2529, 2530, 2534,
<code>\eqe@leadinitem</code> .....	2659, 2720	2574, 2592, 2639, 2641, 2650, 2820, 2832, 2865
<code>\eqe@localTextTitle</code> .....	595–597, 604–606	<code>\eqe@onVersion</code> .....
<code>\eqe@lPanel</code> ....	1946, 1952, 1961, 1968, 1996, 2002	682, 735
<code>\eqe@makeVgrid</code> .....	108,	<code>\eqe@optiont@kenMsg</code> .....
2983, 3192, 3196, 3355, 3458, 3625, 3671, 3752		78, 82, 97, 170, 178, 183
<code>\eqe@margininsert</code> .....	4094, 4095	<code>\eqe@opts@cargi</code> .....
<code>\eqe@marktxt</code> .....	1538, 1541, 1542	4667, 4735
<code>\eqe@MarParBox</code> .....	4088, 4232, 4234	<code>\eqe@opts@cargii</code> ..
<code>\eqe@mkRoomPgBrk</code> .....	2423, 2429	4668, 4731–4733, 4736, 4737, 4895
<code>\eqe@movePTs</code> .....	2579, 2582	<code>\eqe@opts@cargiii</code> .....
<code>\eqe@nDoNum</code> .....	2400, 2402, 2632	4669, 4737
<code>\eqe@next</code> ..	92–94, 618, 619, 629, 631, 640, 1898,	<code>\eqe@optscbf</code> .....
1900, 1903, 1907, 1913, 2199, 2382, 2385, 2388,		1020, 1022, 1071
		<code>\eqe@optscbl</code> .....
		1021, 1022, 1039
		<code>\eqe@optsFillIn</code> ....
		2346, 2347, 2372, 4892, 4897
		<code>\eqe@optsmltf</code> .....
		1082, 1083, 1092
		<code>\eqe@optsRadiof</code> .....
		957, 959, 1009
		<code>\eqe@optsRadiol</code> .....
		958, 959, 975
		<code>\eqe@OutOfNum</code> .....
		2411, 2413, 2633
		<code>\eqe@p@gobnxtpr</code> .....
		2509–2511, 2552
		<code>\eqe@p@gobnxtpr@rDef</code> .....
		2507, 2509
		<code>\eqe@p@gobtop@rnext</code> .....
		2507, 2508
		<code>\eqe@paneli</code> .....
		1850, 1851, 1853, 1855
		<code>\eqe@panelir</code> .....
		1850, 1852
		<code>\eqe@parboxOptArgs</code> .....
		4735, 4738
		<code>\eqe@partsIndent</code> .....
		2909, 2912, 2914
		<code>\eqe@pForPart</code> .....
		2291, 2293–2295
		<code>\eqe@pointsEach</code> .....
		2688, 2784, 2788, 2792
		<code>\eqe@pointsPartsId</code> 2574, 2594, 2650, 2655, 2789, 2791
		<code>\eqe@prehold</code> .....
		3905, 3915, 3932, 3935
		<code>\eqe@prevProbZero</code> ....
		2431, 2520, 2521, 2627, 2628
		<code>\eqe@priorw@content</code> .....
		3714, 3749, 3755
		<code>\eqe@prtsepMrg</code> .....
		3944, 3972, 3975

<code>\eqe@prtsepPrb</code> .....	427,	<code>\eqe@YES</code> 40, 55, 73, 80, 95, 119, 120, 122, 124, 127,
428, 2740, 2770, 2908, 3278, 3959, 3962, 3963		128, 139, 146, 149–151, 168, 176, 181, 369,
<code>\eqe@prtsepSoln</code> .....	3966, 3969, 4016, 4021, 4394	370, 377, 379, 381, 491, 498, 502, 504, 505,
<code>\eqe@ps@gobnxtpar</code> .....	2608	560, 750, 761, 902, 926, 973, 1038, 1102, 1163,
<code>\eqe@ps@gobtop@r</code> .....	2605, 2607	1180, 1412, 1666, 1670, 1678, 1757, 1795, 2032,
<code>\eqe@ptsFmt</code> .....	1479,	2033, 2040, 2042, 2132, 2138, 2150, 2154, 2221,
1484, 1520, 1529, 1562, 1563, 1568, 1570,		2222, 2235, 2247, 2358, 2361, 2431, 2517, 2520,
1576, 1661, 1698, 1699, 1701, 2046, 2563, 2565		2584, 2620, 2621, 2627, 2688, 2737, 2760, 2792,
<code>\eqe@r</code> .....	1856, 1858	2800, 2892, 2895, 3055, 3066, 3109, 3110, 3117,
<code>\eqe@reqPack</code> .....	417, 419	3137, 3188, 3746, 4566, 4675, 4688, 4877, 4880
<code>\eqe@rgi</code> .....	1763, 1764	<code>\eqe@Zero</code> 41, 2531, 2573, 2591, 2640, 2645, 2647, 2789
<code>\eqe@rPanel</code> .....	1949, 1957, 1962, 1969, 2016, 2017	<code>\eqe@zeroTotalsAllowed</code> .....
<code>\eqe@setfillinDefaults</code> .....	4671–4673, 4711, 4713	497, 498, 1667
<code>\eqe@setStartSolns</code> .....	3682	<code>\eqeachLabel</code> .....
<code>\eqe@setVRule</code> .....	3377, 3380, 3421, 3431	1557, 1571, 2566
<code>\eqe@showArg</code> .....	687, 690	<code>\eqeAEFormatting</code> .....
<code>\eqe@striphbox</code> .....	1142, 1143	291, 292, 296, 297
<code>\eqe@Strut</code> .....	2075, 2080	<code>\EQEcalculateAllTotals</code> .....
<code>\eqe@subleftgutter</code> .....	1951, 1967, 1997	1333, 2271
<code>\eqe@SumryHoriz</code> .....	1387, 1389, 1441	<code>\eqeCurrProb</code> .....
<code>\eqe@SumryVert</code> .....	1386, 1389, 1441	2898
<code>\eqe@tableadin</code> .....	2660, 2772	<code>\eqedbfalse</code> .....
<code>\eqe@tb@ship@ut</code> .....	4192, 4196	3347
<code>\eqe@tb@shipout</code> .....	4185, 4365	<code>\eqedecPointSoln</code> .....
<code>\eqe@temp</code> .....	2938, 2940, 2941	3965, 3968, 3989, 4006, 4427
<code>\eqe@tempbox</code> .....	2992, 2996, 2997, 3202, 3203	<code>\eqedepth</code> .....
<code>\eqe@tempcnta</code> 471, 3383, 3418, 3420, 3422, 3423, 3425		1831, 1938, 1946,
<code>\eqe@Three</code> .....	42	1953, 1963, 1970, 1984, 1986, 1990, 1996, 2003
<code>\eqe@tmp</code> .....	656, 660, 664, 674–677,	<code>\eqedsplyOnlyFrst</code> .....
2556, 2557, 2711, 2712, 3360, 3361, 3368, 3369		3901, 4008, 4425
<code>\eqe@tmp@exp</code> .....	27, 31, 245	<code>\eqefillLines@bgonlystar</code> ...
<code>\eqe@tmp@i</code> .....	26, 28, 30, 242–244	3092, 3093, 3448, 3449
<code>\eqe@tmpexp</code> .....	263, 264	<code>\eqefillLines@outlineonlystar</code> 3086, 3087, 3444, 3445
<code>\eqe@tmphold</code> .....	4048, 4049, 4052, 4053	<code>\eqeGenProbNumfalse</code> .....
<code>\eqe@turnOffComment</code> .....	693, 737, 740, 798, 801	4434, 4436
<code>\eqe@turnOnComment</code> .....	688, 732, 734, 796	<code>\eqeGenProbNumtrue</code> .....
<code>\eqe@Two</code> .....	42, 2791	4437
<code>\eqe@usedeffboxrule</code> ...	3042, 3055, 3066, 3102, 3188	<code>\eqeGrandTotal</code> .....
<code>\eqe@webOpts</code> .....	164, 165, 374	2277, 2278
<code>\eqe@wordNumbsError</code> .....	2406, 2410, 2417	<code>\eqeifnext</code> .....
<code>\eqe@wordNums</code> .....	79, 2405, 2409, 2415, 2420	4248, 4251
<code>\eqe@workfill</code> .....	1829, 1833	<code>eqeList</code> (environment) .....
<code>\eqe@workwidth</code> .....	1827, 1831	3941
<code>\eqe@writetoAux</code> .....	1236, 2250, 2253, 2738, 2772	<code>\eqEmail</code> .....
<code>\eqe@writetoSolns</code> .....	1229, 2250, 2253	32, 893, 900
<code>\eqe@wrtLineKernal</code> .....	3201, 3265	<code>\eqEmail@argi</code> .....
<code>\eqe@x</code> .....	3214,	898
3221, 3233, 3265, 3275, 3276, 3281, 3284, 3297		<code>\eqEmail@argii</code> .....
<code>\eqe@y</code> .....	3265, 3276, 3404	894, 896, 897
		<code>\eqemaketitle</code> .....
		1331, 1347
		<code>\eqemargin</code> .....
		301, 308, 416, 424, 432, 1574,
		1602, 1625, 1951, 2185, 2203, 2742, 2755, 2911,
		2913, 2915, 3268, 3284–3286, 3996, 3997, 4376
		<code>\eqEndExamTotalColor</code> .....
		525
		<code>\eqeomarginbox</code> .....
		48, 50, 51, 2149
		<code>\eqeomarginboxleft</code> .....
		50, 1620
		<code>\eqeomarginboxright</code> .....
		48, 51, 1610
		<code>\eqeonlinefalse</code> .....
		117
		<code>\eqeonlinetrue</code> .....
		119
		<code>\eqePANELCUT</code> .....
		1887, 1894
		<code>\eqepanelheight</code> ..
		1840, 1844, 1845, 1848, 1934, 1935
		<code>\eqepanelwidth</code> .....
		1839, 1843, 1847
		<code>eqepartsquestions</code> (environment) .....
		4386
		<code>\eqepques@itemsep</code> .....
		4388, 4396
		<code>\eqepques@parsep</code> .....
		4387, 4395
		<code>\eqepques@topsep</code> .....
		4386, 4397
		<code>\eqepquesitemsep</code> .....
		4388, 4391
		<code>\eqepquesparsep</code> .....
		4387, 4390



<code>\eqepquestopsep</code> .....	4386, 4389	<code>\eqExerSolnHeader</code> .....	2766
<code>\eqeques@itemsep</code> .....	2191, 2205, 4378, 4389, 4391	<code>\eqexheader</code> .....	1101, 1114, 4489, 4499
<code>\eqeques@listparindent</code> .....	2192, 2208, 4381	<code>\eqexheader@wrapper</code> .....	1100, 1114, 4489, 4499
<code>\eqeques@parsep</code> .....	2190, 2204, 2263, 4377, 4390	<code>\eqexlisttabheader</code> .....	2769
<code>\eqeques@topsep</code> .....	2189, 2206, 4379	<code>\eqEXt</code> .....	3795, 3799, 3800
<code>\eqequesitemsep</code> .....	2191, 2195, 4371	<code>\eqExtArg</code> .....	141, 4403
<code>\eqequeslistparindent</code> .....	2192, 2196, 4372	<code>\eqfititin</code> .....	507, 3023, 4487, 4497
<code>\eqequesparsep</code> .....	2190, 2194, 4370	<code>\eqforinstrfalse</code> .....	12, 13
<code>\eqequestopsep</code> .....	2189, 2193, 4369	<code>\eqforinstrtrue</code> .....	11
<code>\eqeSetExamPageParams</code> .....	436, 458	<code>\eqforpaperfalse</code> .....	240
<code>\eqeSolnItemMngt</code> .....	2198, 2202, 4375	<code>\eqforpapertrue</code> .....	157, 159, 161
<code>\eqeSumryHoriz</code> .....	48, 1389, 1391	<code>\eqfortextbookfalse</code> .....	10
<code>\eqeSumryVert</code> .....	1386	<code>\eqfortextbooktrue</code> .....	9
<code>\eqetmplengtha</code> .....		<code>\eqglobalversionfalse</code> .....	669
... 481, 1469, 1470, 1498, 1499, 1780, 1784,		<code>\eqglobalversiontrue</code> .....	703
1787, 3036, 3037, 3198, 3200, 3268–3270, 3273,		<code>\eqgrii</code> .....	323
3274, 3278–3281, 3285, 3290–3292, 3294, 3295,		<code>\eqgriii</code> .....	310
3298, 3299, 3302, 3303, 3308, 3309, 3311, 3312		<code>\eqleftmargin</code> .....	483, 1573, 1638
<code>\eqetmplengthb</code> 482, 3218, 3270, 3292, 4694–4696,		<code>\eqleftmarginbox</code> .....	1555
4698, 4699, 4704, 4747, 4751, 4752, 4755, 4759,		<code>\eqlocalversionfalse</code> .....	670, 2559, 2714
4765, 4769, 4780, 4784, 4785, 4787, 4844, 4845		<code>\eqlocalversiontrue</code> .....	776
<code>\eqevtranstotbox</code> .....	57, 1664, 1665, 1672, 1681	<code>\eqobeylocalversionfalse</code> .....	125, 807
<code>\eqeWrtExamTitleToSolns</code> .....	2221–2223, 2247	<code>\eqobeylocalversiontrue</code> .....	126, 771
<code>\eqex@coverpage</code> .....	56, 152, 1536, 2946	<code>\eqp@rtc@lcm@rk</code> .....	2761, 2840
<code>\eqex@coverpageSubjectFmt</code> .....	1370, 1429	<code>\eqpanelbox</code> .....	64, 1838, 1842–1845
<code>\eqex@coverpageTitleFmt</code> .....	1373, 1435	<code>\eqpointLabel</code> .....	1561, 2048
<code>\eqex@coverpageUniversityFmt</code> .....	1367, 1423	<code>\eqpointsLabel</code> .....	1559, 2047, 2049
<code>\eqExam</code> .....	1641, 1642, 1644	<code>eqpointsofar (environment)</code> .....	19
<code>\eqExam@Ans@ck@sq@f</code> .....	1051, 1099	<code>eqpointsthispage (environment)</code> .....	19
<code>\eqExam@Ans@ck@sq@l</code> .....	1023, 1098	<code>\eqPriorVspace</code> .....	1104
<code>\eqExam@Ans@sq@f</code> .....	989, 1097	<code>\eqptLabel</code> . 1556, 1569, 1571, 1577, 1662, 2564, 2566	
<code>\eqExam@Ans@sq@l</code> .....	960, 1096	<code>\eqptsLabel</code> . 1555, 1569, 1572, 1577, 1663, 2564, 2566	
<code>\eqExam@cargi</code> .....	894	<code>\eqrightmarginbox</code> .....	485, 1578, 1639
<code>\EqExam@SubmitURL</code> .....	910, 932	<code>\eqSavedComment</code> .....	409, 3731
<code>\eqexamargi</code> .....	2240, 2241, 2243	<code>\eqSID</code> .....	32, 884, 890
<code>\eqexamargii</code> .....		<code>\eqSID@cargi</code> .....	884, 888
. 2237–2239, 2241, 2244, 2250, 2251, 2259, 2260		<code>\eqSID@cargii</code> .....	884, 886, 887, 1407
<code>\eqexamCFG</code> .....	19, 25	<code>\eqSolnExCmds</code> .....	3515, 3538
<code>\eqexamdefReq</code> .....	1110, 1113	<code>\eqsolutionshook</code> .....	1308, 4413, 4415
<code>\eqexammargin</code> .....	424	<code>\eqtemptokena</code> .....	586, 612, 623, 625, 627
<code>\eqExamName</code> .....	32, 872, 881	<code>\eqtemptokenb</code> .....	587, 622, 624, 626
<code>\eqExamName@cargi</code> .....	873, 879	<code>\eqTopOfSolnPage</code> .....	279
<code>\eqExamName@cargii</code> .....	873, 876, 878, 1406	<code>\equalCellSizesOff</code> .....	
<code>\eqExamNumPagesSolns</code> .....	1251, 1256, 1257	... 107, 3080, 3116, 3192, 3196, 3206, 3319, 3321	
<code>\eqExamPageLayout</code> .....	446	<code>\equalCellSizesOn</code> .....	107, 3079, 3317
<code>\eqExamPriorVspace</code> .....	1084, 1104	<code>\equsecolorfalse</code> .....	156, 157
<code>\eqExamRunHead</code> .....	465, 1295, 1318	<code>\equsecolortrue</code> .....	159, 161
<code>\eqexamsubject</code> .....	1364, 1431	<code>\eqWLSpacing</code> .....	2986, 3112
<code>\eqexcoverpage</code> .....	56, 1350	<code>\eqWriteLine</code> . 2971, 2972, 2974, 2976, 2978, 2980,	
<code>\eqexcoverpagedesign</code> .....	1358, 1362	2983, 3156, 3157, 3208, 3215, 3237, 3434, 3439	

<code>\eqWriteLineBlankFill</code>	2978, 3004, 3005	<code>\exlabelformatwp</code>	2315
<code>\eqWriteLineColor</code>	2984	<code>\exp@temp</code>	676, 691, 695
<code>\eqWriteLineDashFill</code>	2976, 3002, 3050	<code>\expGT</code>	2288, 2289
<code>\eqWriteLineDashVFill</code>	3012, 3051	<code>\exPrtsep</code>	3963
<code>\eqWriteLineDots</code>	2974, 2998, 3053	<code>\exrtnlabelformat</code>	2324, 2328, 4488, 4498
<code>\eqWriteLineFill</code>	2972, 2980, 2983, 2989, 3047, 3059, 3160	<code>\exrtnlabelformatwp</code>	2325, 2329
<code>\eqWriteLineVDots</code>	3009, 3054	<code>\exsecrunhead</code>	2338, 2339
<code>\eqWriteLineVFill</code>	3006, 3048, 3060, 3161	<code>\exsectitle</code>	2334, 2335
<code>\eqwritetomarginsfalse</code>	14	<code>\exsectitletext</code>	2333, 2334, 2336
<code>\eqwritetomarginstrue</code>	15	<code>\exsllabelformat</code>	2316, 4005
<code>\eqx@separationrule</code>	150, 505, 506, 2150, 2154	<code>\exsllabelformatwp</code>	2319, 4007
<code>\evensidemargin</code>	441, 4139, 4143, 4145, 4254, 4323, 4340, 4347	<code>\exsolafter</code>	2310, 3494, 3497
<code>\every@qRadioButton</code>	978, 1042	<b>F</b>	
<code>\every@radioButton</code>	978, 1013, 1042, 1075	<code>\F</code>	936, 938, 940, 942, 944, 1582, 1588
<code>\every@sqRadioButton</code>	1014, 1076	<code>\fbInsSolnsStyle</code>	4406–4408
<code>\everypar</code>	2759, 2760, 3723	<code>\fbox</code>	1453, 1464, 1487, 1493, 1532, 4626, 4627
<code>\eWebCustomize</code>	248	<code>\fboxrule</code>	1465, 1494, 1551, 3253, 4578, 4623, 4752, 4785, 4895
<code>\ex@solns</code>	294, 306, 322, 1215, 1223, 1978, 4276–4278, 4411	<code>\fboxsep</code>	1465, 1494, 4197, 4578, 4623, 4718, 4752, 4785, 4844, 4895
<code>\Exam</code>	576, 598, 643	<code>\Ff</code>	881, 890, 954, 1093, 4898
<code>\exam</code>	4476	<code>\FfMultiline</code>	1093, 4898
<code>exam (environment)</code>	2213	<code>\FfNoToggleToOff</code>	954
<code>\examAnsKeyLabel</code>	865, 867	<code>\FfRequired</code>	881, 890
<code>\exambegdef</code>	2085, 2258	<code>\FHidden</code>	936, 938, 940, 942, 944, 1582, 1588
<code>\examEmailLabel</code>	891, 892	<code>\fieldName</code>	967, 975, 1002, 1010, 1031, 1039, 1064, 1071, 1085, 1093, 1580, 1583, 1587, 1591, 2363, 2373, 4882, 4893, 4898
<code>\examenddef</code>	2130, 2262	<code>\filename@base</code>	4507, 4508, 4536, 4537
<code>\examenvfalse</code>	2173, 4038	<code>\filename@ext</code>	4506, 4508, 4535, 4537
<code>\examenvtrue</code>	2127	<code>\filename@parse</code>	4505, 4534
<code>\examNameLabel</code>	32, 868, 869	<code>\fillerBgIm@ge</code>	3348, 3349, 3409, 3414
<code>\examNum</code>	574	<code>\fillerCustomBg</code>	108, 3348
<code>example (environment)</code>	4482	<code>\fillerLinesAlignDef</code>	99, 2009, 3018, 3073
<code>example* (environment)</code>	4492	<code>\fillerLinesOnLeftMargin</code>	99, 3017, 3074
<code>\examplenoname</code>	145, 4481	<code>\fillin</code>	153, 2346, 2382, 2386, 4670, 4907, 4911
<code>\examSIDLabel</code>	882, 883	<code>\fillinColor</code>	525
<code>\examSolnHeadFmt</code>	1215, 4452	<code>\fillineol</code>	152, 4681, 4684, 4688
<code>\excludacomment</code>	522, 676, 695, 4055, 4056, 4283, 4313, 4316	<code>\fillineol@i</code>	4690–4692
<code>\execjs</code>	381	<code>\fillineol@ii</code>	4692, 4693
<code>\ExecuteOptionsX</code>	100, 115, 116, 119, 124, 206	<code>\fillineolNoCBMsg</code>	4680, 4690
<code>\exer@solnheadhook</code>	753	<code>\fillineolTooLongMsg</code>	4676, 4695, 4699
<code>\exercisesAtEndOfChapter</code>	4283	<code>\fillInFormatDefault</code>	150, 2396, 4405, 4619, 4620
<code>exerquizOpts (option)</code>	9	<code>\fillinTotalHeight</code>	4806, 4827, 4852, 4873, 4893, 4899
<code>\exerSolnHeader</code>	4446	<code>\fillinWidth</code>	2389
<code>\exerSolnInput</code>	4409	<code>\filllLinesLineWidth</code>	1783, 1784, 3173
<code>\exerSolnsHeadnToc</code>	184, 4406, 4419	<code>\filllLinesNumFmt</code>	101, 3096–3098
<code>\exerwparts@cols</code>	2384, 4909	<code>\fillTypeBlankLine</code>	2978, 3067
<code>\exlabel</code>	2308, 4485, 4486, 4495, 4496	<code>\fillTypeDashLine</code>	2976, 3064
<code>\exlabelformat</code>	2312, 2313, 2315, 4001, 4486, 4496		

<code>\fillTypeDefault</code> .....	2967, 2980	<code>forcolorpaper</code> (option) .....	9
<code>\fillTypeDots</code> .....	2974, 3065	<code>forcolorpaper*</code> (option) .....	9
<code>\fillTypeGrid</code> .....	2982, 3068, 3629	<code>forinstr</code> (option) .....	4
<code>\fillTypeHRule</code> .....	2972, 3063	<code>\foritem</code> .....	2872
<code>\firstemittrue</code> .....	4039	<code>\foritem@cont</code> .....	2879, 2884, 2885
<code>\firstitemfalse</code> .....	2198, 4033	<code>\forleadinitem</code> .....	2872
<code>\firstitemtrue</code> .....	2212, 4032, 4385	<code>\format</code> .....	4568
<code>\firstPageOfExam</code> .....	1258	<code>forpaper</code> (option) .....	9
<code>\firstPartLtr</code> .....	4425, 4430	<code>\forproblem</code> .....	2872
<code>\fitwidth</code> .....	4568	<code>forstudent</code> (option) .....	4
<code>\fl@Cnt</code> .....	3486, 3491, 3502, 3504	<code>fortextbook</code> (option) .....	4
<code>\fl@CutName</code> .....	3486, 3492, 3654, 3691, 3718, 3733	<code>\forVersion</code> ..	134, 597, 606, 644, 648, 669, 2556, 2711
<code>\fl@firstpassfalse</code> .....	3222, 3541	<code>\FPdiv</code> .....	2291
<code>\fl@firstpasstrue</code> .....	3180, 3679	<code>\FPMul</code> .....	2293
<code>\fl@getcontent</code> .....	3501, 3691, 3733	<code>\FPround</code> .....	2294
<code>\fl@iSmash</code> .....	3396, 3398, 3401	<code>\fracPrt</code> .....	3345
<code>\fl@nRows</code> .....	3220, 3242, 3356	<code>\firstProbNumShownfalse</code> .....	2664, 4426
<code>\fl@oSmash</code> .....	3396, 3398, 3400, 3456	<code>\firstProbNumShowntrue</code> .....	4430, 4433
<code>\fl@set@nnotContStrSkip</code> .....	3176, 3181	<code>\ftb@argi</code> .....	3884
<code>\fl@vsplitandplace</code> .....	3530, 3654, 3718	<code>\ftb@CloseEqeList</code> .....	123, 3904
<code>\fl@wlspaceing</code> .....	3099, 3100, 3112	<code>\ftb@defineInsSpan</code> .....	123, 3884, 3914
<code>\flbaselineskip</code> .....	3490, 3561, 3566, 3571	<code>\ftb@endprobstarCks</code> .....	2667, 2709
<code>\fleqnOff</code> .....	358, 359	<code>\ftb@EndSpanPrts</code> .....	3889, 3893
<code>\fleqnOn</code> .....	358, 359	<code>\ftb@EqeListPrtsFmt</code> .....	123, 3895, 3902, 3918
<code>\flEqu@lLineWidth</code> .....	3210, 3211, 3317, 3319, 3341	<code>\ftb@InsSpan</code> .....	3885, 3886, 3921
<code>\fextended</code> (key) .....	7	<code>\ftb@isANSListOpenfalse</code> .....	3844, 3859, 3936
<code>\fextendedInput</code> .....	101, 102, 256	<code>\ftb@isANSListOpentru</code> .....	3929
<code>\flfboxrule</code> .....	2991, 3001, 3006, 3011, 3187, 3188, 3194, 3234, 3235, 3251, 3258, 3260, 3281, 3297, 3325, 3328, 3339, 3357, 3375, 3428, 3477, 3543, 3587, 3591, 3602, 3606	<code>\ftb@OpenEqeListPrts</code> .....	123, 3899, 3921
<code>\flfrstsplittfalse</code> .....	3585, 3600	<code>\ftb@spanPrts</code> .....	123, 3885, 3891
<code>\flfrstsplitttrue</code> .....	3255, 3471	<code>\ftb@tmp@exp</code> .....	4477, 4478
<code>\fliPartNo</code> .....	2886, 2894	<code>\ftbFmtChapter</code> .....	4458, 4464
<code>\flnum</code> .....	3096, 3097	<code>\ftbInputBookAux</code> .....	4504
<code>\flPageBreakMsg</code> .....	3021	<code>\ftbInputEqTextb</code> .....	4562, 4564
<code>\fls@vebaselineskip</code> .....	3489, 3540	<code>\ftbInputSolnFiles</code> .....	4533
<code>\flSeparateCutNames</code> .....	111, 3486	<code>\ftblabel</code> .....	4525, 4531
<code>fltype</code> (key) .....	100	<code>ftbsolns</code> (option) .....	7
<code>\flwriteexsol@fter</code> .....	3493, 3516, 3523, 3526	<code>fullwidthttext</code> (environment) .....	4330
<code>\font</code> .....	1150	<code>\fvsizekip</code> .....	2217
<code>\footnotesize</code> .....	3346		
<code>\footskip</code> .....	444, 455, 4090, 4091		
<code>\forceEqu@lCells</code> .....	3209, 3322		
<code>\forceEqualCellsfalse</code> ..	3115, 3191, 3195, 3205, 3320		
<code>\forceEqualCellstrue</code> .....	3318		
<code>\forceNoColor</code> .....	525		
<code>\ForceNoColorfalse</code> .....	160		
<code>\forceNoColorSet</code> .....	551, 564, 567		
<code>\ForceNoColortrue</code> .....	162, 566		

## G

<code>\g@addto@macro</code> .....	132, 134, 682, 684, 686
<code>\getDimSSPanel</code> .....	64, 1841, 1888
<code>\getSpaceLeftOnPage</code> .....	80, 2453
<code>\gobblelabel</code> .....	4531
<code>\gobbletoEndExt</code> .....	141, 3796, 4402
<code>\gobtodot</code> .....	3345, 3361, 3369
<code>\goodbreak</code> .....	1216
<code>\gridHLineFill</code> 3047, 3050, 3053, 3059, 3156, 3160, 3208	
<code>\gridIndent@dj</code> .....	3352–3354, 3402, 3403
<code>\gridIndentAdj</code> .....	108, 3351

`\gridpgbrkfalse` ..... 3174, 3440, 3655, 3681  
`\gridpgbrktrue` ..... 3223  
`gridtype (key)` ..... 100  
`\gridtypeselected` ..... 3043, 3045, 3103, 3159  
`\gridVLineFill` ..... 3048,  
                   3051, 3054, 3060, 3157, 3161, 3378, 3381, 3420  
`\grpANSDelimiter` ..... 123, 3907, 3932

## H

`\halfHtPtBox` ..... 1631, 2133, 2432, 2467, 2482  
`\hangafter` ..... 966, 994, 1030, 1056  
`\hangindent` ..... 966, 994, 1030, 1056  
`\hangSolWPrtFmt` ..... 4024  
`\hb@xt@` ..... 2996, 2997, 3001, 3202  
`\hbadness` ..... 4078  
`\hCommSpace` ..... 2072  
`\HEADERcolor` ..... 4097  
`\headheight` ..... 437, 452, 4150  
`\headsep` ..... 439, 453, 4150  
`\height` ..... 1604, 1614, 1621, 1672, 1681  
`\hglue` ..... 3168,  
                   3170, 3172, 3275, 3281, 3297, 3403, 3418, 3430  
`\hidden@ttr` ..... 2514, 2515, 2518, 2519, 2551  
`\hiddenbox` ..... 4568  
`\hideproofing` ..... 10, 189  
`\Hidesymbol` ..... 175  
`\hInstrSpace` ..... 2029, 2035  
`\hline` ..... 1549–1551  
`\hrule` ..... 1137, 1138, 2991  
`\hrulefill` ..... 3005  
`\hsc` ..... 2075, 2082  
`\hsi` ..... 2035, 2040–2042  
`\htPtBox` ..... 1635  
`\hyper@anchor` ..... 2056, 2939  
`\hypersetup` ..... 816, 821, 827, 830  
`\hypertarget` ..... 1101, 4449

## I

`\i@eqe@getiiiOpts` ..... 4656, 4658  
`\iacvspace` ..... 2899, 2900, 2902, 2905, 3022  
`\ic@vspace` ..... 2899  
`\ic@vspacedefault` ..... 95, 2901, 2902  
`\idinfoHighlight` ..... 870, 871  
`\if@bypasseqexamheading` .... 204, 1538, 1686, 3780  
`\if@eqalignfilllinestoleft` .....  
                   3016, 3221, 3267, 3307, 3329, 3395, 3723  
`\if@eqeuseclassmaketitle` ..... 202, 1344  
`\if@eqlinedfiller` ..... 3182, 3483, 3658  
`\if@equsedim` ..... 1930, 3013  
`\if@templocalversion` ..... 671, 807

`\ifAB` ..... 576, 577, 666  
`\ifaebshowgrayletters` ..... 190  
`\ifallowcircmc` ..... 113, 1149  
`\ifanswerkey` ..... 70,  
                   91, 255, 421, 866, 999, 1061, 2010, 2323, 2928,  
                   2929, 3513, 3523, 3526, 3538, 3621, 3637, 3663  
`\ifc@sewrap` ..... 2404, 2405, 2408, 2409  
`\ifchapterexercises` ..... 4266, 4311  
`\ifcont@nnot` ..... 2920, 3665, 3743  
`\ifdim` ... 316, 1934, 2144, 2183, 2383, 2428, 2451,  
                   2463, 2465, 2480, 2484, 3558, 3560, 3570, 3814,  
                   3996, 4130, 4695, 4699, 4722, 4811, 4856, 4908  
`\ifdispl@yPoints` ..... 517,  
                   1544, 2536, 2560, 2716, 2822, 2832, 2836, 2864  
`\ifdisplayworkarea` 193, 1756, 1794, 3732, 3735, 3745  
`\ifdisplayworkareaOff` ..... 10  
`\ifdisplayworkareaOn` ..... 10  
`\ifDoNotRecordThisExam` ..... 2226, 2232, 2267  
`\ifeq@globalshowsolutions` 235, 350, 396, 1869, 1906  
`\ifeq@hidesolution` ..... 234, 349, 395, 2765  
`\ifeq@nolink` ..... 238, 353, 399  
`\ifeq@nosolutions` ..... 236, 351,  
                   356, 397, 422, 903, 926, 982, 1046, 1901, 1940,  
                   1991, 2359, 2767, 3730, 4308, 4412, 4836, 4878  
`\ifeq@proofing` ..... 237, 352, 398, 1242, 2352, 4778  
`\ifeq@solutionsafter` .....  
                   229, 348, 393, 394, 423, 903, 926, 1793,  
                   1824, 1830, 1835, 1868, 1892, 1897, 1941, 1942,  
                   1945, 1960, 1979, 2000, 2017, 2360, 2768, 4879  
`\ifeqcustomdesign` ..... 6, 457  
`\ifeqe@fextended` ..... 111, 3664  
`\ifeqe@flnosolns` ..... 3472, 3732, 3735, 3744  
`\ifeqe@nopoints` ..... 488, 1686  
`\ifeqedb` ..... 3347, 3404, 3405, 3422, 3432,  
                   3582–3584, 3586, 3590, 3597–3599, 3601, 3605  
`\ifeqeGenProbNum` ..... 4436, 4438, 4439  
`\ifeqeonline` ..... 117, 254  
`\ifeqexamCFG` ..... 19, 24  
`\ifeqforinstr` ..... 13  
`\ifeqforpaper` ..... 240, 355, 401, 4447  
`\ifeqfortextbook` .... 10, 417, 1119, 1341, 2126,  
                   2173, 2178, 2662, 2709, 2723, 3779, 4362, 4562  
`\ifeqglobalversion` ... 593, 602, 669, 762, 2555, 2710  
`\ifeqlocalversion` ..... 670, 2555, 2710  
`\ifeqbeylocalversion` ..... 125, 751, 769, 775  
`\ifequsecolor` ..... 156  
`\ifeqwritetomargins` ..... 15, 4364  
`\ifexamenv` ..... 4038  
`\iffalse` ..... 2228  
`\IfFileExists` ..... 197, 4565

<code>\iffirstemit</code> .....	4039	<code>\ifvspacewithsolns</code> .....	86,
<code>\iffirstitem</code> .....	2197, 2198		93, 337, 420, 1118, 1248, 1939, 3644, 4279, 4839
<code>\iffll@firstpass</code> .....	3178, 3531	<code>\ifWithinANSGrp</code> ..	2668, 3843, 3851, 3870, 3929, 3930
<code>\iffllfrstsplit</code> .....	3471, 3545, 3585, 3600	<code>\ifwithinparts</code> ..	2910, 3271, 3289, 3509, 3694, 3737
<code>\ifforceEqualCells</code> .....	1782, 3316, 3323	<code>\ifwithinsoldoc</code> .....	961, 990, 1025, 1052,
<code>\ifForceNoColor</code> .....	160, 563, 566		2188, 2202, 2209, 2212, 4375, 4382, 4385, 4427
<code>\iffirstProbNumShown</code> .....	4426, 4432	<code>\ifwriteVertic@lFLines</code> .....	3175, 3406, 3623
<code>\iffbtb@isANSListOpen</code> .....	3844, 3917	<code>\ifxetex</code> .....	116
<code>\ifgridpgbrk</code> .....	3174, 3437	<code>\ignorePTsStar</code> .....	84, 2570
<code>\ifiscarryover</code> .....	4165, 4180, 4248	<code>\ii@eqe@getiiiOpts</code> .....	4660, 4662
<code>\ifisinlineans</code> .....	3910	<code>\iii@eqe@getiiiOpts</code> .....	4657, 4661, 4664–4666
<code>\ifisinstred</code> 2130, 3793, 3811, 3833, 3909, 3939, 4110		<code>\importdljs</code> .....	381
<code>\ifisleadin</code> .....	2732, 4028	<code>\include@quizsolutions</code> .....	2344
<code>\ifismarginans</code> .....	2130, 3833, 3913, 3938, 4110	<code>\include@solutions</code> .....	4269, 4272
<code>\ifisOnline</code> .....	139	<code>\include@solutions@chapter</code> .....	4269, 4271
<code>\ifisstudented</code> .....	3788, 3812	<code>\includecomment</code> ...	523, 691, 4055, 4056, 4284, 4315
<code>\ifkeepdeclaredvspacing</code> .....	3683	<code>\includeexersolutions</code> .....	368, 4270, 4275
<code>\ifkeyalt</code> .....	7, 87, 90	<code>\initChapAfterSolns</code> .....	<u>4352</u>
<code>\ifkeyOrkeyalt</code> .....	91	<code>\inlineans</code> .....	<u>3786</u>
<code>\ifKV@eqefillLines@bgonly</code> .....	3110, 3118,	<code>\input</code> .....	1108, 1109, 1833,
	3128, 3193, 3212, 3234, 3408, 3429, 3484, 3659		1889, 1948, 1955, 1965, 1974, 3540, 4296, 4301
<code>\ifKV@eqefillLines@outlineonly</code> .....	3109,	<code>\InputIfFileExists</code> .....	20,
	3138, 3148, 3189, 3235, 3276, 3419, 3433, 3661		103, 136, 244, 249–252, 4420, 4514, 4538, 4563
<code>\ifKV@eqefillLines@topline</code> ..	3166, 3213, 3373, 3436	<code>\inputlineno</code> .....	3704, 3771
<code>\ifKV@eqFillin@autolift</code> .....	4720, 4816, 4861	<code>\inputRandomizeChoices</code> .....	135, 136
<code>\ifKV@eqFillin@boxed</code> .....	4719, 4730,	<code>\insert@circlesymbol</code> .....	1014
	4739, 4748, 4750, 4781, 4783, 4789, 4796, 4831	<code>\insertContAnnot</code> .....	2443, 2927
<code>\ifKV@eqFillin@enclosesoln</code> .....	4843	<code>\insertGrayLetters</code> .....	996, 1008, 1058, 1070
<code>\ifKV@eqFillin@fitwidth</code> .....	4779	<code>\insertpageifcarryover</code> .....	<u>4246</u>
<code>\ifKV@eqFillin@underline</code> .....	4770, 4773	<code>\insertPointsBoxPDF</code> .....	<u>1578</u> , 1605
<code>\ifl@stDispl@yPoints</code> .....	520, 2434	<code>\insertTotalsBoxPDF</code> ...	<u>1578</u> , 1615, 1622, 1673, 1682
<code>\ifmarginsonleft</code> .....	<u>4135</u> , 4137, 4155, 4258	<code>\insMargHead</code> .....	<u>4104</u>
<code>\ifmmode</code> .....	4596, 4744, 4790, 4809	<code>\insMidMarg</code> .....	2131, 3824, 3829, 3835, 3839,
<code>\ifnosolutions</code> .....	356		3904, 3931, 3934, <u>4076</u> , 4100, 4101, 4105, 4111
<code>\ifObeyPTsStar</code> .....	2568, 2577	<code>\insProbHead</code> .....	<u>4104</u> , 4477
<code>\ifOKToWriteExamData</code> ...	305, 321, 1219, 1222, 1231	<code>instructions (environment)</code> .....	<u>2027</u>
<code>\ifpdf</code> .....	115	<code>\instructionsColor</code> .....	<u>525</u>
<code>\ifpreview</code> .....	239, 354, 400	<code>\insTxtFieldIdInfo</code> .....	33, 879, 888, 898, 901
<code>\ifshowlsols</code> .....	3809	<code>\intPrt</code> .....	3345, 3362, 3385, 3390
<code>\ifsolutionsafter</code> .....	230, 394	<code>\is@bgonly</code> .....	3108, 3110, 3117, 3125, 3133
<code>\ifsolutionsAtEnd</code> .....	72, 1503	<code>\is@outlineonly</code> .....	3108, 3109, 3137, 3145, 3153
<code>\ifsolutionsonly</code> .....		<code>\is@uto</code> .....	2590, 2649
	233, 293, 304, 320, 522, 1222, 1230, 4411, 4414	<code>\iscarryoverfalse</code> .....	4165, 4210
<code>\ifsp</code> .....	4707, 4768	<code>\iscarryovertrue</code> .....	4211
<code>\iftb@shipoutPermitted</code> .....	4185, 4188	<code>\isInExamEnv</code> .....	489, 2225, 2235
<code>\iftherearesolutions</code> .....	4413	<code>\isinlineansfalse</code> ...	3820, 3828, 3830, 3838, 3840
<code>\ifuseNumForParts</code> .....	2885	<code>\isinlineansttrue</code> .....	3823, 3825, 3834, 3836
<code>\ifuserectforms</code> .....	1194	<code>\isinstredfalse</code> .....	3812, 3818
<code>\ifvoid</code> .....	3244, 3555, 3581, 4210, 4235	<code>\isit@uto</code> .....	2591, 2592, 2650
		<code>\isItD@</code> .....	2584, 2656, 2687



`\llap` ..... 1158, 1243, 2342, 3167, 3169, 3247,  
           3582–3584, 3586, 3590, 3597–3599, 3601, 3605  
`\load@exerquiz` ..... 122, 142, 377, 379, 4404  
`\load@web` ..... 120, 122, 141, 369, 370, 560  
`\longTitleText` ..... 592  
`\loop` ..... 710, 726, 790, 2889  
`\lower` ..... 1162, 1209, 4090  
`\lowercase` 729, 793, 1879, 1884, 1987, 1992, 2512, 2887  
`lsol` (environment) ..... 4055  
`\lsols` ..... 3786

## M

`\makeAnsEnvForSolnsAtEnd` ..... 279, 284  
`\makeatletter` ..... 4513  
`\makeatother` ..... 4524  
`\makeDoNum` ..... 2403, 2804, 2808  
`\makeOutOfNum` ..... 2407, 2804, 2805  
`\makeRefsNums` ..... 1275, 1283  
`\makeRoomForProb` .....  
           2234, 2424, 2458, 2526, 2631, 4484, 4494  
`\maketitle` ..... 1322  
`\maketitledesign` ..... 1322  
`\makeVgrid` 1781, 2973, 2975, 2977, 2979, 2981, 2983,  
           3025, 3113, 3192, 3196, 3204, 3219, 3222, 3231,  
           3242, 3243, 3622, 3625, 3666, 3671, 3747, 3752  
`\manswers` ..... 288  
`\manswers@sq` ..... 288  
`\manualcalcparts` ..... 91, 2657, 2785  
`\marginans` ..... 3786  
`\marginboxdesign` .....  
           ... 53, 1547, 1605, 1615, 1622, 1629, 1673, 1682  
`\marginparafterhook` 2537, 2538, 2782, 2823, 2833, 2866  
`\marginparpriorhook` ..... 2539, 2826, 2837, 2867  
`\marginparsep` ..... 442, 1574, 1602,  
           1612, 1625, 1674, 1680, 4138, 4143, 4156, 4254  
`\marginparwidth` ..... 443, 4125  
`\marginpoints` 154, 483, 485, 487, 491, 1544, 2528,  
           2529, 2813, 2816, 2819, 2831, 2838, 2853, 2863  
`\marginpointsboxtext` .. 1576, 1606, 1616, 1623, 1629  
`\marginpointtext` ..... 54, 1562, 1573  
`\marginsonleft` ..... 3786  
`\marginsonleftfalse` ..... 4135  
`\mark` ..... 1541, 1542, 1687, 2131, 4120  
`\markEndFor` ..... 1752  
`\markEndtFor` ..... 1751  
`\markerTotalFmt` ..... 1715  
`\markNumQsFor` ..... 1751  
`\markStartFor` ..... 1751  
`\MarParBoxFmt` ..... 4076, 4202, 4216  
`\marparboxwidth` ..... 4123, 4128, 4132, 4133

`max` (option) ..... 8  
`\maxdepth` ..... 3546, 3550, 3560  
`\MCcolor` ..... 1202, 1203  
`\measurePtBoxHt` ..... 56, 1628, 1637  
`\medbreak` ..... 4452, 4463  
`\medskipamount` ..... 4483, 4493  
`\MidMargcolor` ..... 4084  
`\midMargFmt` ..... 4085, 4087  
`\minipage` ..... 1983, 1986, 1996, 2003, 3512  
`\month` ..... 855, 857, 859  
`\mp@Width` ..... 1892, 1942, 1986  
`\mrgDecPt` ..... 3971  
`\mrgDigitFmt` ..... 3951  
`\mrgNumPrtsep` ..... 3971  
`\mrgPartFmt` ..... 3953  
`\mrgPrtsep` ..... 3971  
`\multicolsep` ..... 414  
`\multiply` ..... 766,  
           785, 2289, 2698, 2703, 3367, 3371, 3387, 3392  
`myconfig` (option) ..... 5  
`myconfigi...myconfigvi` (option) ..... 5

## N

`\nbaselineskip` ..... 81, 2494, 4503  
`\nbaselineskipReset` ..... 2495, 2496  
`\nDoNum` ..... 2400, 2402, 2403, 2632  
`\needsModArith` ..... 750, 752, 761  
`\NeedsTeXFormat` ..... 4550  
`\neutralizeparfixes` ..... 2619  
`\newbox` ..... 479, 2992–2994, 4042–4044  
`\NewCommentCutFile` ..... 4285, 4291, 4302  
`\newdimen` ..... 1128, 1130, 1839, 1840, 3477, 3987  
`\newsavebox` ..... 1127, 1762, 1838  
`\newwrite` ..... 1815, 4277  
`\nExam` ..... 574, 582–585  
`\nlastItem` ..... 2775, 2780  
`\nLocalSelection` .....  
           .... 749, 751, 768, 778, 779, 781, 782, 787, 795  
`\nLocalVersions` 749, 762, 765, 766, 781, 784, 785, 804  
`\nNumberOfP@rts` ..... 2219, 2248, 4451  
`\nobreak` ..... 1092, 1994, 3897, 4098  
`\noExamTitleInSolns` ..... 2223  
`noHiddensolutions` (option) ..... 9  
`nohiddensolutions` (option) ..... 9  
`\nointerlineskip` ..... 1755  
`\nolinkcolor` ..... 525  
`nomarginwrite` (option) ..... 4  
`noparttotals` (option) ..... 5  
`\NoPoints` ..... 36, 489, 492  
`nopoints` (option) ..... 5





preview	9	\panelheight	64, 1805, 1848, 1919
proofing	10	\panelwidth	64, 1805, 1809, 1847, 1920, 1989, 1998
rendition	8	\paperheight	450, 4150
showgrayletters	10	\paperwidth	447, 4140, 4145, 4324, 4325
solutionsafter	9	\paragraph	2023
solutionsonly	9	\parbox	4568
textures	11	\parindent	1409, 1417, 1612, 1758, 3197, 4077, 4335
totalsonleft	5	\parsep	2204, 3948, 4334, 4377, 4395
totalsonright	5	\parsetotals	56, 1641, 1644
usecustomdesign	4	\parshape	2755, 2757, 3724
useforms	7	\parskip	1475, 1502, 3197, 3947, 4079
usexkv	8	\partialspillovertotals	2086, 2092, 2094
vspacewithsolns	7	\partialtotaloe	2093, 2096, 2160, 2168
webOpts	9	\partialtotalpg	2091, 2096, 2161, 2169
xetex	11	\partNames	476, 1693, 2244, 2245
\optsCbBxf	1020	\partopsep	3947
\optsCbBxl	1021	\parts@indent	2740, 2744, 2908
\optsFillIn	2346	\partsitemsep	2501
\optsMlTextField	37, 1082	\partsleadinindent	2745, 2769
\optsRadioBtnf	957	\partsparsep	2501
\optsRadioBtnl	958	\partstabcolsep	2502
outlineonly* (key)	101	\partstabrowsep	2503
\OutOfNum	2409, 2411, 2413, 2633	\partstabtopsep	2503
		\partstopsep	2501
		parttotalsonleft (option)	5
		parttotalsonright (option)	5
		\passedss@Argii	2007, 2008, 2014
		\PassOptionsToPackage	158,
			199, 209, 211, 213, 216, 219, 221, 224, 226, 4553
		\PBS	963, 992, 1027, 1054
		pdf (option)	7
		pdftex (option)	11
		\penalty	2266, 4097
		\percentForPart	1699
		\phantom	4440, 4843, 4858, 4869
		\placeAtxy	1754
		\placeCoverPageLogo	1350
		\placeMarkerHere	1704
		\pnpDflt	2456, 2457
		\pointLabel	1560, 1561
		\pointsAsText	1562
		\pointsLabel	1555
		\pointsmarginparpush	80, 2135, 2459, 2471, 2486
		pointsonboth (option)	5
		\PointsOnBothSides	35, 486
		\PointsOnLeft	33, 483
		pointsonleft (option)	5
		\PointsOnRight	34, 484
		pointsonright (option)	5
		\popEnvir	3725

## P

\p@ssToFLs	2011, 3177,
	3182, 3185, 3186, 3667, 3674, 3727, 3748, 3754
\p@ssToSolns	3651, 3680, 3728, 3773
\PackageError	267, 717, 1859, 2053, 2417, 2669,
	2725, 3847, 3854, 3866, 3875, 4130, 4518, 4542
\PackageInfo	78, 104,
	371, 707, 1277, 2611, 2614, 4249, 4515, 4539, 4775
\PackageWarning	18, 21, 65, 106, 492,
	589, 651, 2397, 2595, 2599, 3032, 3057, 3069,
	3075, 3121, 3129, 3141, 3149, 3244, 3459, 4420,
	4584, 4593, 4606, 4616, 4648, 4676, 4680, 4760
\PackageWarningNoLine	
	74, 257, 364, 1111, 3463, 3700, 3767
\pagegoal	2144, 2181, 2427, 2450, 2454
\pagenumbering	1356, 1360
\pageref	1258, 1259
\pagestyle	3781
\pagetotal	
	2142, 2181, 2427, 2450, 2454, 2462, 2479, 2782
\panel	64, 1850
\panel@cnt	64
\panel@write	
	1815, 1862, 1864, 1865, 1870, 1871, 1877, 1885
\panelgap	64, 1837, 1893, 1943, 1988, 1998, 2003
\panelGetDimen	1832, 1879, 1888, 1947, 1954, 1964, 1973

<code>\popProblem</code> .....	2775	<code>\proofingsymbolColor</code> .....	525
<code>\post@parts@hook</code> .....	4030	<code>\protected@edef</code> .....	4477, 4712, 4716
<code>\postChapSolnHead</code> .....	4459, 4461	<code>\protected@xdef</code> .....	643, 647, 4108, 4109
<code>\postdisplaypenalty</code> .....	3535	<code>\providecommand</code> .....	195,
<code>\postExamSolnHead</code> .....	1215, 4452		196, 425, 427, 429, 435, 525, 527, 529, 531,
<code>\postPNPAction</code> .....	2422, 2423, 3223		533, 535, 537, 539, 541, 543, 545, 548, 568,
<code>\postSubmitJS</code> .....	923, 934		571, 845, 1555, 1556, 2190, 2339, 3961–3964, 3970
<code>\pr@b@secondarg</code> .....	2636, 2654, 2685, 2686	<code>\ProvidesFile</code> .....	1125, 3784
<code>\pr@bl@m@star</code> .....	2623, 2624	<code>\ProvidesPackage</code> .....	4551
<code>\pr@blem@star</code> .....	2625, 2626	<code>\prtsIndntSep</code> .....	3273, 3275
<code>\prbDecPt</code> .....	425, 3958	<code>\ps@eqExamheadings</code> .....	461
<code>\prbNumFmt</code> .....	4001	<code>\ptLabel</code> .....	1556
<code>\prbNumPrtsep</code> .....	429, 3958	<code>\PTs</code> .....	2537, 2538, 2563
<code>\prbPrtsep</code> .....	427, 3958	<code>\ptsLabel</code> .....	1555
<code>\preChapSolnHead</code> .....	4457, 4461	<code>\pushButton</code> .....	927
<code>\predisplaypenalty</code> .....	3534	<code>\pushEnvir</code> .....	3722
<code>\preExamSolnHead</code> .....	1215, 4452	<code>\pushProblem</code> .....	2775
<code>\presets</code> 975, 1009, 1039, 1071, 1092, 2372, 4892, 4897		<code>\put</code> .....	4231, 4233
<code>preview (option)</code> .....	9		
<code>\previewfalse</code> .....	239		
<code>\previewtrue</code> .....	163, 995, 999, 1057, 1061		
<code>\prior@parts@hook</code> .....	4027		
<code>\priorexsectitle</code> .....	4419		
<code>\priorexslinput</code> .....	4419		
<code>\priorP@geBre@kMsg</code> .....	3019, 3020, 3222, 3232		
<code>\priorPageBreakMsg</code> .....	3019, 3022		
<code>\priorPNPAction</code> .....	2421, 2423, 3220		
<code>\priorSubmitJS</code> .....	922, 931		
<code>\priorw@Msg</code> .....	3463, 3466, 3467		
<code>priorworkarea (environment)</code> .....	3714		
<code>\priorWorkAreaCmds</code> .....	1773–1775, 3519, 3539		
<code>\priorworkareaCmds</code> .....	1772		
<code>\prob@Arg</code> .....	2783, 2786		
<code>\problem</code> .....	2512		
<code>problem (environment)</code> .....	2497		
<code>problem* (environment)</code> .....	2594		
<code>\problem@cont</code> .....	2516, 2517		
<code>\probpointseach</code> .....	2638, 2651, 2678, 2679, 2791		
<code>\probSet</code> .....	4471, 4475		
<code>probset (environment)</code> .....	4474		
<code>\probstar</code> .....	948, 950,		
	968, 1003, 1032, 1065, 1086, 2060, 2364, 2384,		
	2390, 2527, 2635, 3846, 3865, 3916, 4883, 4909		
<code>\probvalue</code> .....	1537, 2540, 2827		
<code>\ProcessComment</code> .....	4294, 4303		
<code>\ProcessOptions</code> .....	4554		
<code>\ProcessOptionsX</code> .....	253		
<code>\promoteNewPage</code> .....	2456, 2905		
<code>proofing (option)</code> .....	10		
<code>\proofingsymbol</code> 39, 285, 1144, 1178, 1198, 1201, 2634			

## Q

<code>\Q</code> .....	1582, 1588
<code>\qeSumryVert</code> .....	48
<code>\qNewPage</code> .....	2928
<code>\quiz@solns</code> .....	294, 295,
	297, 306, 322, 323, 1215, 1223, 1225, 1233, 4277

## R

<code>\r</code> .....	929–931, 933
<code>\radio@@Button</code> .....	975, 1009
<code>\RadioFieldSize</code> .....	976, 1010, 1011, 1017,
	1040, 1072, 1079, 1129, 1161, 1208, 1210, 1211
<code>\raggedbottom</code> .....	470
<code>\raggedcolumns</code> .....	414
<code>\raggedright</code> .....	963, 992, 1027, 1054, 4079
<code>\raise</code> .....	3438, 3583, 3598
<code>\raisebox</code> ....	904, 1157, 1201, 1337, 1352, 1604,
	1614, 1621, 1672, 1681, 1759, 3423, 3587, 3591,
	3602, 3606, 4820, 4823, 4867, 4868, 4891, 4896
<code>\RecordThisExamOff</code> .....	74, 2227
<code>\Rect</code> .....	39, 1141, 1204
<code>\refstepcounter</code> ...	962, 991, 1026, 1053, 2735, 2934
<code>\removelastparskip</code> .....	1778, 1796
<code>\removelastskip</code> .....	2074, 2146
<code>rendition (option)</code> .....	8
<code>\repeat</code> .....	715, 743, 804, 2893
<code>\RequirePackage</code> .....	2, 3, 264,
	357, 360, 374, 380, 402–405, 412, 417, 4555, 4566
<code>\reset@color</code> .....	3248
<code>\resetacvspace</code> .....	2897
<code>\resetFillerCustomBg</code> .....	108, 3349, 3350

<code>\resetMargHeadColor</code> .....	4097	<code>\selectVersion</code> .....	748
<code>\RestoreCommentCutFile</code> .....	4286, 4296, 4301	<code>\selVersion</code> .....	27, 706
<code>\restoreFromChapAfterSolns</code> .....	4356	<code>\separationrule</code> .....	54, 2151, 2155, 2185
<code>\restorelabel</code> .....	4531	<code>\separationruleOff</code> .....	53, 506
<code>\restoreLastBotMargin</code> .....	4074	<code>\separationruleOn</code> .....	505
<code>\restoreLastTopMargin</code> .....	4072	<code>\ServerRetnMsg</code> .....	34, 918, 919
<code>\restorele@d infixDef</code> .....	2608	<code>\set@display@protect</code> .....	1224, 1232, 1237
<code>\restorele@d inpf ixDef</code> .....	2607, 2609	<code>\set@typeset@protect</code> .....	1226, 1235, 1239, 4205
<code>\restorePageLayout</code> .....	4338	<code>\setBotMargin</code> .....	130, 4068, 4074
<code>\ReturnTo</code> .....	39, 1143	<code>\setbox</code> 1152, 1628, 3202, 3468, 3469, 3532, 3552–	
<code>\reversemarginpar</code> .....	3806	3554, 3565, 3567, 3568, 3578–3580, 3594, 3610,	
<code>\reverseVSWs</code> .....	341	3613, 3763, 3764, 4198, 4209, 4213, 4223, 4224,	
<code>\rfooteqe</code> .....	1301	4226, 4692, 4792, 4810, 4817, 4834, 4855, 4862	
<code>\rhead</code> .....	1290	<code>\setBtwnExamSkip</code> .....	315, 318
<code>\rheadeqe</code> .....	1284	<code>\setcounter</code> 57, 611, 704, 714, 724, 777, 1649, 1651,	
<code>\rheadSol</code> .....	1308	1710, 1713, 1717, 1725, 1731–1733, 2118–2120,	
<code>\rightmargin</code> .....	4336	2166, 2542, 2636, 2684, 2735, 2780, 2793,	
<code>\rightmark</code> .....	4260, 4262, 4328	2801, 2855, 2870, 2874, 2876, 2881, 2886, 4425	
<code>\rlap</code> .....	1159, 1201,	<code>\setDefaultfvsizeskip</code> .....	2213, 4502
1243, 3171, 3404, 3405, 3411, 3414, 3418, 3422,		<code>\setDefaultnbaselineskip</code> .....	2492, 2493
3430, 3432, 3587, 3591, 3602, 3606, 4837, 4840		<code>\setfillinDefaults</code> .....	152, 4670
<code>\rowsepDefault</code> .....	2502	<code>\setFillLinesFmt</code> .....	101, 3101
<code>\rule</code> .....	1404, 2186, 3001, 3011, 3247, 3412	<code>\setFullWidthHeader</code> .....	4252, 4357
<code>\runExamFooter</code> .....	45, 467, 1307	<code>\setFullWidthLayout</code> .....	4320
<code>\runExamHeader</code> .....	1295	<code>\setMarIndents</code> .....	3977
<code>\runExamHeaderSol</code> .....	1308	<code>\setmulticolprob</code> .....	415
		<code>\setSolnIndent</code> .....	3987
		<code>\setSolnMargins</code> .....	301, 308, 3995
		<code>\setTopMargin</code> .....	130, 4060, 4072
		<code>\settobox</code> .....	467, 1643
		<code>\SetUpCutFile</code> .....	4291
		<code>\sExam</code> .....	576, 607, 625, 647
		<code>\shortquiz</code> .....	2259, 2260
		<code>\shortstack</code> .....	1325–1327
		<code>\shortTitleText</code> .....	592
		<code>\shortVersionAtext</code> .....	578
		<code>\shortVersionBtext</code> .....	578
		<code>\shortwebsubject</code> .....	825, 826, 1285, 1309
		<code>\shortwebtitle</code> .....	817, 818, 1285, 1309
		<code>\showAllAnsAtEnd</code> .....	278
		<code>showgrayletters</code> (option) .....	10
		<code>\showproofing</code> .....	10, 188
		<code>\sloppy</code> .....	3730
		<code>\slshape</code> .....	4257, 4260, 4262, 4327, 4328
		<code>\small</code> .....	1568, 1570, 1576, 1660
		<code>\smallskip</code> .....	1979, 4454
		<code>\smash</code> .....	1157,
		1352, 1799, 3167, 3169, 3171, 3247, 3396, 3398	
		<code>\solAtEndFormatting</code> .....	291
		<code>\solDecPt</code> .....	3965

## S

<code>\S</code> .....	954
<code>\sameVspace</code> .....	1933, 3252, 3254
<code>\save@message</code> .....	673, 678, 725, 744, 755, 789, 805, 809, 3506
<code>\save@setckpt</code> .....	4511, 4528
<code>\save@wlog</code> .....	3505
<code>\save@writefile</code> .....	4509, 4527
<code>\saveBasicLayoutParams</code> .....	4321, 4345
<code>\savedeq@online</code> .....	347, 392
<code>\savedifeq@globalshowsolutions</code> .....	350, 396
<code>\savedifeq@hidesolution</code> .....	349, 395
<code>\savedifeq@nolink</code> .....	353, 399
<code>\savedifeq@nosolutions</code> .....	351, 397
<code>\savedifeq@proofing</code> .....	352, 398
<code>\savedifeq@solutionsafter</code> .....	348, 393
<code>\savedifeqforpaper</code> .....	355, 401
<code>\savedifpreview</code> .....	354, 400
<code>\saveIFEQE</code> .....	329, 332–335
<code>\sbox</code> .....	964, 1028, 1842
<code>\scriptsize</code> .....	3422
<code>\sectionColor</code> .....	548, 550, 561
<code>\selectedMC</code> .....	262, 1140, 1148, 1165, 1175, 1190

<code>\soln@keys@nLines</code> .....	<code>\ssols</code> .....	3786
..... 1918, 1923, 1927, 2006, 2008, 3259, 3263	<code>\stepcounter</code> .....	621,
<code>\solnGutter</code> .....	654, 711, 727, 791, 961, 962, 989, 990, 1024,	
<code>\solnhspace</code> .....	1025, 1051, 1052, 1861, 2236, 2362, 2890, 4881	
<code>\solnItemMngt</code> .....	<code>\stretch</code> .....	1388, 1438, 1440
<code>\solnsAtEndcomment</code> .....	<code>\strip@pt</code> .....	3360, 3368, 4151, 4154, 4157
<code>\solNumPrtsep</code> .....	<code>\stripeqExam</code> .....	56, 1642, 1648
<code>\solPrtsep</code> .....	<code>\strut</code> .....	1325–1327, 1799, 1993, 1994,
<code>\solution</code> .....	2001, 2074, 2075, 2082, 2354, 2356, 2898, 3023,	
<code>\solutionparshape</code> .....	4583, 4587, 4799, 4812, 4821, 4847, 4859, 4870	
1982, 2719, 2721, 2756, 2759, 2912, 3266, 3721	<code>\styleComm</code> .....	2073, 2075, 2078
<code>\SolutionsAfter</code> .....	<code>\styleInstr</code> .....	2030, 2038
4490, 4500	<code>\subject</code> .....	811
<code>solutionsafter</code> (option) .....	<code>\subjectColor</code> .....	525
<code>\solutionsafterfalse</code> .....	<code>\submitButton</code> .....	34, 924, 1337
<code>\solutionsafterSkip</code> .....	<code>\SubmitButtonLabel</code> .....	34, 920, 921
1824, 1835, 2004	<code>\SubmitInfo</code> .....	33, 909
<code>\solutionsaftertrue</code> .....	<code>\summaryPointTotal</code> .....	70, 2033, 2045, 2046, 2048
84, 421–423, 512	<code>\SummaryTotalsOff</code> .....	502
<code>\solutionsAtEndfalse</code> .....	<code>\SummaryTotalsOn</code> .....	501
<code>\solutionsAtEndtrue</code> .....	<code>\summaryTotalsTxt</code> .....	70, 2041, 2046
72, 420	<code>\sumryAnnots</code> .....	61–63, 67, 1442
<code>solutiononly</code> (option) .....	<code>\symbolchoice</code> .....	378, 385, 1177, 1197, 1200
<code>\solutiononlyfalse</code> .....		
<code>\solutiononlytrue</code> .....		
185		
<code>\solWoPrtsFmt</code> .....		
4001		
<code>\solWPrtsFmt</code> .....		
4001, 4034, 4035		
<code>\solWPrtsFmt@hang</code> .....		
4019, 4034		
<code>\spaceskip</code> .....		
4079		
<code>\SpaceToWork</code> .....		
146		
<code>\splitbotmark</code> .....		
4236, 4238		
<code>\splitmaxdepth</code> .....		
3550, 3551		
<code>\splitsolution</code> .....		
1803		
<code>splitsolution</code> (environment) .....		
63, 1803		
<code>\splitsolutioni</code> .....		
1804, 1807		
<code>\splitsolutionii</code> .....		
1808, 1811		
<code>\splitsolutioniii</code> .....		
1805, 1809, 1812, 1814		
<code>\splittopskip</code> .....		
3549, 3576		
<code>\sq@priorhook</code> .....		
2125, 4441		
<code>\sqForms</code> .....		
144		
<code>\sqlabel</code> .....		
2309		
<code>\sqlinks</code> .....		
143		
<code>\sqsllabel</code> .....		
1214		
<code>\sqslrtnlabel</code> .....		
2332		
<code>\sqsolafter</code> .....		
2311		
<code>\sqstar</code> .....		
112, 143, 144, 2259, 2260		
<code>\square</code> .....		
2328–2330, 4488, 4498		
<code>\ss@Argi</code> ...		
1827, 1829, 1856–1858, 1944, 1984, 2009		
<code>\ss@Argii</code> .....		
1816,		
1866, 1872, 1903, 1907, 1915, 1924, 1928,		
1930, 1932, 1936, 1939–1941, 1991, 2007, 2008		
<code>\ss@Argiii</code> .....		
1816, 1827, 1890, 1891, 1893,		
1943, 1946, 1953, 1963, 1971, 1984, 1989, 2003		
<code>ssol</code> (environment) .....		
4056		

## T

<code>\tabcolsep</code> .....	1548, 1601, 1611, 1671, 1679
<code>\tableadin</code> .....	90, 2602, 2660
<code>\tableadinWarningStar</code> .....	2599, 2647
<code>\tb@addtoMargin</code> .....	4051, 4095
<code>\tb@addtoTopMargin</code> .....	4047
<code>\tb@ANS</code> .....	124, 3882, 3883, 3908
<code>\tb@cargi</code> .....	4107, 4114, 4119
<code>\tb@beginexam@code</code> .....	2085, 2127, 3792
<code>\tb@carryoverFmt</code> .....	
4166, 4169, 4170, 4172, 4173, 4176, 4180, 4182	
<code>\tb@co@page</code> .....	4171, 4172
<code>\tb@csr</code> .....	4343, 4350
<code>\tb@EndOfChapterExercises</code> .....	4268, 4310
<code>\tb@esms</code> .....	4340, 4347
<code>\tb@GenProbNum</code> .....	4432, 4438, 4439
<code>\tb@insertCarryOver</code> .....	4178, 4243
<code>\tb@insertDecPoint</code> ....	4427, 4431, 4434, 4439, 4440
<code>\tb@insmargmark</code> .....	2130, 2178
<code>\tb@istart</code> .....	3882, 3883, 3911
<code>\tb@lws</code> .....	4342, 4349
<code>\tb@marginHeader</code> .....	4105, 4115, 4116, 4118
<code>\tb@marginProbHeader</code> .....	4111, 4113
<code>\tb@MHC</code> .....	4100, 4101, 4103
<code>\tb@midMargFmt</code> .....	4086, 4092, 4181
<code>\tb@mrqDigitFmt</code> .....	3925, 3951, 4431, 4433

<code>\tb@mrgPartFmt</code> .....	3889, 3897, 3954	<code>\textcolor</code> .....	983, 1047, 1141,
<code>\tb@next</code> .....	3853, 3860, 3862, 3872, 3874, 3880		1145, 1413, 2039, 2079, 2148, 2990, 2999, 3003,
<code>\tb@osms</code> .....	4339, 4346		3411, 3421, 3431, 3887, 3907, 3955, 3957, 4103
<code>\tb@rest@reMarginFmt</code> .....	4177, 4179, 4181, 4183	<code>\textField</code> .....	905,
<code>\tb@sbm@exp</code> .....	4236, 4237, 4242		936, 938, 940, 942, 944, 1092, 2371, 4892, 4897
<code>\tb@shipoutPermittedfalse</code> .....	4187	<code>\textheight</code> .....	450–455, 1420, 1426,
<code>\tb@shipoutPermittedtrue</code> .....	4185, 4186		2234, 2456, 4088, 4091, 4150, 4201, 4209, 4220
<code>\tb@showlwsols</code> .....	3809, 4055	<code>\textit</code> .....	2310, 2311
<code>\tb@showssols</code> .....	4056, 4057, 4058	<code>\textorpdfstring</code> .....	391
<code>\tb@so@next</code> .....	4190, 4192, 4193	<code>\textsf</code> .....	3346
<code>\tb@soln@choice</code> .....	3809, 4057, 4363	<code>\textsize</code> .....	955
<code>\tb@sols@choice</code> .....	4058	<code>textures (option)</code> .....	11
<code>\tb@tws</code> .....	4341, 4348	<code>\textwidth</code> .....	447–449, 1297, 1298,
<code>\tb@woparts@len</code> .....	3941, 3979		1318, 1319, 1324, 1335, 1399, 1419, 1452, 1602,
<code>\tb@wparts@len</code> .....	3900, 3943, 3982		1612, 1680, 4140, 4144, 4156, 4324, 4341, 4348
<code>\tballowAllNums</code> .....	3798	<code>\TF</code> .....	2380, 4906
<code>\tbBaseName</code> .....	4507, 4514, 4533, 4536, 4538, 4540	<code>\the@cntfillin</code> .....	2366, 2369, 4885, 4888
<code>\tbBotMargin</code> .....	4046, 4067, 4068, 4070, 4212	<code>\the@exno</code> .....	1101, 2873
<code>\tbcontinued</code> .....	4108, 4115, 4122	<code>\thebackofpage</code> .....	2934, 2936, 2938, 2961
<code>\tbfilterOutEvenNums</code> .....	3789, 3794	<code>\thechapter</code> .....	4458
<code>\tbblastpageshipped</code> .....	4189, 4191, 4195	<code>\thededate</code> .....	849, 851
<code>\tbMakeFinalCalcs</code> .....	3815, 4126, 4159	<code>\theeq@count</code> .....	2682,
<code>\tbMarginHeaderFmt</code> .....	4097, 4119, 4238		2705, 2795, 2798, 2803, 2807, 2810, 2857, 2860
<code>\tbmarparboxwidth</code> .....	3814, 4080, 4082, 4089, 4092,	<code>\theeq@numparts</code> .....	1539
	4124, 4125, 4130, 4138, 4143, 4200, 4215, 4254	<code>\theeqpointsthispage</code> .....	1655, 1661, 1662, 1668
<code>\tbminskipbtnlayers</code> .....	4163, 4204, 4222, 4227	<code>\theeqpointvalue</code> .....	1539, 1706, 2299
<code>\tbmrgpartwidth</code> .....	3888, 3896, 3984	<code>\theeqquestionnoi</code> .....	968,
<code>\tbplaceMargins</code> .....	4123, 4148, 4161		970, 1003, 1006, 1032, 1035, 1065, 1068, 1087,
<code>\tbPostMarginHeader</code> .....	4097, 4120, 4239		1089, 1581, 1720, 2058, 2061, 2064, 2068, 2301,
<code>\tbprbNumFmt</code> .....	4002, 4003		2312, 2314, 2318, 2321, 2342, 2365, 2369, 2642,
<code>\tbPreMarginHeader</code> .....	4097, 4118, 4237		2676, 2681, 2691, 2705, 2739, 2773, 3901,
<code>\tbSaveBotMargin</code> .....	4069, 4071, 4075		3911, 3926, 4002, 4006, 4008, 4403, 4884, 4888
<code>\tbSaveTopMargin</code> .....	4063, 4066, 4073	<code>\theexampleno</code> .....	4480, 4486, 4496
<code>\tbSetupForMargins</code> .....	4136, 4160	<code>\theGrandTotal</code> .....	1697
<code>\tbsolnpartwidth</code> .....	3992, 4016, 4022, 4394	<code>\theHeqquestionnoi</code> .....	2057
<code>\tbsolWoPrtsFmt</code> .....	4005, 4011	<code>\theHpartno</code> .....	2067
<code>\tbsolWPrtsFmt</code> .....	4007, 4014, 4020	<code>\theHquizno</code> .....	2059
<code>\tbSourceFile</code> ...	4508, 4516, 4519, 4520, 4537, 4543	<code>\themarkercnt</code> .....	1716
<code>\tbTopMargin</code> .....	4045, 4059, 4061, 4065	<code>\thepage</code> .....	1580,
<code>\temp@Exp</code> .....	2259		1587, 1590, 1654, 2124, 2546, 2849, 4171,
<code>\temp@exp</code> .....	733,		4172, 4250, 4257, 4260, 4263, 4327, 4328, 4472
	735, 736, 738, 797, 799, 1921, 1922, 4642, 4645	<code>\thepanel@cnt</code> .....	1833, 1862, 1889, 1948, 1955, 1965, 1974
<code>\texorpdfstring</code> .....	812	<code>\thePartNames</code> .....	1280, 1480
<code>\text</code> .....	1446, 1562, 1568, 1570, 1571, 1576,	<code>\thepartno</code> .....	968,
	1661, 2046, 2563, 2565, 4583, 4587, 4744, 4790		1004, 1033, 1066, 1087, 2061, 2068, 2322,
<code>\textbf</code> .....	1213, 1217, 1445, 2312, 2313,		2365, 3893, 3897, 3902, 4008, 4009, 4425, 4884
	2317, 2320, 2342, 2399, 2400, 2411, 2412, 2898,	<code>\thequizno</code> .....	948
	4004, 4013, 4018, 4103, 4453, 4462, 4486, 4496	<code>\therearesolutionstrue</code> .....	183
<code>\textbookOpts</code> .....	3786	<code>\thereissolutionfalse</code> .....	2522, 2622
<code>\textColor</code> .....	955, 1582, 1588	<code>\thesection</code> .....	4475



## V

`\V` ..... 937, 939, 941, 943, 945  
`\vbadness` ..... 3653, 3690, 3717, 4078  
`\verbatim@out` ..... 1877,  
     1879, 1884, 1900, 1978, 3495, 3497, 3507, 3508,  
     3511, 3512, 3514, 3518, 3695, 3696, 3738, 3739  
`\verbatimwrite` ..... 1880, 3528  
`\version` ..... 833  
`\VersionAtext` ..... 578  
`\VersionBtext` ..... 578  
`\versionLabel` ..... 839, 840  
`\vfGo@l` ..... 3218, 3225, 3226, 3229, 3250  
`\vfil` ..... 3589, 3593, 3604, 3608, 3609, 4207, 4228  
`\voidb@x` ..... 3469, 3594, 3610, 3613, 3763, 3764  
`\vphantom` ..... 3005  
`\vpwsSimulateNoSolns` ..... 336, 346  
`\vrule` ..... 1138, 1550, 3006  
`\vsize` ..... 2182, 3588, 3592, 3603, 3607, 4201  
`\vspaceFiller` ..... 2965, 2967, 2970, 2982  
`\vspaceFillerDefault` ..... 2970  
`\vspaceFillerLines` .....  
     2966, 2967, 2982, 3175, 3261, 3264, 3324  
`\vspaceFillerLines@i` ..... 3183, 3187  
`\vspacewithkeyOff` ..... 515, 1122, 3689, 3729  
`\vspacewithkeyOn` ..... 515  
`\vspacewithsolns` (option) ..... 7  
`\vspacewithsolnsfalse` ..... 86, 89, 341, 3645  
`\vspacewithsolnstrue` ..... 88, 98, 281, 3631  
`\vsplit` ..... 3553, 3567, 3579, 4209, 4223  
`\vss` ..... 3007, 3010

## W

`\W` ..... 954  
`\w@rningBadOpts` ..... 74, 83, 99, 172, 180, 186  
`\web@continued` ..... 842  
`\web@save@maketitle` ..... 1342, 1345  
`\web@save@title` ..... 813, 818  
`\web@saved@author` ..... 819, 822  
`\web@sectionsColorOld` ..... 548  
`\web@toc` ..... 841  
`\web@version@value` ..... 833, 834, 836, 838  
`\web@versionlabel` ..... 835, 839  
`\webArg` ..... 817, 824, 825  
`\webauthor` ..... 820, 1327, 1413  
`\webauthor@color` ..... 537, 1413  
`\webcopyrightyears` ..... 832  
`\webemail` ..... 828  
`\webkeywords` ..... 829  
`\webnewpage` ..... 4410, 4416, 4417, 4447  
`\webOpts` (option) ..... 9

`\websubject` ..... 826, 913, 1325, 1364, 2337, 2339  
`\websubject@color` ..... 539, 1430  
`\webtitle` ..... 815, 915, 1326, 1437, 2333  
`\webtitle@color` ..... 535, 1436  
`\webuniversity` ..... 831, 1425  
`\webuniversity@color` ..... 533, 1424  
`\webversion` ..... 834  
`\widowpenalty` ..... 3653, 3716  
`\widthOfParts` ..... 3279  
`\widthtpboxes` ..... 53, 1546, 1548, 1551, 1583, 1591  
`\WithinANSGrpfalse` ..... 3843, 3852, 3871  
`\WithinANSGrptrue` ..... 3858  
`\withinsoldocfalse` ..... 2188  
`\withinsoldoctrue` ..... 300  
`\wlog` ..... 3505  
`\wlVspace` ..... 1926,  
     2445, 2986, 2987, 3187, 3198, 3225, 3226, 3233,  
     3241, 3257, 3262, 3324, 3357, 3359, 3374, 3384,  
     3391, 3394, 3416, 3545, 3549, 3576, 3591, 3606  
`\work@reaCmds` ..... 1769–1771, 1790  
`workarea` (environment) ..... 1762  
`\workareaCmds` ..... 1768  
`\workareadepth` ..... 1779, 1787  
`\workareاسب` ..... 61, 1762, 1780, 1799  
`\workareaV@dj` ..... 1765, 1788  
`\workareaVadj` ..... 61, 1763, 1767  
`\write` ..... 277,  
     3495, 3497, 3511, 3512, 3514, 3518, 3695, 3738  
`\writeAllAnsAtEnd` ..... 293, 1118, 1119, 4279  
`\writeallsolutions` ..... 4282  
`\writeBeginEqeQuestions` ..... 304, 1116, 2121  
`\writeEndEqeQuestions` ..... 319, 2174  
`\writelastpage` ..... 1694, 2177, 2303  
`\writeT@SolnFile` ..... 300, 308–310, 325,  
     1228, 2228, 2229, 2663, 2723, 4029, 4031, 4456  
`\writeToSolnFile` ..... 1215  
`\writetotalstoaux` ..... 2175, 2298  
`\writeVertic@lFLinesfalse` ..... 3624, 3669, 3750  
`\writeVertic@lFLinestrue` ..... 3175, 3638, 3676, 3756  
`\writeWithSolDocTrue` ..... 300, 1120  
`\wrtChapSolnHead` ..... 4455  
`\wrtExamTitleInSolns` ..... 2222

## X

`xetex` (option) ..... 11  
`\XKV@rm` ..... 1924, 1929, 1930  
`\xspaceskip` ..... 4079

## Y

`\y@st@Do` ..... 2584, 2588  
`\yest@D@` ..... 2584–2586, 2688, 2800

## 19 Change History

1.6a (2006/01/22)		file should contain one or more custom options. . . . .	12
\selectVersion: added \selectVersion command, also the command was fixed \forVersion so that it can be changed within the document, added switches to control new selection. . . . .	28	\eqEndExamTotalColor: Added easy user access to various colors, \proofingsymbolColor, \instructionsColor, \eqCommentsColor, \authorColor, \titleColor, \universityColor and \subjectColor . . .	21
1.6b (2006/04/03)		v1.6f (2006/10/24)	
\forVersion: Changed the alphabet environments A, B, etc. due to a conflict with \S, the control sequence for the amsmath package for a section: \S expands to §. Names changed to verA, verB, etc. . . . .	26	\forVersion: Fixed a bug in the \eqe@initializeMultiVersions command, made sure that any already defined comment environments are set to relax. . . . .	26
1.6b (2006/3/12)		v1.6g (2006/11/29)	
\popProblem: Added \pushProblem and \popProblem to grant the ability to interrupt a parts environment for, for example, a multicolumn environment. . . . .	91	\itemPTsFormatted: Added \itemPTsTxt and \itemPTsFormatted to work with \PTs. Also added a * option, to \PTs, in this case the points are not typeset. . . . .	83
1.6c (2006/05/02)		v1.6h (2007/01/24)	
\forVersion: At the end of the package, set the initial value of \selectVersion to \selectVersion{26} and set the initial value of \forVersion to \forVersion{A}. . . . .	28	\forleadinitem: Added \forproblem, \foritem, \aNewPage . . . . .	94
2.0a (2010/05/06)		\qNewPage: Added \aNewPage and \qNewPage . .	96
\rheadeqe: Changed the definitions of \lhead, \chead, and \rhead so they don't clash with the fancyhdr package. If fancyhdr is not loaded at the time eqexam is loaded, we \let the old names to the new names. Therefore, when fancyhdr is loaded first use the new definitions. . . . .	44	v1.6i (2007/09/18)	
4.4a (2015/06/01)		\altTitle: Added the command \altTitle as an alternate title for the exam document. This alternate title appear centered under the title of the document. . . . .	46
problem: Added the hook \topofprobstarhook .	83	v1.7a (2007/12/10)	
(2015/06/01)		\separationrule: Added \separationrule so user's can redesign the separation rule that is created between two parts of an exam. . . .	73
problem*: Added the hook \topofprobstarhook	86	v1.7b (2007/21/07)	
(2015/06/02)		General: Added a solutionsonly option . . . . .	9
problem*: Added \@markttotalvalue which seems to solve a problem with multicol. Sometimes the totals are not correct, they do not 'migrate' out as they should. . . . .	88	\encloseProblemsWith: Added \encloseProblemsWith to support the solutionsonly option . . . . .	21
(2015/07/21)		v1.7c (2008/08/21)	
General: add \let\CT@arc@\relax . . . . .	13	General: Added the showgrayletters option to eqexam (ported from exerquiz) . . . . .	10
v1.6e (2006/05/07)		v1.8 (2008/11/02)	
General: Added a custom option feature. Just before the options are processed, the tex compiler looks for the file eqecus.opt. This		\placeMarkerHere: Added a set of commands \placeMarkerHere, \calcFromMarkers, and \markerTotalFmt to enable the calculation of totals of segments of the exam. . . . .	59
		v1.9a (2009/28/09)	
		\coverpageTitleFmt: Added these various Fmt commands for coverage. . . . .	47



v1.9b (2009/09/29)	defined in <code>eqexam.def/exerquiz</code> . . . . .	20
\placeCoverPageLogo: Added		
\placeCoverPageLogo to insert a logo on the cover page. . . . .		47
v1.9f (2009/10/06)		
\endshortTitleText: Modified \longTitleText, \shortTitleText to have an optional argument (A–Z;a–z). You can select a particular title from a list of titles. If no optional argument is passed, then the title determined by \forVersion is used. . . . .		23
v2.0 (2010/03/05)		
General: Added <code>exambuilder.cfg</code> for use by AeB Exam Builder, to pass the values of the options <code>max</code> and <code>rendition</code> to <code>eqexam</code> . . . .		12
Switched over to <code>xkeyval</code> , added <code>max</code> and <code>rendition</code> to be consistent with the <code>renditions</code> package, though we don't use the <code>rendition</code> package itself. <code>eqexam</code> has a more extensive <code>renditions</code> system already. Introduced this mostly for use AeB Exam Builder. . . . .		8
v2.0c (2011/01/11)		
<code>problem*</code> : Changed <code>\@next</code> to <code>\eqe@next</code> . There was conflict in the use of this command with one of the float environments. When user used the table environment inside the <code>problem*</code> environment, the compiled stopped because <code>\@next</code> was overwritten. . . . .		84
v2.0d (2011/03/04)		
General: Added the <code>vspacewithsolns</code> option . . .		7
v2.0e (2011/03/07)		
\useFillerLines: Added the feature of filling the vertical space with ruled lines of different types. This feature is available for paper options and for <code>nosolutions</code> and <code>vspacewithsolution</code> options. . . . .		97
v2.0h (2011/04/14)		
\fillin: Modified the calculation of the width of \fillin, the width of enclosing box now equals the requested width . . . . .		156
v2.0i (2011/04/17)		
General: Added the switch <code>\ifdisplayworkarea</code> to better control when the work area is to be displayed. . . . .		10
v2.0j (2011/04/19)		
<code>enclosesoln</code> : added the <code>enclosesoln</code> key to <code>eqFillin</code> family. . . . .		148
v2.0k (2011/04/29)		
\vspacewithkeyOff: Added user interface to the switch <code>\ifkeepdeclaredvspacing</code> , which is		
v2.0l (2011/05/05)		
\firstPageOfExam: Returns the page number of the beginning of the exam with a given name. . . .		43
\lastPageOfExam: Returns the page number of the end of the exam with a given name. . . .		43
v2.0n (2011/05/13)		
\promoteNewPage: A simple variation on \makeRoomForProb designed for user use. . .		80
<code>problem</code> : Added <code>*&lt;num&gt;</code> to signal in-line display of points. . . . .		82
v3.0f (2011/08/13)		
\ftbInputBookAux: Added \ftbInputBookAux to support solution manual . . . . .		145
\ftbInputSolnFiles: Added \ftbInputSolnFiles to support solution manual . . . . .		147
v3.0g (2011/08/15)		
\hangSolWPrtsFmt: Use this to use “hanging indentation” for the parts for problems with parts in the solutions file. . . . .		128
v3.0h (2011/08/17)		
General: 2011/08/17 v3.0h Added the <code>vspacewithsolns</code> option . . . . .		7
v3.0i (2011/08/18)		
<code>fitwidth</code> : added the <code>fitwidth</code> key to <code>eqFillin</code> family. . . . .		149
v3.0l (2011/08/22)		
\annotContStr: Defined \eqe@insertContAnnot and related commands . . . . .		95
v3.0n (2011/09/18)		
<code>probset</code> : Added an <code>\edef</code> in case <code>\thesection</code> does not get expanded early enough to display correctly in the margins. . . . .		144
v3.0o (2011/09/20)		
\cngMargHeadColorTo: Added \cngMargHeadColorTo and \resetMargHeadColor to make it easier to change the color of the header globally, or just once. . . . .		131
v3.0p (2011/09/22)		
General: Added <code>\NoSolutions</code> to be executed in preamble, needed with the <code>fortextbook</code> package. . . . .		20
Added the wrapper package <code>fortextbook</code> . . .		147
v3.0q (2011/14/22)		
\eGrpANS: Added \bGrpANS and \eGrpANS . . .		121
v3.0s (2012/01/01)		
\eqexammargin: Moved a copy of \eqe@spannerSoln out of the <code>ftbsty</code> to the		

package section, its needed here as well. . . .	18	v3.2 (2012/12/10)	
v3.0t (2012/25/01)		splitssolution: Rewrote splitsolution and related code to change the syntax. . . . .	62
General: Added four more CFG files are the request of a user. . . . .	5	v3.2c (2013/02/23)	
v3.0u (2012/09/03)		General: Added \selectedMC at end of package .	13
General: Added the cfg option for inputting a custom config file. . . . .	5	v3.3 (2013/04/07)	
v3.0v (2012/03/14)		lsols: exerquiz changed \eqEXt to two variables, so we make the same change here. . . . .	120
\textbookOpts: Include \tbMakeFinalCalcs at end of \textbookOpts . . . . .	120	v3.3a (2013/05/03)	
v3.0w (2012/03/27)		\leadinitem: Changed name of \eq@writeexheader to \eqExerSolnHeader	90
parbox: Added parbox . . . . .	149	v3.5a (2014/12/19)	
hiddenbox: Added hiddenbox options . . . . .	149	General: Made usexkeys as default . . . . .	8
v3.0x (2012/04/03)		v3.6a (2015/01/31)	
hiddenbox: Added \eqe@align@hfill to align property. used to set position of content when parbox is used. . . . .	149	\forleadinitem: Added \eqe@fpmrk to add to the uniqueness of \theHeqqquestionnoi. . . .	94
v3.0y (2012/04/20)		v3.7 (2015/02/11)	
\altTitle: Moved \EQEcalculateAllTotals from the bottom to the top of \eqemaketitle. In case the author wants the grand total of the exam in the title, we need to make all calculations before \maketitledesign. . . . .	46	v3.7a (2015/02/26)	
v3.1 (2012/05/16)		\eqe@tb@shipout: Added a new Boolean switch \iftb@shipoutPermitted to turn off the shipout. Two convenience commands in support are also added, \turnOffFTBShipout and \turnOffFTBShipout. . . . .	134
\calcQsBtwnMarkers: Added \calcQsBtwnMarkers . . . . .	59	v3.8 (2015/03/17)	
v3.1a (2012/05/21)		\leadinitem: Spacing is not correct with standard itemize and enumerate environments, this is a fix. . . . .	90
problem*: Using a more robust method of detecting the presence of \auto . . . . .	86	v4.1 (2015/04/23)	
v3.1b (2012/06/18)		General: Moved this segment of code to \manualcalcparts, it is taken from the main problem* env. . . . .	92
General: Added \eqe@auto@chk@drivers . . . . .	7	v4.4 (2015/05/24)	
New requirement ifpdf . . . . .	4	exam: Finish with a \vskip to clear the last item or problem. . . . .	75
v3.1c (2012/07/30)		v4.5 (2015/11/10)	
ulcmd: Added ulcmd to \fillin . . . . .	150	\fillin: Added \setfillinDefaults . . . . .	152
boxcmd: Added boxcmd to \fillin . . . . .	150	v4.5a (2015/11/10)	
v3.1d (2012/09/29)		hiddenbox: Added some logic to the underline key . . . . .	149
General: Added \ifeqexamCFG and cseqexamCFG; these are used by ATB to attached the configuration file. . . . .	5	v4.6 (2016/01/18)	
v3.1e (2012/11/10)		\eqe@tb@shipout: Removed \vfil between unboxing, this allows correct breaking of margin content across pages. . . . .	135
General: Changed default for \proofingsymbol .	41	v4.6a (2016/06/04)	
\makeAnsEnvForSolsAtEnd: Removed hard-wired \proofingsymbol . . . . .	13	General: Added \ifequsecolor to easily distinguish between forpaper and forcolorpaper options. . . . .	9
problem*: Removed hard-wired \proofingsymbol . . . . .	86		
v3.1f (2012/11/27)			
\leadinitem: Created \leadinitem command for a lead in question for the parts* environment. . . . .	88		

v4.6b (2016/10/02)	v5.1.1 (2018/12/05)
\showAllAnsAtEnd: Include \eqTopOfSolnPage from exerquiz to support copying question to solution page. .... 13	General: Added \eqprior ..... 14
v4.6c (2017/01/28)	v5.1.2 (2018/12/05)
ulcmd: added custom underline option ..... 151	General: Use version 3.2 of comment style under the name of aeb-comment ..... 16
v4.7 (2017/01/04)	v5.1.3 (2018/12/13)
\forleadinitem: Modified \foritem@cont to allow for numbering parts ..... 94	General: \ifdisplayworkarea conditionally defined; defined \displayworkareaOn and \displayworkareaOff. Consistent with exerquiz. .... 10
v4.9 (2017/03/16)	workarea: added solutions-after condition to displaying workarea ..... 62
General: Updated eqexam to reflect changes in eqexam.def. .... 38	\postExamSolnHead: Write to solution file if not solutions-only ..... 42
v4.9a (2017/04/05)	v5.1.4 (2019/01/31)
\leadinitem: Added a test for solutions after within lead-in item ..... 90	General: \let \comment and \endcomment to \relax we use the definition of the comment environment given in verbatim. .... 16
v4.9b (2017/10/10)	v5.1.5 (2019/10/28)
\eqEndExamTotalColor: Provides color for check boxes when a PDF-related option is not taken 22	General: Define \eq@Radio@driver in case eqexam.def/exerquiz still use \eq@RadioCheck@driver ..... 36
v4.9c (2017/11/03)	Insert \bWebCustomize and \eWebCustomize to conform with web dated 2019/10/23. ... 12
\fillTypeGrid: Added \fillTypeGrid ..... 98	Replace \eq@RadioCheck@driver by \eq@Radio@driver ..... 36
v5.0 (2017/11/08)	v5.1.6 (2019/10/29)
General: Added \setFillLinesFmt ..... 102	\fillin: Allow the third argument of \filllineol* to have verbatim text as well. 153
Added hooks \priorPNPAction and \postPNPAction to \makeRoomForProb .... 79	v5.1.7 (2019/12/17)
Added support for \fillTypeGrid, added keys numbers, numbersep, and color as well .... 99	General: Added \autoInsSolns ..... 141
Added switch \ifgridpgbrk ..... 103	v5.1.8 (2020/01/06)
Added vertical rules ..... 99	General: Define \aebc@end as a work around for aeb-comment ..... 16
workarea: Added \workareaVadj ..... 61	v5.1a (2018/01/21)
Changes in workarea to support \fillTypeGrid ..... 62	problem: added \eqe@p@gobnxtpr ..... 82
Within workarea, adjust \leftskip when the problem is a lead-in type ..... 62	problem*: added \eqe@gobnxtpr ..... 85
v5.0a (2017/11/14)	v5.1b (2018/01/24)
General: Placed \eqe@makeVgrid in a group, \count0 was leaking out. .... 108	\vspaceFillerLines: Added ..... 105
v5.0c (2017/11/18)	v5.1c (2018/01/31)
General: Added the outlineonly, bgonly, and bgcolor keys to the eqeFileLines family. ... 101	General: Added additional lines to get a better break point ..... 112
v5.0d (2017/11/19)	v5.1d (2018/02/02)
General: Added a custom background feature . 108	\fillin: removed \space\ignorespaces from \fillin ..... 159
v5.0g (2017/12/05)	exam: Added \RecordThisExamOff ..... 74
General: Added filler lines with nosolutions option ..... 116	v5.1e (2018/02/09)
Added option flxextended. .... 7	problem: Support for h and H argument ... 82, 83
priorworkarea: Added the priorworkarea environment ..... 116	
v5.0o (2017/12/26)	
General: Added \chnGToNoSolns ..... 20	

