

eqexam: An Exam Construction Package

D. P. Story
Email: dpstory@uakron.edu

Processed January 23, 2021

Contents

1	Introduction	3
2	Package options and Process Options	4
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
3	Required Packages	16
4	Page Layout	18
5	Counters, Lengths and Tokens	19
6	Some Macros to Support the Options	20
7	Colors	21
8	Version Control	22
9	Title Definitions from Web	30
10	Identification Information	32
11	Stand alone Code	39
12	Switching proofing symbols	40

13 The Main Code	42
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	59
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	63
13.7 The <code>exam</code> Environment	71
13.8 <code>problem</code> Environments	79
14 Vertical Space Filling Options	99
14.1 General filler lines commands and controls	99
14.2 Utility commands used with the <code>flexextended</code> option	113
14.3 Filler lines with <code>answerkey</code> option	116
14.4 Filler lines with <code>nosolutions</code> option	118
15 Insertion of figures into a problem or solution	120
15.1 Enclose problem in a minipage	120
15.2 Insertions using a wrapping package	121
16 Insertions in the solution section	122
17 Concerning the <code>fortextbook</code> option	123
17.1 Setting options with <code>\textbookOpts</code>	124
17.2 Macros to display answers/shortsolns	126
17.3 Marginal Matter.	134
17.4 In support of solutions at end of document and chapter	142
17.5 Modifying and restoring the Layout	143
17.6 We shipout in support of <code>fortextbook</code>	145
17.7 Modify <code>eqquestions</code> environment	145
17.8 Modifications for solutions page	146
17.9 Some Convenience/Formatting Commands	148
17.10The <code>probset</code> and <code>example</code> environments	149
17.11Commands in support of Solution Manuals	150
18 <code>fortextbook</code> Style File	152
19 <code>xkeyval</code> Extensions	153
20 Index	165

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, `eqquestions`, 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 123, see also ‘[fortextbook Style File](#)’ on page 152.

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 123.

```
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\eqe@marginbox{\eqe@marginboxright}}
49 \DeclareOptionX{parttotalsonleft}{%
50   \def\eqe@marginbox{\eqe@marginboxleft}}
51 \def\eqe@marginbox{\eqe@marginboxright}
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 `coverpage-sumry`. 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 `solutiononly` 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 `\ifkeyalt` 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\l@stPO{vspacewithsolns}\let\thisOpt@OK\eqe@NO
97   \eqe@option@kenMsg
98   \vspacewithsolnstrue\expandafter\displayworkareatru
99   \else\w@rningBadOpts\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

```

```

116 \ifxetex\ExecuteOptionsX{xetex}\fi\fi}
117 \newif\ifeqonline \eqonlinefalse
118 \DeclareOptionX{online}{\eq@auto@chk@drivers\displayworkareafalse
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
rendition 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      Here is the list of options of exerquiz we plan to recognize.
forcolorpaper
forcolorpaper* 156 \@ifundefined{ifequusecolor}{\newif\ifequusecolor\equsecolorfalse}{
157 \DeclareOptionX{forpaper}{\equsecolorfalse\eqforpapertrue
158   \PassOptionsToPackage{monochrome}{\eq@ColorPackage}}
159 \DeclareOptionX{forcolorpaper}{\equsecolortrue\eqforpapertrue}
160 \newif\ifForceNoColor\ForceNoColorfalse
161 \DeclareOptionX{forcolorpaper*}{\equsecolortrue\eqforpapertrue
preview      162   \ForceNoColortrue} % for print
              Preview shows outlines for form fields.
webOpts      163 \DeclareOptionX{preview}{\previewtrue}
              Pass options to web, when pdf or higher is used.
              164 \define@key{eqexam.sty}{webOpts}[]{\def\eqe@webOpts{#1,\eqDriverName}}
              165 \def\eqe@webOpts{\eqDriverName}

```

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.
              166 \define@key{eqexam.sty}{exerquizOpts}[]{\def\eqe@eqOpts{#1,nodljs}}
              167 \def\eqe@eqOpts{nodljs}
nosolutions   Solutions related options
nohiddensolutions
noHiddensolutions 168 \DeclareOptionX{nosolutions}{\ifx\thisOpt@OK\eqe@YES
solutionsafter 169   \def\l@stPO{nosolutions}\let\thisOpt@OK\eqe@NO
solutiononly 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@solutionsaftertrue\expandafter

```

```

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}{\eq@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.

```

\showproofing \showproofing
\hideproofing \hideproofing
188 \newcommand{\showproofing}{\eq@proofingtrue}
showgrayletters 189 \newcommand{\hideproofing}{\eq@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 \ifdisplayworkareaOff
193 \@ifundefined{ifdisplayworkarea}{\newif\ifdisplayworkarea
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\eq@ColorPackage{xcolor}\def\eqe@color@opt{table}}%
199     \PassOptionsToPackage{xcolor}{table}}
200 {\def\eq@ColorPackage{color}\let\eqe@color@opt\@empty}

```

`noxcolor` forces the use of the color package.

```

201 \DeclareOptionX{noxcolor}{\def\eq@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\if@eqeuseclassmaketitle \@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@globalshowssolutions \eq@globalshowssolutionsfalse
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 \def\previewOn{\previewtrue}\def\previewOff{\previewfalse}
241 \newif\ifeqforpaper \eqforpaperfalse

```

We define the commands for inputting the CFG files.

```

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

```

```

244 \edef{eqemyconfig\eqe@tmp@i}{\noexpand
245 \InputIfFileExists{eqexam\eqe@tmp@i.cfg}{}}}%
246 \eqe@tmp@exp
247 }

```

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.

```

248 \let\bWebCustomize\endinput
249 \let\wWebCustomize\relax
250 \InputIfFileExists{web.cfg}{}{}
251 \InputIfFileExists{eqecus.opt}{}{}

```

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

```

252 \InputIfFileExists{rendition.cfg}{}{}
253 \InputIfFileExists{exambuilder.cfg}{}{}

```

2.7 Process Options

Now process the options.

```

254 \ProcessOptionsX

```

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

```

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

```

```

262 \fi
    The \selectedMC command contains the choice for the styling for the region
    multiple choice questions: rectangles or circles. The default is rectangles.
263 \AtEndOfPackage{\selectedMC}
    (2015/07/12) Move the inclusion of the color package prior to loading many of the
    other required packages.
264 \edef\eqe@tmpexp{\noexpand
265 \RequirePackage{\eq@ColorPackage}}\eqe@tmpexp
    We require a minimal version for xcolor.
266 \ifpackageloaded{xcolor}{\AtEndOfPackage{\let\CT@cell@color\relax
267 \let\CT@arc@relax}%
268 \@ifpackagelater{xcolor}{2004/07/04}{\PackageError{eqexam}{%
269 *****\MessageBreak
270 * Your Version of 'xcolor.sty' is too old!\MessageBreak
271 * You need the version from 2004/07/04 or newer!\MessageBreak
272 * or use: \string\usepackage[noxcolor]{eqexam}\MessageBreak
273 * or \string\documentclass[noxcolor]{article}\MessageBreak
274 *****}\MessageBreak}}{}}
275 }%
276 }{}
    If nocustomdesign option is taken, we set the switch \eqcustomdesignfalse.
277 \if\eqe@nocustomdesign1\eqcustomdesignfalse\fi
    Define a \immediate\write helper macro.
278 \long\def\eqe@IW0#1{\immediate\write#1}

Early definitions for the fortetbook option.

\showAllAnsAtEnd If the user has chosen the vspacewithsolns option, we must turn of all other
solution options, namely answerkey. This command is used internally.
(2016/10/02) Include \eqTopOfSolnPage from exerquiz to support copying ques-
tion to solution page.
279 \newcommand{\showAllAnsAtEnd}{%
280 \makeAnsEnvForSolnsAtEnd\eqTopOfSolnPage
281 \answerkeytrue\eq@proofingtrue
282 \eq@solutionsaftertrue\vspacewithsolnstrue
283 \displayworkareafalse
284 }

\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.
285 \newcommand{\makeAnsEnvForSolnsAtEnd}{%
286 % \proofingsymbol{\ding{52}}%
287 \let\answers\answers@sq

```

```

288 \let\endanswers\endanswers@sq
289 \let\manswers\manswers@sq
290 \let\endmanswers\endmanswers@sq
291 }

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

\writeAllAnsAtEnd writes the \showAllAnsAtEnd command to the solutions file.
294 \def\writeAllAnsAtEnd{\ifsolutionsonly\else
295 \let\quiz@solns\ex@solns
296 \eqe@IWO\quiz@solns{\string\showAllAnsAtEnd}%
297 \ifx\eqeAEFormatting\@empty\else
298 \eqe@IWO\quiz@solns{\string\eqeAEFormatting}\fi
299 \fi
300 \protect\cqqsfalse
301 }

302 \def\writeWithSolDocTrue{\writeT@SolnFile{%
303 \protect\withinsoldoctrue\protect\cqqsfalse}}

```

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

```

304 \newcommand{\setSolnMargins}[1]{\setlength\eqmargin{#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\eqmargin}` 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.

```

305 \def\altSetSolnMargins#1{\def\alt@SetSolnMargins{#1}}
306 \let\alt@SetSolnMargins\@empty
307 \def\writeBeginEqeQuestions{\ifsolutionsonly\else
308 \ifOKToWriteExamData
309 \let\quiz@solns\ex@solns
310 \ifx\alt@SetSolnMargins\@empty
311 \writeT@SolnFile{\string\setSolnMargins{\the\eqmargin}}\else
312 \writeT@SolnFile{\alt@SetSolnMargins}\fi
313 \writeT@SolnFile{\protect\eqgriii\string\noindent
314 \string\begin{eqequestions}}%
315 \fi
316 \fi
317 }

```

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

```

318 \newcommand{\setBtwExamSkip}[1]{\gdef\btwnExamSkipAmt{#1}}

```

```

319 \def\btwnExamSkip{\ifdim#1=0pt\else
320 \vskip#1\relax\fi}}
321 \setBtwnExamSkip{6pt}
322 \def\writeEndEqeQuestions{%
323 \ifsolutiononly\else
324 \ifOKToWriteExamData
325 \let\quiz@solns\ex@solns
326 \eqe@IWO\quiz@solns{\string\eqgrii
327 \string\end{eqequestions}^^J}%
328 \writeT@SolnFile{\string\btwnExamSkip^^J}%
329 \fi\fi
330 }

```

If \ifvspacewithsolns we set the switches need to simulate nosolutions.

```

331 \def\csarg#1#2{\expandafter#1\csname#2\endcsname}
332 \let\w@csarg\csarg
333 \def\saveIFEQE#1{\def\ARG{#1ifSave}%
334 \expandafter\csarg\expandafter
335 \let\expandafter\ARG\csname#1\if#1true\else false\fi\endcsname}
336 \saveIFEQE{vspacewithsolns}\saveIFEQE{answerkey}
337 \saveIFEQE{eq@proofing}\saveIFEQE{eq@solutionsafter}
338 \saveIFEQE{eq@nolink}\saveIFEQE{eq@nosolutions}
339 \saveIFEQE{displayworkarea}
340 \def\vpwsSimulateNoSolns{% dps28
341 \ifvspacewithsolns
342 \answerkeyfalse\eq@proofingfalse\eq@solutionsafterfalse
343 \eq@nolinkfalse\eq@nosolutionsfalse\displayworkareatru
344 \fi}
345 \def\reverseVSWs{\vspacewithsolnsfalse
346 \answerkeyifSave\eq@proofingifSave
347 \eq@solutionsafterifSave\eq@nolinkifSave
348 \eq@nosolutionsifSave\displayworkareainifSave
349 }
350 \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.

```

351 \let\savedeq@online\eq@online
352 \let\savedifeq@solutionsafter\ifeq@solutionsafter
353 \let\savedifeq@hidesolution\ifeq@hidesolution
354 \let\savedifeq@globalshowsolutions\ifeq@globalshowsolutions
355 \let\savedifeq@nosolutions\ifeq@nosolutions
356 \let\savedifeq@proofing\ifeq@proofing
357 \let\savedifeq@nolink\ifeq@nolink
358 \let\savedifpreview\ifpreview
359 \let\savedifeqforpaper\ifeqforpaper
360 \let\ifnosolutions\ifeq@nosolutions

```

3 Required Packages

The following are the required packages for eqexam.

```
361 \RequirePackage{amstext,amssymb}
362 \@ifundefined{if@fleqn}{\let\fleqnOn\relax\let\fleqnOff\relax}
363   {\def\fleqnOn{\@fleqntrue}\def\fleqnOff{\@fleqnfalse}}
```

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

```
364 \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`.

```
365 \@ifundefined{BeforeIncludedComment}{\let\aebc@end\endgroup}
366   {\let\aebc@end\relax}
367 \def\eqe@commentChkMsg{\@ifpackageloaded{comment}
368   {\PackageWarningNoLine{eqexam}
369     {The comment package is incompatible with the\MessageBreak
370     aeb-comment package, do not use the comment package}}{}}
371 \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.

```
372 \AtEndDocument{\includeexersolutions}
```

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

```
373 \@ifpackageloaded{web}{\let\load@web\eqe@YES}{%
374   \ifx\load@web\eqe@YES\ifnum\eqe@drivernum=5
375     \PackageInfo{eqexam}{You have not selected a driver %
376     for eqexam. Perhaps the \MessageBreak
377     driver is introduced through web.cfg}\fi
378     \expandafter\RequirePackage\expandafter[\eqe@webOpts]{web}%
379     \edef\@pdfcreator{\@pdfcreator, The eqexam Package}\fi
380 }
```

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

```
381 \@ifpackageloaded{exerquiz}{\let\load@exerquiz\eqe@YES}{%
382   \let\symbolchoice\@gobble
383   \ifx\load@exerquiz\eqe@YES\expandafter
384     \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.

```
385   \let\importdljs\eqe@YES\let\execjs\eqe@YES
386   \fi
387 }
388 \@ifpackageloaded{eforms}{}{%
```


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

```
389 \let\symbolchoice\@gobble
390 }
```

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.

```
391 \def\eqDvipsone{dvipsone}
392 \@ifpackageloaded{hyperref}
393   {\ifx\eqDriverName\eqDvipsone
394     \renewcommand\@pdfviewparams{ null null null}\fi
395   }{\let\textorpdfstring\@firstoftwo}
```

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

```
396 \let\eq@online\savedeq@online
397 \let\ifeq@solutionsafter\savedifeq@solutionsafter
398 \def\ifsolutionsafter{\ifeq@solutionsafter} % user interface
399 \let\ifeq@hidesolution\savedifeq@hidesolution
400 \let\ifeq@globalshowsolutions\savedifeq@globalshowsolutions
401 \let\ifeq@nosolutions\savedifeq@nosolutions
402 \let\ifeq@proofing\savedifeq@proofing
403 \let\ifeq@nolink\savedifeq@nolink
404 \let\ifpreview\savedifpreview
405 \let\ifeqforpaper\savedifeqforpaper
```

Other packages of interest.

```
406 \RequirePackage{calc}
407 \RequirePackage{pifont}
408 \RequirePackage{array}
```

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

```
409 \RequirePackage{verbatim}
410 \@ifundefined{dlcomment}{%\typeout{!! defining dlcomment}}%
411 \let\dlcomment\comment
412 \let\enddlcomment\endcomment
413 \let\eqSavedComment\dlcomment
414 \let\endeqSavedComment\enddlcomment
415 }{}
```

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

```
416 \RequirePackage{multicol}
417 \setlength\columnseprule{.4pt}
418 \raggedcolumns\multicolsep=3pt
419 \newcommand{\setmulticolprob}{%
420   \setlength{\linewidth}{\linewidth+\eqmargin}}
```

For the `fortextbook` option, we require `eso-pic`.

```
421 \edef\eqe@reqPack{\ifeqfortextbook\noexpand\RequirePackage{eso-pic}\else
422 \relax\fi}
423 \eqe@reqPack
```

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

```
424 \ifspacewithsolns\solutionsAtEndtrue\fi
425 \ifanswerkey\solutionsAtEndfalse\fi
426 \ifeq@nosolutions\solutionsAtEndfalse\fi
427 \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.

```
428 \@ifundefined{eqemargin}{\newlength{eqemargin}}{}
429 \providecommand{\prbDecPt}[1]{\def\eqe@decPointPrb{#1}}
430 \def\eqe@decPointPrb{.}\def\eqe@dpsepPrb{\ }
431 \providecommand{\prbPrtsep}[1]{\def\eqe@prtsepPrb{#1}}
432 \def\eqe@prtsepPrb{\ }
433 \providecommand{\prbNumPrtsep}[1]{\def\eqe@hspannerPrb{#1}}
434 \def\eqe@hspannerPrb{\ }
435 \newcommand{\eqexammargin}[2][\normalsize\normalfont\bfseries]{%
436   \settowidth{eqemargin}{#1#2\eqe@decPointPrb\eqe@hspannerPrb}}
```

(2011/05/08) The default margin for the `eqexam` environment, two digits and a space.

```
437 \eqexammargin{00}

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

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

```
440 \newcommand{\eqeSetExamPageParams}{%
441   \setlength{\headheight}{12pt}
442   \setlength{\topmargin}{-.5in}
443   \setlength{\headsep}{20pt}
444   \setlength{\oddsidemargin}{0pt}
445   \setlength{\evensidemargin}{0pt}
446   \setlength{\marginparsep}{11pt}
447   \setlength{\marginparwidth}{35pt}
448   \setlength{\footskip}{11pt}
449 }
```

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

```
450 \newcommand{\eqExamPageLayout}{%
451   \setlength\textwidth\paperwidth
452   \addtolength{\textwidth}{-2in}
453   \addtolength{\textwidth}{-\oddsidemargin}}
```

```

454 \setlength\textheight{\paperheight}
455 \addtolength\textheight{-2in}
456 \addtolength\textheight{-\headheight}
457 \addtolength\textheight{-\headsep}
458 \addtolength\textheight{-\topmargin}
459 \addtolength\textheight{-\footskip}
460 }

```

(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

```

461 \ifeqcustomdesign\else
462 \eqeSetExamPageParams
463 \eqExamPageLayout
464 \fi

```

A simple page layout scheme for this exam.

```

465 \newcommand{\ps@eqExamheadings}
466 {%
467   \renewcommand{\@oddhead}{%
468     {\normalfont\normalsize\ifnum\value{page}<2
469       \hfil\else\eqExamRunHead\fi}}%
470   \renewcommand{\@evenhead}{\@oddhead}
471   \renewcommand{\@oddfoot}{\settotalsbox\runExamFooter}
472   \renewcommand{\@evenfoot}{\@oddfoot}
473 }
474 \raggedbottom

```

5 Counters, Lengths and Tokens

eqpointsofar 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.

```

475 \newcount\eqe@tempcnta
476 \newcounter{eqpointsofar}
477 \newcounter{eqpointsthispage}
478 \newcounter{eq@numparts}
479 \newcounter{eq@count}
480 \newtoks\partNames \partNames={ }
481 \newlength{\eq@tmplengthA}
482 \newlength{\eq@tmplengthB}
483 \newbox{\eq@pointbox}
484 \newlength{\eq@pointboxtotalheight}

```

Some scratch registers to do calc calculations.

```

485 \newlength{\eqetmplengtha}
486 \newlength{\eqetmplengthb}

```

6 Some Macros to Support the Options

We make a few definitions to support various options.

```

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

514 \def\NoSolutions{\eq@nolinktrue\eq@nosolutionstrue
515   \displayworkareatrue}
516 \@onlypreamble\NoSolutions

```

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

```

517 \def\chngToNoSolns{\solutionsAtEndfalse\answerkeyfalse
518   \eq@proofingfalse\eq@solutionsafterfalse
519   \eq@nosolutionstrue\displayworkareatrue}

```

`\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`.

```

520 \def\vspacewithkeyOn{\keepdeclaredvspacingtrue}
521 \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.

```
522 \newif\ifdispl@yPoints\displ@yPointstrue
523 \newcommand{\displayPointsOn}{\displ@yPointstrue}
524 \newcommand{\displayPointsOff}{\displ@yPointsfalse}
525 \newif\ifl@stDispl@yPoints\l@stDispl@yPointsfalse
```

`\encloseProblemsWith` `\encloseProblemsWith` to support the `solutionsonly` option

```
526 \def\encloseProblemsWith#1{%
527   \ifsolutionsonly\excludecomment{#1}\else
528     \includecomment{#1}\fi
529 }
```

7 Colors

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

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

```

562 \instructionsColor{black}\eqCommentsColor{black}%
563 \eqCommentsColorBody{black}%
564 \eqEndExamTotalColor{black}\ckboxColor{}%
565 \ckcirColor{}\if\load@web\eqe@YES
566 \sectionColor{black}\fi
567 }
568 \ifForceNoColor
569 \AtBeginDocument{\forceNoColorSet}
570 \fi
571 \newcommand{\forceNoColor}{\ifForceNoColor\else\ForceNoColortrue
572 \AtBeginDocument{\forceNoColorSet}\fi}
\ckboxColor The \ckboxColor 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.
573 \providecommand{\ckboxColor}[1]{\def\@rgi{#1}\ifx\@rgi\@empty
574 \let\ckbox@Color\relax\else
575 \def\ckbox@Color{\color{#1}}\fi}\ckboxColor{}
576 \providecommand{\ckcirColor}[1]{\def\eq@rgi{#1}\ifx\eq@rgi\@empty
577 \let\ckcir@Color\relax\else
578 \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`.

```

579 \def\examNum#1{\def\nExam{#1}}
580 \examNum{1}

```

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

```

\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`.

```

581 \def\Exam{\ifAB{\eq@VersionAtext}{\eq@VersionBtext}}
582 \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 583 \def\VersionAtext#1{\def\eq@VersionAtext{#1}}
\shortVersionBtext 584 \def\VersionBtext#1{\def\eq@VersionBtext{#1}}
                    585 \def\shortVersionAtext#1{\def\eq@shortVersionAtext{#1}}
                    586 \def\shortVersionBtext#1{\def\eq@shortVersionBtext{#1}}
                    587 \VersionAtext{Exam~\nExam--Version A}
                    588 \VersionBtext{Exam~\nExam--Version B}
                    589 \shortVersionAtext{Exam~\nExam A}
                    590 \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.

```

591 \newtoks\eqtemptokena
592 \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
593 \def\numVersions#1{\ifnum#1>26\def\eq@nVersions{26}%
594   \PackageWarning{eqexam}{The value of \string\numVersions\space is
595    too large. \MessageBreak Choose a natural number less than 27}
596   \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 L^AT_EX 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.

```

597 \newcommand{\longTitleText}[1] [] {%

```

```

598 \ifeqglobalversion\let\eq@selectedVersion@save\eq@selectedVersion
599 \else\let\eq@selectedVersion@save\relax\fi
600 \uppercase{\def\eqe@localTextTitle{#1}}%
601 \ifx\eqe@localTextTitle\empty\else
602 \expandafter\forVersion\expandafter{\eqe@localTextTitle}\fi
603 \eqe@contTitleText{\Exam}{\endlongTitleText}%
604 }
605 \def\endlongTitleText{1}
606 \newcommand{\shortTitleText}[1] [] {%
607 \ifeqglobalversion\let\eq@selectedVersion@save\eq@selectedVersion
608 \else\let\eq@selectedVersion@save\relax\fi
609 \uppercase{\def\eqe@localTextTitle{#1}}%
610 \ifx\eqe@localTextTitle\empty\else
611 \expandafter\forVersion\expandafter{\eqe@localTextTitle}\fi
612 \eqe@contTitleText{\sExam}{\endshortTitleText}%
613 }
614 \def\endshortTitleText{s}

```

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

```

615 \def\eqe@contTitleText#1#2{%
616 \setcounter{eq@count}{0}%
617 \eqtemptokena={}\let\endtitleMarker#2
618 \@gatherTitleText{#1}%
619 }

```

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`.

```

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

```



```

637             \if\endtitleMarker\endlongTitleText
638                 \expandafter\eqe@absorbTokensLong
639             \else
640                 \expandafter\eqe@absorbTokensShort
641             \fi
642         }%
643     \fi
644 \fi
645 \eqe@next
646 }
647 \long\def\eqe@absorbTokensLong#1\endlongTitleText{%
648     \protected\xdef\Exam{\Exam}\ifx\eq@selectedVersion@save\relax
649     \eqe@offVersion\else\expandafter\forVersion\expandafter
650     {\eq@selectedVersion@save}\fi}
651 \long\def\eqe@absorbTokensShort#1\endshortTitleText{%
652     \protected\xdef\sExam{\sExam}\ifx\eq@selectedVersion@save\relax
653     \eqe@offVersion\else\expandafter\forVersion\expandafter
654     {\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.

```

655 \def\eq@shortTitlesFix#1{%
656     \PackageWarning{eqexam}{You have defined an insufficient number
657     of titles\MessageBreak for the number of versions declared in
658     \string\numVersions.\MessageBreak Please fix the problem}%
659     \stepcounter{eq@count}%
660     \if\endtitleMarker\endlongTitleText
661         \edef\eqe@tmp{\noexpand\@gatherTitleText{\noexpand#1}
662             {??---Title \# \the\value{eq@count}---??}%
663             \noexpand\endlongTitleText}
664     \else
665         \edef\eqe@tmp{\noexpand\@gatherTitleText{\noexpand#1}
666             {T\#\the\value{eq@count}??}\noexpand\endshortTitleText}
667     \fi
668     \addtocounter{eq@count}{-1}%
669     \eqe@tmp
670 }

```

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

```

671 \def\ifAB#1#2{\if\eq@selectedVersion A#1%
672     \else\if\eq@selectedVersion B#2\fi\fi}
673 \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 $\vA{\text{place}}\vB{\text{date}}\vC{\text{year}}$ 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  $\vA{x}\vB{y}\vC{z}$ :
\[
\begin{verA}
2x + 4 = 7
\end{verA}
\begin{verB}
5y + 2 = 4
\end{verB}
\begin{verC}
3z - 2 = 2
\end{verC}
\]

674 \newif\ifeqglobalversion \eqglobalversionfalse
675 \newif\ifeqlocalversion \eqlocalversionfalse
676 \newif\if@templocalversion \@templocalversionfalse
677 \def\eqe@initializeMultiVersions{%
678   \let\save@message\message\let\message@gobble
679   \@tfor\eqe@tmp:=ABCDEFGHIJKLMNQRSTUWXYZ\do{%
680     \csarg\let{v\eqe@tmp}\@gobble
681     \edef\exp@temp{\noexpand\excludcomment{ver\eqe@tmp}}\exp@temp
682     \csarg\let{Afterver\eqe@tmp Comment}\aebc@end
683   }\let\message\save@message
684 }
685 \AtBeginDocument{\let\eqe@initializeMultiVersions\relax}

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

686 \let\eqe@@onVersion@empty
687 \def\eqe@@onVersion{g@addto@macro\eqe@@onVersion}
688 \let\eqe@@offVersion@empty
689 \def\eqe@@offVersion{g@addto@macro\eqe@@offVersion}
690 \let\eqe@@holdTemp@empty
691 \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.

```

692 \def\eqe@showArg#1{#1}
693 \def\eqe@turnOnComment#1{%
694 %   \csarg\let{v#1}\@empty
695   \csarg\let{v#1}\eqe@showArg
696   \edef\exp@temp{\noexpand\includecomment{ver#1}}\exp@temp
697 }
698 \def\eqe@turnOffComment#1{%
699   \csarg\let{v#1}\@gobble
700   \edef\exp@temp{\noexpand\excludecomment{ver#1}}\exp@temp
701   \csarg\let{Afterver#1Comment}\aebc@end
702 }

```

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

```

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

```

```

735     {%
736         \if#1\alph{eq@count}%
737         \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@offVeresion.
738         \edef\temp@exp{\noexpand
739             \eqe@turnOnComment{\Alph{eq@count}}}%
740         \expandafter\eqe@onVersion\expandafter{\temp@exp}%
741         \edef\temp@exp{\noexpand
742             \eqe@turnOffComment{\Alph{eq@count}}}%
743         \expandafter\eqe@offVersion\expandafter{\temp@exp}%
744     \else
745         \eqe@turnOffComment{\Alph{eq@count}}%
746     \fi
747     }%
748     \ifnum\value{eq@count}<\eq@nVersions\repeat
749     \let\message\save@message
750 }

```

Let us assume version A initially, user with reset this in document.

```

751 \AtEndOfPackage{\numVersions{26}\forVersion{A}%
752     \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.

```

753 \def\selectVersion#1#2{% #1 \le #2
754     \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`.

```

755     \let\nneedsModArith\eqe@YES
756     \ifeqobeylocalversion\ifx\nLocalSelection\@empty\else
757     \let\nneedsModArith\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.

```

758     \edef\exer@solnheadhook{%
759         \string\selectVersion{#1}{#2}}%

```

Turn off messaging.

```

760     \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.

```
761   \eqe@@onVersion
762   \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`.

```
763       \eqe@@onVersion\eqe@@holdTemp
764       \let\eqe@@holdTemp\@empty
765   \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.

```
766 \ifx\needsModArith\eqe@YES
767   \ifeqglobalversion\ifnum\eq@nSelectedVersion>\nLocalVersions
```

If we perform modular arithmetic, turn off original choice.

```
768       \eqe@@offVersion
```

Now perform mod arithmetic

```
769       {\count0=\eq@nSelectedVersion \count2=\count0
770       \advance\count0by-1 \divide\count0by\nLocalVersions
771       \multiply\count0by\nLocalVersions %\count2=\eq@nSelectedVersion
772       \advance\count2by-\count0
773       \xdef\nLocalSelection{\the\count2 }%
774       \ifeqobeylocalversion\else
775         \global\@templocalversiontrue
776         \global\eqobeylocalversiontrue\fi}%
777       \global\let\eqe@@holdTemp\@empty
778   \fi\fi
779 \fi
780   \ifeqobeylocalversion
781     \global\eqlocalversiontrue
782     \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

```
783       \ifx\nLocalSelection\@empty
784         \def\nLocalSelection{\eq@nSelectedVersion}%
785       \fi
```

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

```
786       \ifnum\nLocalSelection>\nLocalVersions
787         {\count0=\nLocalSelection
788         \count2=\count0
789         \advance\count0by-1 \divide\count0by\nLocalVersions
```

```

790         \multiply\count0by\nLocalVersions
791         \advance\count2by-\count0
792         \xdef\nLocalSelection{\the\count2 }}%
793     \fi
794     \let\save@message\message\let\message@gobble
795     \loop
796         \stepcounter{eq@count}%
797         \csarg\let{Afterver\Alph{eq@count}Comment}\aebc@end
798         \lowercase
799         {%
800             \ifnum\value{eq@count}=\nLocalSelection
801                 \eqe@turnOnComment{\Alph{eq@count}}}%
802                 \edef\temp@exp{\noexpand
803                     \eqe@turnOffComment{\Alph{eq@count}}}%
804                     \expandafter\eqe@holdTemp\expandafter{\temp@exp}%
805             \else
806                 \eqe@turnOffComment{\Alph{eq@count}}}%
807             \fi
808         }%
809         \ifnum\value{eq@count}<\nLocalVersions\repeat
810         \let\message\save@message
811     \fi
812     \if@templocalversion\global\eqobeylocalversionfalse\fi
    added 09/10/03 reset back to default
813     \@templocalversionfalse
814     \let\message\save@message
815 }

```

9 Title Definitions from Web

```

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

```

```

832     \@ifundefined{hypersetup}{}{\hypersetup{pdfsubject={#2}}}}
833 \def\email#1{\gdef\webemail{#1}}
834 \def\keywords#1{\gdef\webkeywords{#1}%
835     \@ifundefined{hypersetup}{}{\hypersetup{pdfkeywords={#1}}}}
836 \def\university#1{\gdef\webuniversity{#1}}
837 \def\copyrightyears#1{\gdef\webcopyrightyears{#1}}
838 \def\version#1{\gdef\web@version@value{#1}%
839     \edef\webversion{\ifx\web@version@value\empty\else
840         \noexpand\web@versionlabel\noexpand\
841         \noexpand\web@version@value\fi}%
842 }
843 \let\web@version@value\empty
844 \def\versionLabel#1{\def\web@versionlabel{#1}}
845 \versionLabel{Version}
846 \def\web@toc{Table of Contents}
847 \def\web@continued{cont.}
848 % set some defaults
849 \title{} \author{} \email{} \subject{} \keywords{} \university{}
850 \providecommand{\optionalPageMatter}[2] [] {%
851     \def\optionalpagematter{#2}}
852 \def\optionalpagematter{}
853 }

```

`\date` L^AT_EX (T_EX) defines a `\date` command that is also used by eqexam.

```

854 \def\duedate#1{\def\theduedate{#1}}
855 \duedate{}

```

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

```

856 \def\duedate#1{\def\theduedate{#1}}
857 \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.

```

858 \newcommand\thisterm{%
859 % if prior to June (Jan--May)
860     \ifnum\month<6Spring\else
861 % if prior to August (June-July)
862     \ifnum\month<8Summer\else
863 % August or later; if Sept--Dec
864     \ifnum\month>8Fall\else
865 % Month of August; if after 25th
866     \ifnum\day>25Fall\else
867 % if it's 25th or earlier in month of August
868     Summer\fi\fi\fi\fi
869 }

```

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.

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

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

```
886 \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.

```
887 \newcommand\examSIDLabel[1]{\gdef\@examSIDLabel{#1}}
888 \examSIDLabel{SID:}
889 \newcommand\eqSID[2] []{\def\eqSID@argi{#1}\def\eqSID@argii{#2}}
890 \def\eq@SID{\bgroup\settowidth\eq@tmplengthA{\@examSIDLabel\ }%
891   \@tempdima=\eqSID@argii\relax\advance\@tempdima by-\eq@tmplengthA
892   \eqe@idinfohl{\makebox[\eqSID@argii][l]{\@examSIDLabel}}%
893   \expandafter\insTxtFieldIdInfo\expandafter[\eqSID@argi]%
894   {\@tempdima}{IdInfo.SID}\egroup}
```

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

```
895 \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.

```
896 \newcommand\examEmailLabel[1]{\gdef\@examEmailLabel{#1}}
897 \examEmailLabel{Email:}
898 \newcommand\eqEmail[2][]{%
899     \def\eqExam@argi{#1}\def\eqEmail@argii{#2}}
900 \gdef\eq@Email{\bgroup\settowidth\eq@tmplengthA{\@examEmailLabel\ }%
901     \@tempdima=\eqEmail@argii\relax\advance\@tempdima by-\eq@tmplengthA
902     \underbar{\makebox[\eqEmail@argii][l]{\@examEmailLabel}}%
903     \expandafter\insTxtFieldIdInfo\expandafter[\eqEmail@argi]%
904     {\@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.)

```
905 \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.

```
906 \def\insTxtFieldIdInfo[#1]#2#3{%
907     \ifundefined{@quiz}{\if\eq@online\eqe@YES
908         \ifeq@nosolutions\ifeq@solutionsafter\else
909             \raisebox{-1bp}{\makebox[Opt][r]{%
910                 \textField[\BC{#1}{#3}{#2}{11bp}}}%
911             \fi\fi\fi
912     }%
913 }
```

`\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.)

```
914 \def\SubmitInfo#1#2{%
915     \def\EqExam@SubmitURL{#1}\def\@EmailInstr{#2}%
916 }
```

`\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.

```
917 \def\EmailCourseName#1{\def\@EmailCourseName{#1}}
```

Here's the default value.

918 `\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.)

919 `\def\EmailExamName#1{\def\@EmailExamName{#1}}`
 Here's the default value.

920 `\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.

921 `\def\EmailSubject#1{\def\@EmailSubject{#1}}`
 Here's the default value, which generates no custom subject line.

922 `\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 example.

`\ServerRetnMsg` Unless submitted in `silent` mode, the `eqAttach.asp` returns a message acknowledging the receipt of the data. `\ServerRetnMsg` is used to customize this message.

923 `\def\ServerRetnMsg#1{\def\@ServerRetnMsg{#1}}`
 Here's the default value, which generates no custom return message

924 `\ServerRetnMsg{}`

`\SubmitButtonLabel` is the label that appears on the submit button.

925 `\def\SubmitButtonLabel#1{\def\@SubmitButtonLabel{#1}}`
 Here's the default value.

926 `\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.

927 `\let\priorSubmitJS@gobble`
 928 `\let\postSubmitJS@empty`
 929 `\def\SubmitButton`
 930 `{%`
 931 `\ifx\use@email\eqe@YES\ifeq@nosolutions\ifeq@solutionsafter\else`
 932 `\makebox[Opt][l]{\pushButton`
 933 `[\CA{\@SubmitButtonLabel}\A{\JS{%`
 934 `var _eqEok2Submit = true;\r`
 935 `var aSubmitFields = new Array("eqexam", "IdInfo");\r`
 936 `\priorSubmitJS\r`
 937 `if(_eqEok2Submit) this.submitForm("\EqExam@SubmitURL",`

```

938         true, false, aSubmitFields);r
939         \postSubmitJS
940     }]{Submit}{1.5in}{16bp}}%
941     \makebox[Opt][l]{\textField[\F\FHidden\DV{\@EmailInstr}
942         \V{\@EmailInstr}]{IdInfo.mailTo}{11bp}{11bp}}%
943     \makebox[Opt][l]{\textField[\F\FHidden\DV{\@EmailCourseName}
944         \V{\@EmailCourseName}]{IdInfo.courseName}{11bp}{11bp}}%
945     \makebox[Opt][l]{\textField[\F\FHidden\DV{\@EmailExamName}
946         \V{\@EmailExamName}]{IdInfo.examName}{11bp}{11bp}}%
947     \makebox[Opt][l]{\textField[\F\FHidden\DV{\@EmailSubject}
948         \V{\@EmailSubject}]{IdInfo.subject}{11bp}{11bp}}%
949     \makebox[Opt][l]{\textField[\F\FHidden\DV{\@ServerRetnMsg}
950         \V{\@ServerRetnMsg}]{IdInfo.retnmsg}{11bp}{11bp}}%
951     \fi\fi\fi
952 }
953 \def\thequizno{\if\probstar*\Alph{quizno}\else\alph{quizno}\fi}
954 \def\linkContentFormat{%
955     \if\probstar*\Alph{quizno}\else\alph{quizno}\fi
956 \def\linkContentWrapper{(\hfil\linkContentFormat\hfil)}%
957 \def\Ans@r@l@Defaults
958 {%
959     \BC{} \S{S} \W{1} \Ff{\FfNoToggleToOff}
960     \textSize{12}\textColor{0 g}
961 }

```

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

```

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

```

```

985     \edef\@linkcolor{\@nolinkcolor}%
986     \fi
987     \ifeq@nosolutions\edef\@linkcolor{\@nolinkcolor}\fi
988     \textcolor{\@linkcolor}{\makebox[\eq@tmpdima]{\link@@Content}}%
989     \Ans@proofing{\eq@tmpdima}%
990     \eq@hspanner\ignorespaces
991 }
(2019/10/28) Legacy assignment, in case eqexam.def/exerquiz are still using
\eq@RadioCheck@driver.
992 \@ifundefined{eq@RadioCheck@driver}{%
993   {\let\eq@Radio@driver\eq@RadioCheck@driver}
994 \def\eqExam@Ans@sq@f{\if\eq@listType1\stepcounter{quizno}\else
995   \ifwithinsoldoc\stepcounter{quizno}\else
996     \refstepcounter{quizno}\fi\fi
997   \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@f}%
998   \eq@tmpdima=\wd\eq@tmpbox%
999   \hangindent=\eq@tmplength\hangafter=1\relax
1000  \if\eq@online\eqe@N0\previewtrue
1001    \insertGrayLetters
1002    \Ans@sq@f@driver
1003  \else
1004    \ifanswerkey\previewtrue
1005      \Ans@sq@f@driver
1006    \else
1007      \edef\fieldName{%
1008        \if\probstar*eqexam.\curr@quiz.\theequestionnoi.%
1009        part\thepartno%
1010      \else
1011        eqexam.\curr@quiz.\theequestionnoi%
1012      \fi
1013      }\insertGrayLetters
1014      \radio@@Button{\presets{\eqe@optsRadiof}}%
1015      {\fieldName}{\RadioFieldSize}%
1016      {\RadioFieldSize}{\Ans@choice\alph{quizno}}{\eq@protect\A}%
1017      {\eq@setWidgetProps\eq@Radio@driver}%
1018      {\@@Ans@sq@f@Defaults\Ans@sq@f@Actions\every@RadioButton
1019      \every@sqRadioButton\insert@circlesymbol}%
1020    \fi
1021  \fi
1022  \Ans@proofing{\RadioFieldSize}%
1023  \eq@hspanner\ignorespaces
1024 }
1025 \newcommand{\optsCkBxf}[1]{\def\eqe@optscbf{#1}}
1026 \newcommand{\optsCkBl}[1]{\def\eqe@optschl{#1}}
1027 \let\eqe@optschl@empty\let\eqe@optscbf@empty
1028 \def\eqExam@Ans@ck@sq@l{\leavevmode
1029   \if\eq@listType1\stepcounter{quizno}\else
1030   \ifwithinsoldoc\stepcounter{quizno}\else
1031     \refstepcounter{quizno}\fi\fi

```

```

1032 \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@l}%
1033 \sbox{\eq@tmpbox}{\eq@l@l}\eq@tmpdima=\wd\eq@tmpbox
1034 \def\link@@Content{\linkContentWrapper}%
1035 \hangindent=\eq@tmplength\hangafter=1\relax
1036 \edef\fieldName{%
1037     \if\probstar*eqexam.\curr@quiz.\thequestionnoi.%
1038     part\thepartno.\alph{quizno}%
1039     \else
1040     eqexam.\curr@quiz.\thequestionnoi.\alph{quizno}%
1041     \fi
1042 }%
1043 \if\eq@online\eqe@YES\relax
1044 \makebox[Opt][l]{\check@@Box{\presets{\eqe@optscbl}}{\fieldName}%
1045     {\eq@tmpdima}{\RadioFieldSize}{\Ans@choice\alph{quizno}}%
1046     {\eq@protect\A}{\eq@setWidgetProps\eq@l@check@driver}%
1047     {\Ans@r@l@Defaults\every@RadioButton\every@qRadioButton}}%
1048 \else
1049     \edef\@linkcolor{\@nolinkcolor}%
1050 \fi
1051 \ifeq@nosolutions\def\@linkcolor{\@nolinkcolor}\fi
1052 \textcolor{\@linkcolor}{\makebox[\eq@tmpdima]{\link@@Content}}%
1053 \Ans@proofing{\eq@tmpdima}%
1054 \eq@hspanner\ignorespaces
1055 }
1056 \def\eqExam@Ans@ck@sq@f{\if\eq@listType1\stepcounter{quizno}\else
1057     \ifwithinsoldoc\stepcounter{quizno}\else
1058     \refstepcounter{quizno}\fi\fi
1059 \PBS\raggedright\settowidth{\eq@tmplength}{\eq@lw@f}%
1060 \eq@tmpdima=\wd\eq@tmpbox%
1061 \hangindent=\eq@tmplength\hangafter=1\relax
1062 \if\eq@online\eqe@NO\previewtrue
1063     \insertGrayLetters
1064     \Ans@sq@f@driver
1065 \else
1066     \ifanswerkey\previewtrue
1067     \Ans@sq@f@driver
1068 \else
1069     \edef\fieldName{%
1070         \if\probstar*eqexam.\curr@quiz.\thequestionnoi.%
1071         part\thepartno.\alph{quizno}%
1072         \else
1073         eqexam.\curr@quiz.\thequestionnoi.\alph{quizno}%
1074         \fi
1075     }\insertGrayLetters
1076     \mbox{\check@@Box{\presets{\eqe@optscbf}}{\fieldName}%
1077         {\RadioFieldSize}{\RadioFieldSize}%
1078         {\Ans@choice\alph{quizno}}{\eq@protect\A}%
1079         {\eq@setWidgetProps\eq@Radio@driver}%
1080         {\@@Ans@sq@f@Defaults\Ans@sq@f@Actions\every@RadioButton
1081         \every@sqRadioButton}}%

```

```

1082     \fi
1083     \fi
1084     \Ans@proofing{\RadioFieldSize}%
1085     \eq@hspanner\ignorespaces
1086 }

```

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

```

1087 \def\optsMlTxFld#1{\def\eqe@optsmultf{#1}}
1088 \let\eqe@optsmultf\@empty

```

`\wdthMlTxFld{<length>}` sets the underlying multi-line text field, when `online` option is in effect, to `<length>`. The default length of `\linewidth` is reset after the field has been created.

```

1089 \def\eqeLW{\linewidth}
1090 \def\wdthMlTxFld#1{\setlength\eqetmplengtha{#1}%
1091   \edef\eqeLW{\the\eqetmplengtha}}
1092 \def\resetFldWdth{\gdef\eqeLW{\linewidth}}%
1093 \def\eqExamPriorVspace#1{%
1094   \edef\fieldName{%
1095     \if\probstar*eqexam.\curr@quiz.%
1096       \thequestionnoi.part\thepartno.solution%
1097     \else
1098       eqexam.\curr@quiz.\thequestionnoi.solution%
1099     \fi
1100   }%
1101   \nobreak\noindent\textField[BC]{\presets{\eqe@optsmultf}
1102     \Ff\FfMultiline}{\fieldName}\eqeLW{#1}\resetFldWdth\@gobble
1103 }

```

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.

```

1104 \@ifpackageloaded{exerquiz}{%
1105   \let\Ans@sq@l\eqExam@Ans@sq@l
1106   \let\Ans@sq@f\eqExam@Ans@sq@f
1107   \let\Ans@ck@sq@l\eqExam@Ans@ck@sq@l
1108   \let\Ans@ck@sq@f\eqExam@Ans@ck@sq@f
1109   \def\eqexheader@wrapper{\makebox[0pt][r]{%
1110     \hypertarget{qex.\the@exno}{\eqexheader}}}%
1111   \if\eq@online\eqe@YES\relax
1112     \newcounter{@cntfillin}%
1113     \let\eqPriorVspace\eqExamPriorVspace
1114   \fi
1115 }%
1116 {%
1117   \input{eqalone.def}
1118   \input{eqexam.def}
1119   \@ifl@ter{def}{eqexam}{\eqexamdefReq}{%
1120     {\PackageWarningNoLine{eqexam}
1121       {This version of eqexam requires eqexam.def\MessageBreak

```

```

1122   dated \eqexamdefReq\space or later}}
1123   \def\eqexheader@wrapper{\makebox[Opt][r]{\eqexheader}}
1124 }

```

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

```

1125 %\writeBeginEqeQuestions

```

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

```

1126 \AtBeginDocument{%
1127   \ifvspacewithsolns\writeAllAnsAtEnd\else
1128   \ifeqfortextbook\writeAllAnsAtEnd\else
1129   \writeWithSolDocTrue\fi\fi
1130 }

```

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

```

1131 \vspacewithkeyOff
1132 </package>

```

11 Stand alone Code

```

1133 <*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.

```

1134 \ProvidesFile{eqalone.def}
1135 [2012/25/01 v3.0t Minimal code used by eqexam (dps)]
1136 \ifundefined{eq@tmpbox}{\newsavebox{eq@tmpbox}}{}% defined in eforms
1137 \ifundefined{eq@tmpdima}{\newdimen\eq@tmpdima}{} % defined in eforms
1138 \def\RadioFieldSize{11bp}
1139 \newdimen\eqcenterWidget

```

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

```

1140 \def\centerWidget
1141 #1{%
1142   \eqcenterWidget=#1
1143   \eqcenterWidget=.5\eqcenterWidget
1144   \advance\eqcenterWidget by-4bp
1145 }

```

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

```

1146 \def\eqe@BboxRect#1#2{\hbox{\ckbox@Color\vbox{\hrule width #1
1147   \hbox to#1{\vrule height#2\hfill\vrule height#2}\vfill\hrule}}}
1148 \def\useRectForMC{\let\Bbox\eqe@BboxRect}
1149 \def\selectedMC{\useRectForMC}

```

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

```
1150 \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.

```
1151 \def\eqe@striphbox\mbox#1{#1}
1152 \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.

```
1153 \newcommand{\proofingsymbol}[1]{%
1154   \def\@proofingsymbol{\textcolor{\@proofingsymbolColor}{#1}}
   This is the answers for the link-style and is called from the eqexam.def file.
1155 </standalone>
1156 <*package>
```

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

```
1157 \def\selectedMC{\useRectForMC}
1158 \ifallowcircmc
1159   \font\eqe@lcir=lcircle10 at 12pt
1160   \bgroup
```

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

```
1161   \setbox0=\hbox{\eqe@lcir h}
1162   \xdef\eqe@cirDiam{\the\wd0}
1163   \@tempdima=.5\wd0
1164   \xdef\eqe@cirRadius{\the\@tempdima}
1165 \egroup
```

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

```
1166   \def\circ@Glyph#1#2{\hbox{\smash{\raisebox{\eqe@cirRadius}%
1167     {\makebox[\eqe@cirDiam]{\llap{\ckcir@Color
1168       \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.

```
1169   \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`.

```
1170   \def\circProofingForCirc{\centerWidget\RadioFieldSize
1171     \lower\eqcenterWidget\circ@Glyph{x}{1.4pt}}
1172   \if\eq@online\eqe@YES
```


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

```

1173     \def\useCircForMC{\let\Bbox\eqe@BboxCirc\useMCCircles}
1174     \def\selectedMC{\useCircForMC}
1175     \@ifundefined{eqe@BboxRect}
1176         {\def\useRectForMC{\let\Bbox\ef@Bbox\useMCRects}}
1177         {\def\useRectForMC{\let\Bbox\eqe@BboxRect\useMCRects}}
1178 %     \let\useRectForMC\relax
1179 \else % if not online
1180     \@ifundefined{eqe@BboxRect}
1181         {\def\useRectForMC{\let\Bbox\ef@Bbox}}
1182         {\def\useRectForMC{\let\Bbox\eqe@BboxRect}}
1183     \def\useCircForMC{\let\Bbox\eqe@BboxCirc}
1184     \def\selectedMC{\useCircForMC}
1185 \fi
1186 \def\useCircForProof{\symbolchoice{circle}%
1187     \proofingsymbol{\circProofingForCirc}}
1188 \else
    If allowcircmc is not taken
1189     \if\eq@online\eqe@YES
1190         \def\useCircForMC{\let\Bbox\eqe@BboxCirc\useMCCircles}
1191         \@ifundefined{eqe@BboxRect}
1192             {\def\useRectForMC{\let\Bbox\ef@Bbox\useMCRects}}
1193             {\def\useRectForMC{\let\Bbox\eqe@BboxRect\useMCRects}}
1194     \else
1195         \@ifundefined{eqe@BboxRect}
1196             {\def\useRectForMC{\let\Bbox\ef@Bbox}}
1197             {\def\useRectForMC{\let\Bbox\eqe@BboxRect}}
1198         \def\useCircForMC{\useRectForMC}
1199         \def\selectedMC{\useRectForMC}
1200     \fi
1201     \let\useCircForProof\relax
1202 \fi

```

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

```

1203 \newif\ifuserectforms
1204 \def\useRectForMS{\userectformstrue}
1205 \def\useCircForMS{\userectformsfalse}

1206 \newcommand{\useCheckForProof}{\symbolchoice{check}%
1207     \proofingsymbol{\ding{52}}}
1208 \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.

```

1209 \newcommand{\useCrossForProof}{\symbolchoice{cross}%
1210     \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.

```
1211 \def\MCcolor{black}
1212 \def\Ans@sq@l@driver{\edef\@linkcolor{\MCcolor}%
1213   \Rect{\makebox[\eq@tmpdima]{\linkContentWrapper}}}%
1214   \Ans@proofing{\eq@tmpdima}%
1215 }
```

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

```
1216 \def\Ans@sq@f@driver{%
1217   \centerWidget\RadioFieldSize
1218   \leavevmode\lower\eqcenterWidget\Bbox %\eqe@Bbox
1219   {\RadioFieldSize}{\RadioFieldSize}%
1220   \Ans@proofing{\RadioFieldSize}%
1221 }
```

Write quiz solutions to the exercise solutions file

```
1222 \def\eq@sq@llabel{\string\textbf{Solution to Quiz:}}
1223 \def\sqsllabel{\eq@sq@llabel}
```

`\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

```
1224 \let\quiz@solns\ex@solns
1225 \newcommand{\preExamSolnHead}{\goodbreak\noindent}
1226 \newcommand{\examSolnHeadFmt}[1]{\textbf{#1}}
1227 \newcommand{\postExamSolnHead}{\par\medskip}
```

Write to solution file if *not* solutions-only

```
1228 \@ifundefined{ifOKToWriteExamData}{\newif\ifOKToWriteExamData
1229   \OKToWriteExamDatatrue}{-}
1230 \newcommand{\writeToSolnFile}[1]{%
1231   \ifsolutionsonly\else\ifOKToWriteExamData
1232     \let\quiz@solns\ex@solns
1233     \set@display@protect
1234     \eqe@IWO\quiz@solns{#1}%
1235     \set@typeset@protect
1236   \fi\fi}
1237 \let\writeT@SolnFile\writeToSolnFile
```

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

```
1238 \def\eqe@writetoSolns#1{% dpsD17
1239   \ifsolutionsonly\else
```

```

1240 \ifOKToWriteExamData
1241 \set@display@protect
1242 \eqe@IW0\quiz@solns{\string\preExamSolnHead
1243   \string\examSolnHeadFmt{#1}\string\postExamSolnHead}%
1244 \set@typeset@protect\fi\fi}
1245 \def\eqe@writetoAux#1{%
1246   \set@display@protect
1247   \eqe@IW0\@auxout{#1}%
1248   \set@typeset@protect}

```

Turn off interactivity of short quiz.

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

```

1249 \def\Ans@proofing
1250 #1{%
1251   \ifeq@proofing\if\Ans@choice1\relax
1252     \llap{\rlap{\, \@proofingsymbol}\hspace#1\relax}%
1253   \fi\fi
1254 }

```

`\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.

```
1255 \def\eq@ExamLastPage{\csname eqExamLastPage\endcsname}
```

`\nPagesOnExam` is the number of pages in the exam.

```
1256 \newcommand{\nPagesOnExam}{\csname eqExamLastPage\endcsname}
```

`\nPagesOnQues` is the number of pages in the questions.

```

1257 \newcommand{\nPagesOfQues}{\ifvspacewithsolns
1258   \csname eqExamQuesLastPage\endcsname\else
1259   \csname eqExamLastPage\endcsname\fi}

```

`\nPagesOnSols` is the number of pages of solutions.

```

1260 \newcommand{\nPagesOfSols}{\def\eqExamNumPagesSols{0}%
1261   \csarg\ifx{eqExamLastPage}\relax\else
1262   \csarg\ifx{eqExamLastPage}\relax\else
1263   {\count\z@=\nPagesOnExam\relax\advance
1264     \count\z@-\nPagesOfQues\relax
1265     \xdef\eqExamNumPagesSols{\the\count\z@}}\fi\fi
1266   \eqExamNumPagesSols}

```

`\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`.

```

1267 \newcommand{\lastPageOfExam}[1]{\pageref{#1PageEnd}}
1268 \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 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.

```
1269 \newcommand{\eqe@defNumRefii}[1]{%
1270   \ifundefined{hyperref}{\let\@getsecondOf\@secondoftwo}
1271   {\let\@getsecondOf\@secondoffive}%
1272   \ifundefined{r@#1}{%
```

If the reference `r@#1` is undefined, define the value to be 0

```
1273   \csarg\gdef{nRefii@#1}{0}}{%
```

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

```
1274   \csarg\xdef{nRefii@#1}{\expandafter\expandafter\expandafter
1275   \@getsecondOf\csname r@#1\endcsname}%
1276   }%
1277 }
```

`\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.

```
1278 \def\eqe@numRefii#1{%
1279   \csarg\ifx{nRefii@#1}\relax 0\else
1280   \@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}
```

```
1281 \newcommand{\numLastPageOfExam}[1]{\eqe@numRefii{#1PageEnd}}
1282 \newcommand{\numFirstPageOfExam}[1]{\eqe@numRefii{#1PageBegin}}
```

```
1283 \newcommand{\makeRefsNums}{%
1284   \ifundefined{thePartNames}{\begingroup
1285     \def\##1{\PackageInfo{eqexam}{processing exam: ##1}%
1286     \eqe@defNumRefii
1287     {##1PageEnd}\eqe@defNumRefii{##1PageBegin}}%
1288     \thePartNames
1289   \endgroup}%
1290 }
```

```
1291 \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 1292 \newcommand{\lheadeq}[1]{\def\eq@lhead{#1}}
```

```
\rheadeqe 1293 \lheadeqe{\shortwebsubject/\shortwebtittle}
```

```
1294 \newcommand{\cheadeqe}[1]{\def\eq@chead{#1}}
```

```

1295 \cheadeqe{-- Page \arabic{page} of {\nPagesOnExam} --}
1296 \newcommand{\rheadeqe}[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.

```

1297 \rheadeqe{\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*.

```

1298 \@ifpackageloaded{fancyhdr}{}{%
1299     \let\lhead\lheadeqe
1300     \let\chead\cheadeqe
1301     \let\rhead\rheadeqe
1302 }

```

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

```

\eqExamRunHead 1303 \newcommand{\runExamHeader}{\eq@lhead\hfill\eq@chead\hfill\eq@rhead}
1304 \newcommand\eqExamRunHead{%
1305     \addtolength\textwidth{\oddsidemargin}%
1306     \noindent\hspace*{-\oddsidemargin}\makebox[\textwidth]
1307     {\runExamHeader}%
1308 }

```

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.

```

1309 \newcommand{\lfooteqe}[1]{\def\eq@lfoot{#1}}
1310 \lfooteqe{}
1311 \newcommand{\cfooteqe}[1]{\def\eq@cfoot{#1}}
1312 \cfooteqe{}
1313 \newcommand{\rfooteqe}[1]{\def\eq@rfoot{#1}}
1314 \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`)

```

1315 \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 document, perhaps for posting the solutions to the exam, or publication of a “pretest”.
`\cheadSol` Note that `\eqsolutionshook` is defined in `exerquiz/eqexam.def`. May be redefined.
`\rheadSol` The running header of the exam, when solutions are included at the end of the document, perhaps for posting the solutions to the exam, or publication of a “pretest”.
`\runExamHeaderSol` Note that `\eqsolutionshook` is defined in `exerquiz/eqexam.def`. May be redefined.
`\eqsolutionshook`

```

1316 \newcommand{\lheadSol}[1]{\def\eq@lheadSol{#1}}
1317 \lheadSol{\shortwebsubject/\shortwebtittle}
1318 \newcommand{\cheadSol}[1]{\def\eq@cheadSol{#1}}
1319 \cheadSol{-- Page \arabic{page} of {\eq@ExamLastPage} --}
1320 \newcommand{\rheadSol}[1]{\def\eq@rheadSol{#1}}
1321 \rheadSol{SOLUTIONS}

```

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

```

1322 \newcommand{\runExamHeaderSol}
1323     {\eq@lheadSol\hfill\eq@cheadSol\hfill\eq@rheadSol}

```

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

```

1324 \def\eq@solutionshook
1325 {%
1326     \gdef\eqExamRunHead{\addtolength\textwidth{\oddsidemargin}%
1327     \noindent\hspace*{-\oddsidemargin}\makebox[\textwidth]
1328     {\runExamHeaderSol}}%
1329 }

```

13.2 \maketitle definitions

`\maketitle` Standard L^AT_EX 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 has course name, exam name, and a place for the student to put his/her name.
`\maketitledesign` (2) second row has date and instructor. May be redefined.
`\altTitle`

Modify the title by redefining `\maketitledesign`, the `\maketitle` command itself has L^AT_EX commands in it that should not be changed.

```

1330 \newcommand\maketitledesign
1331 {%
1332     \makebox[\textwidth]{\normalsize
1333         \shortstack[l]{\strut\websubject\\\@date}\hfill
1334         \shortstack[c]{\webtitle\\\strut\@altTitle}\hfill
1335         \shortstack[l]{\strut\eq@ExamName\\\webauthor}}%
1336 }
1337 \def\altTitle#1{\def\@altTitle{#1}}
1338 \let\@altTitle\@empty
1339 \def\eqemaketitle
1340 {%

```

`\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}`.

```

1341 \EQEcalculateAllTotals
1342 \begingroup
1343 % \addtolength\textwidth{\oddsidemargin}%
1344 \noindent\hspace*{-\oddsidemargin}%
1345 \raisebox{.7in}[0pt][0pt]{\SubmitButton}%
1346 \maketitledesign
1347 \endgroup
1348 }

```

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.

```

1349 \ifeqfortextbook
1350 \ifpackageloaded{web}{\let\maketitle\web@save@maketitle}{}%
1351 \else
1352 \ifeqeuseclassmaketitle
1353 \@ifpackageloaded{web}{\let\maketitle\web@save@maketitle}{}%
1354 \else
1355 \let\maketitle\eqemaketitle
1356 \fi
1357 \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{nwfsc_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.

```

1358 \newcommand\placeCoverPageLogo[3]{%
1359 \def\eqe@insertLogo{\hbox to0pt{%
1360 \hspace*{#1}\smash{\raisebox{#2}{#3}}\hss}}
1361 \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.

```

1362 \def\eqexcoverpage{%
1363 \begingroup
1364 \pagenumbering{roman}
1365 \eqe@insertLogo
1366 \eqexcoverpagedesign
1367 \endgroup\newpage
1368 \pagenumbering{arabic}
1369 }

```

`\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-`
`\coverpageUniversityFmt` `subject` is used to provide a special subject for the cover page, different from
`\coverpageSubjectFmt` `\websubject`. The others are used for formatting.

```

\coverpageTitleFmt 1370 \newcommand{\coverpagesubject}[1]{\def\coverpage@subject{#1}}
1371 \let\coverpage@subject\empty
1372 \def\eqexamsubject{\ifx\coverpage@subject\empty\websubject
1373   \else\coverpage@subject\fi}
1374 \newcommand{\coverpageUniversityFmt}[1]{%
1375   \def\eqex@coverpageUniversityFmt{#1}}
1376 \coverpageUniversityFmt{\bfseries\large}
1377 \newcommand{\coverpageSubjectFmt}[1]{%
1378   \def\eqex@coverpageSubjectFmt{#1}}
1379 \coverpageSubjectFmt{\bfseries\large}
1380 \newcommand{\coverpageTitleFmt}[1]{%
1381   \def\eqex@coverpageTitleFmt{#1}}
1382 \coverpageTitleFmt{\bfseries\large}
1383 \newcommand\cpCID[1]{\def\cp@@CID{#1}}
1384 \let\cp@@CID\empty
1385 %\newcommand\coverpageCID[1]{\def\cpCID@argi{#1}}
1386 \newcommand{\cpSetCIDWidth}[1]{\bgroup\setlength{\@tempdima}{#1}
1387   \xdef\cpCID@argi{the\@tempdima}\egroup}
1388 \def\cp@CID{\bgroup\settowidth\eq@tmplengthA{\cp@@CID}%
1389   \@tempdima=\cpCID@argi\relax
1390   \advance\@tempdima-\eq@tmplengthA
1391   \eqe@idinfohl{\makebox[\cpCID@argi][l]{\cp@@CID}}\egroup}
1392 \cpSetCIDWidth{2.25in}
1393 %\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` (`\qeSumryVert`) or to the right (`\qeSumryHoriz`).
`\qeSumryHoriz` 1394 \def\qeSumryVert{\def\eqe@SumryVert{\par\vfill}%
1395 \let\eqe@SumryHoriz\relax
1396 \def\eqe@@SumryVert{\vspace{\stretch{-1}}\bigskip}}
1397 \def\qeSumryHoriz{\def\eqe@SumryHoriz{\hfill}\let\eqe@SumryVert\relax
1398 \let\eqe@@SumryVert\relax}

The default is a horizontal orientation.

```
1399 \qeSumryHoriz
```

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

```



```

1400 \def\eqe@coverPageNaming{0}
1401 \newcommand{\useUIPartNames}{\def\eqe@coverPageNaming{1}}
1402 \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**

```

1403 \newcommand{\customNaming}[2]{\@namedef{userCustom#1}{#2}}
1404 \newcommand{\cpSetNameAndIDWidth}[1]{\bgroup
1405   \setlength{\@tempdima}{#1}%
1406   \xdef\cp@SetNameAndIDWidth{\the\@tempdima}\egroup}
1407 \cpSetNameAndIDWidth{.45\textwidth}
1408 \newcommand{\cpEnclNameAndID}[1]{\def\cp@EnclNameAndID{#1}}
1409 \let\cp@EnclNameAndID\empty
1410 \newcommand{\cpSetHghtFrstLn}[1]{\def\cp@HghtFrstLn{#1}}
1411 \def\cp@HghtFrstLn{Opt}
1412 \def\cp@setHghtFrstLn{\rule[\cp@HghtFrstLn]{0pt}{0pt}}
1413 \newcommand\cpNameAndID{\noindent
1414   \edef\eqExamName@argii{\cp@SetNameAndIDWidth}%
1415   \edef\eqSID@argii{\cp@SetNameAndIDWidth}%
1416   \edef\cpCID@argi{\cp@SetNameAndIDWidth}%
1417   \cp@EnclNameAndID{\parbox[b]{\cp@SetNameAndIDWidth}{\parindentOpt
1418     \cp@setHghtFrstLn\eq@ExamName\\[2ex]
1419     \ifx\cp@CID\empty\eq@SID\else\cp@CID\fi\\[2ex]
1420     \ifx\use@email\eqe@YES\eq@Email\\[2ex]\fi
1421     \textcolor{\webauthor@color}{\webauthor}, \@date\vskipOpt}}}}

```

Now we present the definition of `\eqexcoverpagedesign`.

```

1422 \newcommand{\eqexcoverpagedesign}
1423 {%
1424   \begingroup
1425   \parindentOpt
1426   \thispagestyle{empty}
1427   % \addtolength\textwidth{\oddsidemargin}
1428   \vspace*{.1\textheight}
1429   \noindent%\hspace*{-\oddsidemargin}%
1430   \makebox[\linewidth]{\parbox{\linewidth}%
1431     {\eqex@coverpageUniversityFmt
1432       \color{\webuniversity@color}%
1433       \centering\webuniversity}}
1434   \par\vspace{.1\textheight}
1435   \noindent%\hspace*{-\oddsidemargin}%
1436   \makebox[\linewidth]{\parbox{\linewidth}%
1437     {\eqex@coverpageSubjectFmt
1438       \color{\websubject@color}%
1439       \centering\eqexamsubject}}
1440   \par\vspace{\bigskipamount}
1441   \noindent%\hspace*{-\oddsidemargin}%
1442   \makebox[\linewidth]{\parbox{\linewidth}%

```

```

1443     {\eqex@coverpageTitleFmt
1444       \color{\webtitle@color}%
1445       \centering\webtitle}}
1446   \par\vspace{\stretch{1}}
1447   \optionalpagematter
1448   \par\vspace{\stretch{1}}\cpNameAndID
1449   \eqe@SumryHoriz\eqe@SumryVert
1450   \sumryAnnots
1451   \endgroup
1452 }

```

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
1453 \newcommand{\cpSumryHeader}{\textbf{Exam Record}}
1454 \newcommand{\cpSumryPts}{\, \text{pts}}
1455 \newcommand{\cpSumryPage}{Page}
1456 \newcommand{\cpSumryTotal}{Total:}
1457 \newcommand{\cpSumryGrade}{Grade:}
1458 \newcommand{\cpSetSumryWidth}[1]{\bgroup\setlength{\@tempdima}{#1}
1459   \xdef\cp@SetSumryWidth{\the\@tempdima}\egroup}
1460 \cpSetSumryWidth{.5\textwidth}
1461 \newcommand{\cpUsefbox}{\let\cp@Usefbox\fbox}
1462 \cpUsefbox
1463 \newcommand{\cpNofbox}{\let\cp@Usefbox\mbox}
1464 \newcommand{\cprulelength}{1.5in}

```

A helper command used in `\cpSumrybyparts`

```

1465 \def\cp@IsertNaming#1{%
1466   \ifcase\eqe@coverPageNaming
1467   #1\or\@nameuse{userFriendly#1}\or
1468   \@nameuse{userCustom#1}\else#1\fi
1469 }

```

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

```

1470 \newcommand{\cpSumrybyparts}{%

```

```

1471 \eqe@@SumryVert\cp@Usefbox{%
1472 \ifx\cp@Usefbox\fbx
1473   \def\cp@sumryWdth{\cp@SetSumryWidth-2\fbxrule-2\fbxsep}\else
1474 \def\cp@sumryWdth{\cp@SetSumryWidth}\fi
1475 \begin{minipage}[b]{\cp@sumryWdth}\kern0pt
1476   \begin{flushleft}
1477     \setlength{\eqetmplengtha}{\cprulelength}%
1478     \edef\eqe@cprulelength{\the\eqetmplengtha}%
1479     \csarg\ifx{NumberOfParts}\relax
1480     \else
1481       \count2=0
1482       \medskip
1483       \cpSumryHeader\par\parskip\bigskipamount
1484       \def\##1{\advance\count2by\csname ##1total\endcsname
1485         \underbar{\makebox[Opt][l]{%
1486           \cp@IsertNaming{##1}}\hspace*{\eqe@cprulelength}}%
1487         / $\eqe@ptsFmt{\csname ##1total\endcsname}%
1488         \cpSumryPts$\par}\thePartNames
1489       \fi
1490       \underbar{\makebox[Opt][l]{\cpSumryTotal}}%
1491       \hspace{\eqe@cprulelength}/
1492       $\eqe@ptsFmt{\the\count2\relax}\cpSumryPts$ \par
1493       \underbar{\makebox[Opt][l]{\cpSumryGrade}}%
1494       \hspace{\eqe@cprulelength}}%
1495     \ifx\cp@Usefbox\fbx\medskip\fi
1496   \end{flushleft}\kern0pt
1497 \end{minipage}}\par
1498 }

```

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

```

1499 \newcommand{\cpSumrybypages}{%
1500 \eqe@@SumryVert\cp@Usefbox{%
1501 \ifx\cp@Usefbox\fbx
1502   \def\cp@sumryWdth{\cp@SetSumryWidth-2\fbxrule-2\fbxsep}\else
1503 \def\cp@sumryWdth{\cp@SetSumryWidth}\fi
1504 \begin{minipage}[b]{\cp@sumryWdth}\kern0pt
1505   \begin{flushleft}
1506     \setlength{\eqetmplengtha}{\cprulelength}%
1507     \edef\eqe@cprulelength{\the\eqetmplengtha}%
1508     \csarg\ifx{NumberOfParts}\relax
1509     \else\medskip\count0=0\relax\count2=0\relax
1510     \cpSumryHeader\par\parskip\bigskipamount
1511     \ifsolutionsAtEnd
1512       \@ifundefined{eqExamQuesLastPage}{\def\NUMPAGES{0}}
1513       {\edef\NUMPAGES{\@nameuse{eqExamQuesLastPage}}}%
1514     \else
1515       \@ifundefined{eqExamLastPage}{\def\NUMPAGES{0}}
1516       {\edef\NUMPAGES{\@nameuse{eqExamLastPage}}}%
1517     \fi

```

```

1518         \@whilenum\count0<\NUMPAGES\relax\do{%
1519             \advance\count0by1\relax
1520             \@ifundefined{Page\the\count0total}{\count4=0\relax}
1521             {\count4=\@nameuse{Page\the\count0total}\relax}%
1522             \csarg\ifx{Page\the\count0spilltotal}\relax
1523             \else\advance\count4by\csname%
1524                 Page\the\count0spilltotal\endcsname\relax\fi
1525             \advance\count2by\count4\relax\underbar{%
1526             \makebox[0pt][l]{\cpSumryPage~\the\count0}%
1527             \hspace*{\eqe@cprulelength}}/
1528             $\eqe@ptsFmt{\the\count4\relax}%
1529             \cpSumryPts$\csarg\ifx{Page\the\count0spilltotal}\relax
1530             \else\ $\(@nameuse{Page\the\count0spilltotal}%
1531                 \cpSumryPts+\@nameuse{Page\the\count0total}%
1532                 \cpSumryPts)$\fi\par
1533         }% while
1534     \fi
1535     \underbar{\makebox[0pt][l]{\cpSumryTotal}%
1536     \hspace{\eqe@cprulelength}}/
1537     $\eqe@ptsFmt{\the\count2\relax}\cpSumryPts$\par
1538     \underbar{\makebox[0pt][l]{\cpSumryGrade}%
1539     \hspace{\eqe@cprulelength}}%
1540     \ifx\cp@Usefbox\fbbox\medskip\fi
1541     \end{flushleft}}\kern0pt
1542 \end{minipage}}\par
1543 }

```

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

```
1544 \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 `\autocalcparts`. **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.

```
1545 \def\@markttotalvalue{\global\let\eqe@innermarkpts\relax
1546     \if@bypasseqexamheading\else\def\eqe@marktxt{%

```

```

1547     \theeqpointvalue\@nameuse{eqExam}\theeq@numparts}%
1548 \ifx\endparts\endexercise@parts@tabular
1549     \xdef\eqe@innermarkpts{\noexpand\mark{\eqe@marktxt}}\else
1550     \mark{\eqe@marktxt}\fi\fi}
1551 \def\probvalue#1#2{\addtocounter{eqpointvalue}{#1}%
1552     \ifdisplyPoints\marginpoints{#1}{#2}\fi
1553     \@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.

```
1554 \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.

```

1555 \newcommand{\marginboxdesign}[2] [] {%
1556     \parbox{\widthtpboxes}{\tabcolsep=0pt\relax
1557         \begin{tabular}{|c|}\hline
1558             \vrule height15pt width0pt#1\\\hline
1559             \makebox[\widthtpboxes-2\fbboxrule]{#2}\\\hline
1560         \end{tabular}}%
1561     }%
1562 }

```

`\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 strings to other languages.

```

\pointsLabel 1563 \providecommand{\ptsLabel}[1]{\def\eqptsLabel{#1}}\ptsLabel{pts}
1564 \providecommand{\ptLabel}[1]{\def\eqptLabel{#1}}\ptLabel{pt}
1565 \newcommand{\eachLabel}[1]{\def\eqeachLabel{#1}}\eachLabel{ea.}

```

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

```
1566 \newcommand{\pointsLabel}[1]{%
1567     \def\eqpointsLabel{#1}}\pointsLabel{points}
1568 \newcommand{\pointLabel}[1]{%
1569     \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` 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.

```
1570 \def\pointsAsText{\def\eqe@ptsFmt{\text}}
1571 \let\eqe@ptsFmt\relax
1572 \@onlypreamble\pointsAsText
```

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

```
1573 \newcommand{\marginpointtext}[2]{\ifnum#2=0\leftmarginPtsTxt{#1}\else
1574     \leftmarginPtsEaTxt{#2}\fi
1575 }
```

`\leftmarginPtsTxt` displays points in the left margin: (`#1Pt`) or (`#1Pts`), where `#1` is the number of points for this problem.

```
1576 \newcommand{\leftmarginPtsTxt}[1]{(\small$\eqe@ptsFmt{#1}^{\text{%
1577     \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi}}$)}
```

`\leftmarginPtsEaTxt` displays ‘points each’ in the left margin: (`#1Ptea.`) or (`#1Ptsea.`), where `#1` is the number of points for each part of this problem.

```
1578 \newcommand{\leftmarginPtsEaTxt}[1]{(\small$\eqe@ptsFmt{#1}_{\text{%
1579     \eqeachLabel}}^{\text{\ifnum#1=1\relax\eqptLabel\else
1580     \eqptsLabel\fi}}$)}
```

`\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`.

```
1581 \newcommand{\eqleftmargin}[2]{\makebox[0pt][r]{\marginpointtext{#1}{#2}%
1582     \setlength{\@tempdima}{\marginparsep+\eqemargin}%
1583     \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.

```
1584 \newcommand{\marginpointsboxtext}[2]{\small$\eqe@ptsFmt{#1}\,\text{%
1585   \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi}$}
```

`\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` and `\insertTotalsBoxPDF` These two commands are \letting 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.

```
1586 \def\@insertPointsBoxPDF
1587 {
1588   \def\fieldName{pointsgiven.\curr@quiz.page\thepage.%
1589     \theequationnoi}%
1590   \calcTextField[\F\FHidden\BC{ }\Q1\textColor{1 0 0 rg}]
1591     {\fieldName}{\widthtpboxes}{15pt}%
1592 }
1593 \def\@insertTotalsBoxPDF
1594 {
1595   \def\fieldName{pagetotals.\curr@quiz.page\thepage}%
1596   \calcTextField[\F\FHidden\BC{ }\Q1\textColor{1 0 0 rg}
1597     \AA{\AACalculate{AFSimple_Calculate("SUM",
1598       new Array("pointsgiven.\curr@quiz.page\thepage"))};}]
1599     {\fieldName}{\widthtpboxes}{15pt}%
1600 }
1601 \ifx\use@email\eqe@YES
1602   \let\insertPointsBoxPDF\@insertPointsBoxPDF
1603   \let\insertTotalsBoxPDF\@insertTotalsBoxPDF
1604 \else
1605   \let\insertPointsBoxPDF\@empty
1606   \let\insertTotalsBoxPDF\@empty
1607 \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.

```
1608 \newcommand{\eqrightmarginbox}[2]{%
1609   \makebox[0pt][l]{%
```

If this problem is in a minipage, (as declared by `\probInMinipage`), we adjust the text width to be the original text width, rather than the text width determined by the width of the minipage.

```
1610   \ifx\probInMinipage\relax
1611     \let\eq@TW\textwidth\else
1612     \let\eq@TW\eqTWSave\fi
1613   \setlength\tabcolsep{0pt}%
```

(2011/05/08) new calculation for the right point box.

```

1614 \setlength{\@tempdima}{\eq@TW-\eq@margin+\marginparsep}%
1615 \hspace*{\@tempdima}%
1616 \raisebox{-.5\height}[Opt][Opt]
1617   {\marginboxdesign[\insertPointsBoxPDF]}%
1618   {\marginpointsboxtext{#1}{#2}}%
1619   }\hfil
1620 }\let\probInMinip@ge\relax
1621 }

```

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.

```

1622 \newcommand{\eq@marginboxright}[2]{\makebox[Opt][l]{%
1623   \setlength\@tabcolsep{0pt}%
1624   \setlength{\@tempdima}{\textwidth+\marginparsep-\parindent}%
1625   \hspace*{\@tempdima}%
1626   \raisebox{-.5\height}[Opt][Opt]{%
1627     \marginboxdesign[\insertTotalsBoxPDF]}%
1628     {\marginpointsboxtext{#1}{#2}}%
1629   }\hfil
1630   }%
1631 }

```

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.

```

1632 \newcommand{\eq@marginboxleft}[2]{\noindent\makebox[Opt][r]{%
1633   \raisebox{-.5\height}[Opt][Opt]{%
1634     \marginboxdesign[\insertTotalsBoxPDF]}%
1635     {\marginpointsboxtext{#1}{#2}}%
1636   }\settowidth{\@tempdimb}{\eq@hspannerPrb}%
1637   \setlength{\@tempdima}{\marginparsep+\eq@margin-\@tempdimb}%
1638   \hspace*{\@tempdima}}\hfil
1639 }

```

`\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.

```

1640 \newcommand{\measurePtBoxHt}{\setbox\eq@pointbox\hbox{%
1641   \marginboxdesign{\marginpointsboxtext{00}{0}}%
1642   }\setlength\eq@pointboxtotalheight{.5\ht\eq@pointbox}%
1643   \xdef\halfHtPtBox{\the\eq@pointboxtotalheight}%
1644   \setlength\eq@pointboxtotalheight{\dp\eq@pointbox+\ht\eq@pointbox}%
1645   \xdef\totHtPtBox{\the\eq@pointboxtotalheight}%
1646   \xdef\dpPtBox{\the\dp\eq@pointbox}%
1647   \xdef\htPtBox{\the\ht\eq@pointbox}%
1648 }
1649 \measurePtBoxHt

```

A helper command to set both margin boxes.

```

1650 \newcommand{\eq@bothmargins}[2]{\eq@leftmargin{#1}{#2}%
1651   \eq@rightmargin{#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.

```
1652 \def\lastparttotal{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.

```
1653 \def\parsetotals#1\eqExam#2\end{\def\argi{#1}\def\argii{#2}}
```

```
1654 \def\stripeqExam#1\eqExam{\def\argii{#1}}
```

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

```
1655 \def\settotalsbox{%
```

```
1656     \expandafter\parsetotals\botmark\eqExam\end
```

```
1657     \ifx\argi\@empty\hfil
```

```
1658     \else\ifx\argii\@empty\hfil
```

```
1659         \else
```

```
1660             \expandafter\stripeqExam\argii
```

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

```
1661         \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.

```
1662         \addtocounter{eqpointsthispage}{-\value{eqpointsofar}}%
```

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

```
1663         \setcounter{eqpointsofar}{\argi}%
```

And save to the auxiliary file for later usage.

```
1664         \xdef\lastparttotal{\argii}%
```

```
1665         \eqe@IWO\@auxout{\string\csarg\string
```

```
1666             \gdef{Page\thepage total}%
```

```
1667             {\theeqpointsthispage}}%
```

Finally, place the totals box.

```
1668         \totalsbox
```

```
1669         \fi
```

```
1670     \fi
```

```
1671 }
```

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

```
1672 \newcommand\totalsboxtext{\small$
```

```

1673 \eqe@ptsFmt{\theeqpointsthispage}\,\text{%
1674 \ifnum\theeqpointsthispage=1\relax\eqptLabel\else
1675 \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.

1676 \newlength\eqevtranstotbox
1677 \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.

1678 \def\eqe@chkZeroTotals{\let\eqe@allow\eqe@YES
1679 \ifx\eqe@zeroTotalsAllowed\eqe@NO
1680 \ifnum\theeqpointsthispage=0 \let\eqe@allow\eqe@NO\fi\fi}
1681 \newcommand{\totalsboxleft}{\eqe@chkZeroTotals
1682 \ifx\eqe@allow\eqe@YES
1683 \makebox[Opt][r]{\setlength\tabcolsep{Opt}%
1684 \raisebox{-\height+\eqevtranstotbox}[Opt][Opt]{%
1685 \marginboxdesign[\insertTotalsBoxPDF]{\totalsboxtext}%
1686 }\hspace*{\marginparsep}%
1687 }\fi\hfil
1688 }
1689 \newcommand{\totalsboxright}{\eqe@chkZeroTotals
1690 \ifx\eqe@allow\eqe@YES
1691 \makebox[Opt][l]{\setlength\tabcolsep{Opt}%
1692 \hspace*{\textwidth}\hspace*{\marginparsep}%
1693 \raisebox{-\height+\eqevtranstotbox}[Opt][Opt]{%
1694 \marginboxdesign[\insertTotalsBoxPDF]{\totalsboxtext}%
1695 }%
1696 }\fi\hfil
1697 }

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!

1698 \ifeqe@nopoints\else\if@bypasseqexamheading\else
1699 \AtBeginDocument{\mark{0\csname eqExam\endcsname0}}
1700 \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.

1701 \AtEndDocument{%
1702 \eqe@IWO\@auxout{\string\csarg\string\gdef{NumberOfParts}%
1703 {\arabic{eq@numparts}}}%
1704 \eqe@IWO\@auxout{\string\csarg\string\gdef{thePartNames}%
1705 {\the\partNames}}%
1706 \clearpage\addtocounter{page}{-1}\writelastpage\relax
1707 \addtocounter{page}{1}%
1708 }

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

```

```
1709 \newcommand\theGrandTotal{%
1710     $\eqe@ptsFmt{\csname eqeGrandTotal\endcsname}$}
```

`\totalForPart` User access to the total for an 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.

```
1711 \newcommand{\totalForPart}[1]{$\eqe@ptsFmt{\csname#1total\endcsname}$}
1712 \def\tot@lForPart#1{\csname#1total\endcsname}
1713 \newcommand{\percentForPart}[1]{$\eqe@ptsFmt
1714     {\csname#1percent\endcsname}$\%}
```

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.

```
1715 \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`.

```
1716 \def\placeMarkerHere#1{%
1717     \eqe@IWO\@auxout{\string\csarg\string\gdef
1718         {#1SaveTotalHere}{\theeqpointvalue}}%
1719 }
```

`\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`.

```
1720 \newcommand{\calcFromMarkers}[3][\@markerTotalFmt]{%
1721     \@ifundefined{#2SaveTotalHere}{}%
1722         {\@ifundefined{#3SaveTotalHere}{\setcounter{markerCnt}%
1723         {\@nameuse{#2SaveTotalHere}-\@nameuse{#3SaveTotalHere}}%
1724         \ifnum\value{markerCnt}<0\relax
1725         \setcounter{markerCnt}{-\value{markerCnt}}\fi#1}}%
1726 }
```

`\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`

```

1727 \newcommand{\markerTotalFmt}[1]{\def\@markerTotalFmt{#1}}
1728 \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 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*.

```

1729 \newcommand{\calcQsBtwnMarkers}[2] []{\setcounter{markerCnt}{0}}
1730 \def\eqe@argi{#1}%

```

`\Mrk1thisQnum` is the question number in effect at the point where the command `\calcQsBtwnMarkers` is inserted.

```

1731 \eqe@IW0\@auxout{\string\csarg\string\gdef
1732 \#2thisQnum}{\thequestionnoi}}%

```

`\Mrk1Start` is 0 if undefined.

```

1733 \ifundefined{#2thisQnum}{\csarg\xdef{#2Start}{0}}%
1734 \eqe@IW0\@auxout{\string\csarg\string\gdef
1735 \#2Start}{0}}%
1736 }{%

```

`\Mrk1Start` is the current question number plus 1.

```

1737 \setcounter{markerCnt}{\value{questionnoi}+1}\csarg
1738 \xdef{#2Start}{\the\value{markerCnt}}%
1739 \eqe@IW0\@auxout{\string\csarg\string\gdef
1740 \#2Start}{\@nameuse{#2Start}}}%
1741 }

```

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 `\Mrk1thisQnum`.

```

1742 \ifx\eqe@argi\@empty\else
1743 \ifundefined{#2thisQnum}{\setcounter{markerCnt}{0}}%
1744 {\ifundefined{#1thisQnum}{\setcounter{markerCnt}{0}}{%
1745 \setcounter{markerCnt}%
1746 {\@nameuse{#1thisQnum}-\@nameuse{#2thisQnum}}}%
1747 }}%
1748 \fi

```

`\Mrk1nQs` is the number of questions that appear between *Mrk1* and *Mrk2*.

```

1749 \csarg\xdef{#2nQs}{\the\value{markerCnt}}%
1750 \eqe@IW0\@auxout{\string\csarg\string\gdef
1751 \#2nQs}{\the\value{markerCnt}}}%

```

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

```

1752     \ifx\eqe@argi\@empty\else
1753         \@ifundefined{#1thisQnum}{\csarg\xdef{#2End}{0}}%
1754         \eqe@IW0\@auxout{\string
1755             \csarg\string\gdef{#2End}{0}}%
1756     }{%
    \Mrk1End is \Mrk2thisQnum.
1757     \csarg\xdef{#2End}{\@nameuse{#1thisQnum}}%
1758     \eqe@IW0\@auxout{\string\csarg\string\gdef
1759         {#2End}{\@nameuse{#2End}}}%
1760     }%
1761     \fi
1762 }

```

`\markStartFor` We provide a user interface to the three macros defined above. The re-
`\markEndtFor` quired parameter is a mark (a name that was used as the first argument of
`\markNumQsFor` `\calcQsBtwnMarkers`).

```

1763 \newcommand{\markStartFor}[1]{\@nameuse{#1Start}}
1764 \newcommand{\markEndtFor}[1]{\@nameuse{#1End}}
1765 \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:

$$\text{\placeAtxy}\{x_dim\}\{y_dim\}\{content\}$$

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

```

1766 \newcommand{\placeAtxy}[3]{%
1767     \par\nointerlineskip
1768     \ifdisplayworkarea
1769         \ifx\eq@insertverticalspace\eqe@YES
1770             \makebox[0pt][l]{\hspace*{-\parindent}\hspace*{#1}}%
1771             \raisebox{#2}[0pt][0pt]{#3}\fi
1772     \fi
1773 }}

```

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}

```

1774 `\newsavebox{\workareasb}`

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

```

1775 \def\workareaVadj#1{\bgroup\def\eqe@rgi{#1}\ifx\eqe@rgi\@empty
1776 \def\eqe@rgi{0pt}\fi\setlength{\@tempdima}{\eqe@rgi}%
1777 \xdef\workareaV@dj{\the\@tempdima}\egroup
1778 } % dps1
1779 \workareaVadj{3pt}
1780 \def\workareaCmds#1{\def\@rgi{#1}\ifx\@rgi\@empty
1781 \let\work@reaCmds\relax\else
1782 \def\work@reaCmds{#1}}
1783 \let\work@reaCmds\relax
1784 \def\priorworkareaCmds#1{\def\@rgi{#1}\ifx\@rgi\@empty % dpsj6
1785 \let\priorWorkAreaCmds\relax\else
1786 \def\priorWorkAreaCmds{#1}\fi}
1787 \let\priorWorkAreaCmds\relax

```

```

1788 \newenvironment{workarea}[2][\linewidth]
1789 {%
1790 \removelastparskip
1791 \edef\workareadepth{\if\currhideopt HOpt\else#2\fi}%
1792 \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.

```

1793 \ifx\makeVgrid\relax\else % dps2
1794 \ifforceEqualCells
1795 \ifx\fillLinesLineWidth\@empty\else
1796 \setlength{\eqetmplengtha}{\fillLinesLineWidth}\fi
1797 \fi
1798 \fi
1799 \begin{minipage}[b][\workareadepth][t]{\eqetmplengtha}%

```

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

```

1800 \vglue\workareaV@dj

```

Adjust `\leftskip` when the problem is a lead-in.

```

1801 \@ifundefined{leadinIndentPrtSep}{}{\ifx\solutionparshape\@empty
1802 \else\leftskip\leadinIndentPrtSep\fi}\work@reaCmds
1803 }{%
1804 \end{minipage}\end{lrbox}%

```

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

```

1805     \ifeq@solutionsafter\else
1806     \par\ifdisplayworkarea
1807         \ifx\eq@insertverticalspace\eqe@YES
1808             \removeastparskip
1809             \vglue-\baselineskip
1810             \if\currhideopt H\else
1811                 \noindent\strut\smash{\usebox{\workareasb}}%
1812             \fi\fi
1813     \fi\fi
1814 }

```

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][height]
\begin{panel}[l|r]
...
\end{panel}
\begin{solution}
...
\end{solution}
\end{splitsolution}

```

Old Syntax

```

\begin{splitsolution}{height}
\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.

```

1815 \def\splitsolution{%
1816     \@ifnextchar[{\splitsolutioni}%
1817         {\splitsolutioniii{\panelwidth}{\panelheight}}}%
1818 }
1819 \def\splitsolutioni[#1]{%
1820     \@ifnextchar[{\splitsolutionii{#1}}
1821         {\splitsolutioniii{\panelwidth}{#1}}}%
1822 }
1823 \def\splitsolutionii#1[#2]{%
1824     \splitsolutioniii{#1}{#2}%
1825 }
1826 \def\splitsolutioniii#1#2{%
1827     \@ifundefined{panel@write}{\newwrite\panel@write}{}%
1828     \gdef\ss@Argiii{#1}\gdef\ss@Argii{#2}%

    Here, we \let \solution to \sssolution and replace it with \@ssSolution
1829     \let\@sssolution\solution
1830     \let\end@sssolution\endsolution
1831     \let\solution\@ssSolution
1832     \let\endsolution\end@ssSolution
1833     \par\noindent\ignorespaces

    The panel environment should appear next, just after the \begin{splitslution}
1834 }
1835 \def\endsplitsolution{%
1836     \ifeq@solutionsafter\solutionsafterSkip\fi % dpsj10
1837     \ifx\solutionparshape\@empty\else
1838         \advance\linewidth-\leadinIndentPrtSep\fi %%dps
1839     \edef\eqe@workwidth{\if\ss@Argi 1\noexpand\ss@Argiii
1840         \else\noexpand\linewidth\fi}%
1841     \edef\eqe@workfill{\if\ss@Argi 1\else\hfill\fi}%
1842     \ifeq@solutionsafter\else
1843         \begin{workarea}[\eqe@workwidth]{\eqe@depth}%
1844         \let\panelGetDimen\relax%
1845         \eqe@workfill\input{panel\thepanel@cnt.cut}%
1846         \end{workarea}
1847     \fi\ifeq@solutionsafter\else\solutionsafterSkip\fi
1848 }

```

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

```

panel@cnt 1849 \newcommand\panelgap{3pt}
1850 \newsavebox{\eqpanelbox}
1851 \newdimen\eqepanelwidth
1852 \newdimen\eqepanelheight

```

`\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).

```
1853 \newcommand{\getDimSSPanel}[1]{%
1854   \sbox{\eqpanelbox}{#1}%
1855   \eqpanelwidth=\wd\eqpanelbox
1856   \eqpanelheight=\ht\eqpanelbox
1857   \advance\eqpanelheight by\the\dp\eqpanelbox
1858 }
```

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

```
1859 \def\panelwidth{\eqpanelwidth}
1860 \def\panelheight{\eqpanelheight}
1861 \newcounter{panel@cnt}
```

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

```
1862 \def\panel{\def\eqe@panelir{\eqe@paneli[r]}%
1863   \@ifnextchar[{\eqe@paneli}
1864     {\@ifnextchar\relax{\expandafter\eqe@panelir\@gobble}%
1865       {\eqe@paneli[r]}}%
1866 }
```

We continue the `panel` environment.

```
1867 \def\eqe@paneli[#1]{%
1868   \gdef\ss@Argi{#1}\def\eqe@l{1}\def\eqe@r{r}%
1869   \ifx\ss@Argi\@empty\gdef\ss@Argi{r}\else
1870   \ifx\eqe@l\ss@Argi\else\ifx\eqe@r\ss@Argi\else
1871   \PackageError{eqexam}{Permissible arguments for panel are l and r}
1872   {Use l or r for the argument of panel.}\fi\fi\fi
1873   \stepcounter{panel@cnt}%
1874   \immediate\openout \panel@write panel\thepanel@cnt.cut
```

Manage the `h`, `H`, and global overrides.

```
1875   \if\currhideopt H%
1876     \eqe@IW0\panel@write{\vfill}%
1877     \immediate\closeout\panel@write
1878     \gdef\ss@Argii{Opt}%
1879   \else
1880     \if\currhideopt h\ifeq@solutionsafter\else
1881     \ifeq@globalshowsolutions\else
1882     \eqe@IW0\panel@write{\vfill}%
1883     \immediate\closeout\panel@write
1884     \gdef\ss@Argii{Opt}%
1885     \fi\fi
1886   \fi
1887 \fi
1888 \begingroup
```

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

```
1889 \let\verbatim@out\panel@write
```

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

```
1890     {\lccode'C='\}\lccode'P='\{
1891         \lowercase{\eqe@IW0\verbatim@out{\string\panelGetDimen PC}}}%
```

Followed by the verbatim listing of the panel environment.

```
1892     \verbatimwrite
1893 }
1894 \def\endpanel{%
1895     \endverbatimwrite
```

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

```
1896     {\lccode'P='\}\lowercase{\eqe@IW0\verbatim@out{P}}}%
1897     \immediate\closeout\panel@write
1898     \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.

```
1899     \gdef\eqePANELCUT{%
1900         \let\panelGetDimen\getDimSSPanel
1901         \input{panel\thepanel@cnt.cut}%
1902         \setlength{\eq@tmpdima}{\ss@Argiii}%
1903         \xdef\ss@Argiii{\the\eq@tmpdima}%
1904         \xdef\mp@Width{\ifeq@solutionsafter\linewidth
1905             \else\linewidth-\ss@Argiii-\panelgap\fi}%
1906     }\aftergroup\eqePANELCUT
1907 }
```

`\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`.

```
1908 \long\def\eqe@IW#1{%
1909     \ifeq@solutionsafter
1910         \let\eqe@next\@empty
1911     \else
1912         \def\eqe@next{\eqe@IW0\verbatim@out{#1}}%
1913         \ifeq@nosolutions\else
1914             \if\currhideopt H%
1915                 \gdef\ss@Argii{0pt}\let\eqe@next\@empty
1916             \else
1917                 \if\currhideopt h%
1918                     \ifeq@globalshowssolutions\else
1919                         \gdef\ss@Argii{0pt}\let\eqe@next\@empty
1920                     \fi
1921                 \fi
1922             \fi
1923         \fi
1924     \fi
1925     \eqe@next
```

1926 }

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`.

```
1927 \newenvironment{@ssSolution}[1][\ss@Argii]%
1928 {%
1929     \def\eqe@argi{#1}\ifx\eqe@argi\@empty\else
1930         \setlength{\@tmpdima}{\ss@Argii}% chng
1931         \edef\ss@Argii{\the\@tmpdima}%
1932         \let\soln@keys@nLines\@empty
1933         \edef\panelheight{\the\panelheight}%
1934         \edef\panelwidth{\the\panelwidth}%
1935         \edef\temp@exp{\noexpand
1936             \setkeys*{soln@keys}{#1}}\temp@exp
1937         \ifx\soln@keys@nLines\@empty
1938             \edef\ss@Argii{\XKV@rm}%
1939         \else
1940             \@tmpdima\wlVspace\relax
1941             \@tmpdima=\soln@keys@nLines\@tmpdima
1942             \edef\ss@Argii{\the\@tmpdima}%
1943             \ifx\XKV@rm\@empty\else
1944                 \if@equsedim\edef\ss@Argii{\XKV@rm}\fi\fi
1945         \fi
1946         \setlength{\eq@tmpdima}{\ss@Argii}%
1947 %         \ifx\sameVspace\@empty\gdef\sameVspace{0pt}\fi
```

If the `panelheight` is greater than the requested height, we make the requested height equal to `panelheight`.

```
1948     \ifdim\eqepanelheight>\eq@tmpdima
1949         \eq@tmpdima\eqepanelheight\fi
1950     \xdef\ss@Argii{\the\eq@tmpdima}%
1951 \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`.

```
1952 \xdef\eqedepth{%
1953     \ifvspacewithsolns\ss@Argii\else
1954         \ifeq@nosolutions\ss@Argii\else
1955         \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.

```
1956 \xdef\mp@Width{\ifeq@solutionsafter\noexpand\linewidth\else
1957     \noexpand\linewidth-\ss@Argiii-\panelgap\fi}%
```

If its a left panel...

```
1958 \if\ss@Argi l%
1959     \ifeq@solutionsafter
1960     \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.

```

1961             {\let\panelGetDimen\relax
1962             \input{panel\thepanel@cnt.cut}\vfill}\hfill}%
1963             \def\eqe@rPanel{\@empty}%
1964     \else
    For the left panel, we need to subtract \eqmargin
1965             \def\eqe@subleftgutter{-\string\eqmargin}%
1966             \def\eqe@lPanel{%
1967                 \string\parbox[b][\string\eqedepth]{\ss@Argiii}%
1968                 {\string\let\string\panelGetDimen\string\relax
1969                 \string\input{panel\thepanel@cnt.cut}}%
1970                 \string\vfill}\string\hfill^^J}%
1971             \def\eqe@rPanel{\@empty}%
1972     \fi
1973 \else
    If its a right panel...
1974     \ifeq@solutionsafter
1975         \def\eqe@lPanel{\@empty}%
1976         \def\eqe@rPanel{%
1977             \hfill\parbox[b][\eqedepth][t]{\ss@Argiii}
1978             {\let\panelGetDimen\relax
1979             \hfill\input{panel\thepanel@cnt.cut}\vfill}}%
1980     \else
    For the right panel, we do not to subtract \eqmargin as above.
1981         \let\eqe@subleftgutter\@empty
1982         \def\eqe@lPanel{\@empty}%
1983         \def\eqe@rPanel{%
1984             \string\hfill\string\parbox[b][\string\eqedepth]
1985             {\ss@Argiii}{\string\endgraf%
1986             \string\noindent\string\hfill%
1987             \string\let\string\panelGetDimen\string\relax
1988             \string\input{panel\thepanel@cnt.cut}}%
1989             \string\vfill}}%
1990     \fi
1991 \fi
1992 \let\verbatim@out\ex@solns
1993 \par\ifeq@solutionsafter\smallskip\fi
    Since everything is put a box, we set the \linewidth, and set \solutionparshape
    to \@empty.
1994     \ifx\solutionparshape\@empty\else
1995         \advance\linewidth-\leadinIndentPrtSep\fi %%dps
1996     \let\solutionparshape\@empty
1997     \noindent\minipage{\linewidth}%
1998     \if\ss@Argi l\noindent\parbox[b][\eqedepth][t]{\ss@Argiii}%
1999         {\vfill}\hfill\fi
2000     \minipage[b][\eqedepth][t]{\mp@width}%

```

```

2001   {\lccode'C='%\lowercase{\eqe@IW{%
2002     \string\def\string\panelgap{\panelgap}C^^J%
2003     \string\setlength{\string\panelwidth}{\ss@Argiii}C^^J%
2004     \string\setlength{\string\linewidth}{\string
2005       \linewidth-\string\eqemargin}C^^J%
2006     \string\def\string\eqedepth
2007       {\ifeq@nosolutions\ss@Argii\else\ss@Argii\fi}C}}}%

```

(2020/03/14) Change to `\noindent\hskip\eqemargin` to get the margin correct within the `eqequestions` environment on the solutions page.

```

2008   {\lccode'C='%\lowercase{\eqe@IW{\string\vskip2pt
2009     \string\noindent\string\hskip\string\eqemargin
2010 %   \string\strut\string\noindent%
2011 %   \string\strut\par\string\nobreak\string\noindent%
2012 %   \string\vskip2pt\string\noindent%
2013     \string\hbox\space to\string\linewidth\bgroup^^J%
2014     \eqe@lPanel\string\minipage[b][\string\eqedepth][t]%
2015     {\string\linewidth\eqe@subleftgutter%
2016       -\string\panelwidth-\string\panelgap}C
2017   }}}%
2018   \ifeq@solutionsafter
2019     \noindent\strut\hbox to\linewidth\bgroup
2020     \eqe@lPanel
2021     \minipage[b][\eqedepth][t]{\linewidth-\ss@Argiii-\panelgap}%
2022   \def\solutionsafterSkip{}%
2023   \fi

```

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

```

2024   \ifx\soln@keys@nLines@empty
2025     \edef\passedss@Argii{\ss@Argii}\else
2026     \edef\passedss@Argii{\ss@Argii,nLines=\soln@keys@nLines}\fi

```

For a split solution, cannot allow align left

```

2027   \if\ss@Argi 1\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.

```

2028   \ifanswerkey % dpsj10
2029     \gdef\p@ssToFLs{\let\eq@insertverticalspace\eqe@NO
2030     \@qlinedfillerfalse}\fi % dpsj10
2031   \eqe@fextendedfalse
2032   \expandafter\@sssolution\expandafter[\passedss@Argii]%
2033 }{%
2034   \eqe@IW{\string\endminipage\eqe@rPanel\egroup}%
2035   \ifeq@solutionsafter\endminipage\eqe@rPanel\egroup\fi
2036   \end@sssolution
2037   \endminipage\endminipage
2038 }
2039 % Redefine the \cs{paragraph} command
2040 %   \begin{macrocode}

```

```

2041 \renewcommand{\paragraph}
2042     {\@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.

2043 \newcommand{\defaultInstructions}[1]{\def\eq@default@Instructions{#1}}
2044 \defaultInstructions{Instructions.}

instructions Each test, or a part of a test usually have instructions. This instruction envi-
ronment 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.

2045 \newcommand{\beforeInstrSkip}{1ex}
2046 \newcommand{\afterInstrSkip}{-0em}
2047 \newcommand{\hInstrSpace{\ }
2048 \newcommand{\styleInstr{\bfseries}
2049 \newenvironment{instructions}[1][\eq@default@Instructions]{%
    If \summaryPointTotal is zero, show no summary points.
2050     \@ifundefined{\thisexamlabel total}{\let\eq@nosummarytotals\eqe@YES}
2051     {\ifnum\summaryPointTotal=0 \let\eq@nosummarytotals\eqe@YES\fi}%
2052     \expandafter\def\expandafter\eq@argi\expandafter{#1}%
2053     \def\hsi{\hInstrSpace}%
    We start a paragraph environment
2054     \@startsection{paragraph}{4}{\z@}%
2055     {\beforeInstrSkip\space\@plus1ex \@minus.2ex}%
2056     {\normalfont\normalsize\styleInstr}*%
2057     {\textcolor{\@instructionsColor}{#1}\normalcolor%
2058     \ifx\eq@nosummarytotals\eqe@YES\else\ifx\eq@argi\@empty\else\hsi\fi
2059     {\normalfont\summaryTotalsTxt}\hsi\fi}\hskip-\lastskip
2060     \ifx\eq@nosummarytotals\eqe@YES\ifx\eq@argi\@empty\else\hsi\fi\fi
2061     \normalfont\normalsize\ignorespaces
2062 }{\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.

2063 \newcommand\summaryPointTotal{\csname\thisexamlabel total\endcsname}
2064 \newcommand\summaryTotalsTxt{(\$ \eqe@ptsFmt{\summaryPointTotal}\, \text{%
2065     \@ifundefined{\thisexamlabel total}{\eqpointsLabel}%
2066     {\ifnum\summaryPointTotal=1\relax\eqpointLabel\else
2067     \eqpointsLabel\fi}}\$)}
2068 \newcommand{\nQuesInExam}[1][\thisexamlabel]{%
2069     \def\eqe@argi{#1}%
2070     \ifx\thisexamlabel\@empty\ifx\eqe@argi\thisexamlabel
2071     \PackageError{eqexam}{The optional argument for
2072     \string\nQuesInExam\MessageBreak must be specified}{\fi\fi
2073     \csname#1nQuestions\endcsname}

```

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

```

2074 \ifx\hyper@anchor@undefined\else
2075   \renewcommand\theHeqquestionnoi
2076     {\curr@quiz.\theeqquestionnoi\eqe@fpmrk}
2077   \renewcommand\theHquizno{%
2078     \if\probstar*\curr@quiz.%
2079       \theeqquestionnoi.part\thepartno.\arabic{quizno}%
2080     \else
2081       \curr@quiz.%
2082       \theeqquestionnoi.\arabic{quizno}%
2083     \fi
2084   }
2085   \renewcommand\theHpartno{\curr@quiz.%
2086     \theeqquestionnoi.part\thepartno}
2087 \fi

```

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

```

2088 \newcommand{\beforeCommentSkip}{1.25ex}
2089 \newcommand{\afterCommentSkip}{-1ex}
2090 \newcommand\hCommSpace{\ }
2091 \newcommand\styleComm{\bfseries}
2092 \newenvironment{eqComments}[1][\strut]{\removelastskip
2093   \def\eqe@argi{#1}\def\eqe@Strut{\strut}\def\hsc{\styleComm}%
2094   \@startsection{paragraph}{4}{\z@}%
2095   {\beforeCommentSkip\space\@plus1ex \@minus.2ex}%
2096   {\afterCommentSkip}{\normalfont\normalsize\styleComm}*%
2097   {\textcolor{\@eqCommentsColor}{#1}}%
2098   \ifx\eqe@argi\eqe@Strut\hskip\afterCommentSkip
2099   \else\ifx\eqe@argi\@empty\hskip\afterCommentSkip\else
2100     \hsc\strut\fi\fi\normalfont\normalsize
2101   \color{\@eqCommentsColorBody}\ignorespaces
2102 }{\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.

```

2103 \let\tb@beginexam@code\relax
2104 \def\partialspillovertotals{0}
2105 \def\exambegdef
2106 {%
2107   \csname\thisexamlabel pagemark\endcsname
2108   \@ifundefined{partialtotalpg}{\z@}%
2109     \ifnum\partialtotalpg=\arabic{page}%
2110       {\count0=\partialspillovertotals

```

```

2111         \advance\count0by\partialtotaleoe
2112         \xdef\partialspillovertotals{\the\count0}%
2113         \eqe@IWO\@auxout{\string\csarg\string\gdef
2114         {Page\partialtotalpg spilltotal}{\partialtotaleoe}}%
2115     }%
2116     \fi
2117 }%
2118 \csarg\ifx{\thisexamlabel pageno}\relax
2119 \else
2120     \csarg\ifx{pagenofirstprob\thisexamlabel}\relax
2121     \else
2122         \csarg\ifnum{\thisexamlabel pageno}
2123         < \@nameuse{pagenofirstprob\thisexamlabel}%
2124     \else
2125         \csarg\ifx{\thisexamlabel pagemark}\relax
2126     \else
2127         \ifnum\value{page}=%
2128             \csname\thisexamlabel pageno\endcsname
2129             \eqe@IWO\@auxout{\string\csarg\string\gdef
2130             {\thisexamlabel pagemark}{\string\newpage}}%
2131     \fi
2132     \fi
2133     \fi
2134     \fi
2135 \fi
2136 \setcounter{eqquestionnoi}{0}\setcounter{eqpointvalue}{0}%
2137 \setcounter{eqpointsofar}{0}\setcounter{eqpointsthispage}{0}%
2138 \setcounter{eq@count}{0}%
    We wrote \begin{eqequestions} to the top of the solutions file (\jobname.sol.
2139     \writeBeginEqeQuestions
2140     \label{\thisexamlabel PageBegin}%
2141     \eqe@IWO\@auxout{\string\csarg\string\gdef
2142     {\thisexamlabel pageno}{\thepage}}%
2143     \ifeqfortextbook
2144         \global\examenvtrue\tb@beginexam@code
2145     \fi
2146 }

```

\examendef 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.

```

2147 \def\tb@insmargmark{\ifisinstred\ifismarginans
2148     \insMidMarg{\mark{}}\fi\fi}
2149 \def\eqe@adjForSepRule{\ifx\eq@parttotals\eqe@YES
2150     \eq@pointboxtotalheight\halfHtPtBox\relax
2151     \advance\eq@pointboxtotalheight\dpPtBox\relax
2152     \advance\eq@pointboxtotalheight\pointsmarginparpush
2153     \vspace*{\eq@pointboxtotalheight}\fi}
2154 \def\eqe@afterexamsepcode{%
2155     \ifx\eq@parttotals\eqe@YES

```



```

2156 \@actionsAtPageBreak{\global\let\@spacetobreak\eqe@One}%
2157 {\global\let\@spacetobreak\eqe@Zero}%
2158 \ifx\@spacetobreak\eqe@One
2159 \bgroup\@tempdima=\pagetotal
2160 \advance\@tempdima\eqe@pointboxtotalheight
2161 \ifdim\@tempdima>\pagegoal\aftergroup\newpage
2162 \else
2163 \ifnum\arabic{eqe@count}>\z@
2164 \removeLastskip\vskip6pt\kern0pt
2165 \ifx\@reportpoints\eqe@One\else\@checkSpacing{0}\fi
2166 \def\@emitPartTotalsBox{\textcolor{\endexamtotal@color}%
2167 {\eqeomarginbox{\arabic{eqe@count}}{0}}}%
2168 \insertContent\@emitPartTotalsBox
2169 \fi
2170 \ifx\eqx@separationrule\eqe@YES
2171 \def\@emitSepRule{\separationrule\eqe@adjForSepRule}%
2172 \insertContent\@emitSepRule
2173 \fi
2174 \fi\egroup
2175 \else
2176 \ifnum\arabic{eqe@count}>\z@
2177 \ifx\@reportpoints\eqe@One\else\@checkSpacing{0}\fi
2178 \def\@emitPartTotalsBox{\textcolor{\endexamtotal@color}%
2179 {\eqeomarginbox{\arabic{eqe@count}}{0}}}%
2180 \insertContent\@emitPartTotalsBox
2181 \fi
2182 \ifx\eqx@separationrule\eqe@YES
2183 \def\@emitSepRule{\separationrule\eqe@adjForSepRule}%
2184 \insertContent\@emitSepRule
2185 \else
2186 \eqe@adjForSepRule
2187 \fi
2188 \fi
2189 \else
2190 \ifx\eqx@separationrule\eqe@YES
2191 \def\@emitSepRule{\separationrule\bigskip}%
2192 \insertContent\@emitSepRule\fi
2193 \fi
2194 }
2195 \def\insertContent#1{%
2196 \@ifundefined{NextAfter\currExamName}
2197 {#1}
2198 {\edef\x{%
2199 \numFirstPageOfExam{\@nameuse{NextAfter\currExamName}}}%
2200 \edef\y{\numLastPageOfExam{\currExamName}}%
2201 \ifnum\x=\y\relax#1\fi
2202 }%
2203 }
2204 \def\examenddef
2205 {%

```

```

2206 \global\let\partialtotalear\relax
2207 \global\let\partialtotalpg\relax
2208 \global\let\afterexamsepcode\relax
2209 \csarg\ifx{NumberOfParts}\relax
2210 \else
2211     \ifnum\value{eq@numparts}<\NumberOfParts
2212         \setcounter{eq@count}{\value{eqpointvalue}}%
2213         \addtocounter{eq@count}{-\value{eqpointsofar}}%
2214         \xdef\partialtotalear{\arabic{eq@count}}%
2215         \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.

```

2216     \global\let\afterexamsepcode\eqe@afterexamsepcode
2217     \fi
2218     \fi
2219     \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}`.

```

2220     \writeEndEqeQuestions
2221     \writetotalstoaux
2222     \addtocounter{page}{-1}%
2223     \writelastpage[\thisexamlabel]\addtocounter{page}{1}%
2224     \ifeqfortextbook\tb@insmargmark\fi
2225 }
2226 \def\actionsAtPageBreak#1#2{%
2227     \bgroup\@tempdima\pagegoal\advance\@tempdima-\pagetotal
2228     \@tempdimb\@fvsizeskip\vsizel
2229     \ifdim\@tempdima < \@tempdimb #1\else #2\fi\egroup
2230 }

```

`\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.

```

2231 \newcommand{\separationrule}{\makebox[\linewidth-\eqmargin][c]{
2232     {\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 \LaTeX documents. We give control over the page layout so an `eqexam` document can be used within a textbook.

```

2233 \def\eqe@hspannerPrb{\ }
2234 \@ifundefined{ifwithinsoldoc}{\newif\ifwithinsoldoc\withinsoldocfalse}{
2235 \newcommand{\eqquestopsep}[1]{\def\eqques@topsep{#1}}
2236 \providecommand{\eqquesparsep}[1]{\def\eqques@parsep{#1}}
2237 \newcommand{\eqquesitemsep}[1]{\def\eqques@itemsep{#1}}
2238 \newcommand{\eqqueslistparindent}[1]{\def\eqques@listparindent{#1}}
2239 \eqquestopsep{0pt}

```

```

2240 \eqequesparsep{Opt}
2241 \eqequesitemsep{Opt}
2242 \eqequeslistparindent{Opt}
2243 \newif\iffirstitem
2244 \def\eqeSolnItemMngt{\iffirstitem\global\firstitemfalse
2245   \let\eqe@next\@empty\else\let\eqe@next\item\fi\eqe@next\relax}
2246 \newenvironment{eqequestions}{%
2247   \begin{list}{}{%
2248     \ifwithinsoldoc\let\solnItemMngt\eqeSolnItemMngt\fi
2249     \setlength{\labelwidth}{\eqemargin}%
2250     \setlength{\parsep}{\eqeques@parsep}%
2251     \setlength{\itemsep}{\eqeques@itemsep}%
2252     \setlength{\topsep}{\eqeques@topsep}%
2253     \setlength{\itemindent}{0pt}%
2254     \setlength{\listparindent}{\eqeques@listparindent}%dps%
2255     \ifwithinsoldoc\settowidth{\labelsep}{\eqe@hspannerSoln}\else
2256     \settowidth{\labelsep}{\eqe@hspannerPrb}\fi
2257     \setlength{\leftmargin}{\labelwidth}%
2258   }\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.

```

2259 \def\setDefaultfvsskip#1{\def\default@fvsskip{#1}%
2260   \def\@fvsskip{#1}}
2261 \def\default@fvsskip{.3}
2262 \edef\@fvsskip{\default@fvsskip}
2263 \newcommand{\fvsskip}[1]{\def\@fvsskip{#1}}
2264 \def\autoExamName{exam\the\value{eq@numparts}}
2265 \def\nNumberOfParts{\csname NumberOfParts\endcsname}
2266 \abovesqskip{}
2267 \let\eqeWrtExamTitleToSolns\eqe@YES
2268 \def\wrtExamTitleInSolns{\let\eqeWrtExamTitleToSolns\eqe@YES}
2269 \def\noExamTitleInSolns{\let\eqeWrtExamTitleToSolns\eqe@NO}
2270 \let\thisexamlabel\@empty
2271 \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.

```

2272 \newif \ifDoNotRecordThisExam \DoNotRecordThisExamfalse % dpsf02
2273 \def\RecordThisExamOff{\DoNotRecordThisExamtrue} % dpsf02
2274 \def\bIFFalseWrtSolns{\writeT@SolnFile{\protect\iffalse^^J}}
2275 \def\eIFFalseWrtSolns{\writeT@SolnFile{\protect\fi^^J}}

```

The beginning of the `exam` environment.

```

2276 \let\currExamName\@empty
2277 \let\prevExamName\@empty
2278 \let\nextExamName\@empty

```

```

2279 \newenvironment{exam}[2] []
2280 {%
2281   \xdef\eqTWSave{\the\textwidth}%
2282   \ifDoNotRecordThisExam
2283     \expandafter\bIFFalseWrtSolns\fi
2284   \makeRoomForProb{\@fvsize\skip\textheight}{0}%
2285   \ifx\currExamName\@empty % chng
2286     \xdef\currExamName{#2}\else
2287     \xdef\prevExamName{\currExamName}%
2288     \eqe@IWO\@auxout{\string\csarg\string
2289       \gdef{NextAfter\prevExamName}{#2}}%
2290     \xdef\currExamName{#2}%
2291     \eqe@IWO\@auxout{\string\csarg\string
2292       \gdef{PrevTo\currExamName}{\prevExamName}}%
2293   \fi
2294   \let\isInExamEnv\eqe@YES
2295   \stepcounter{eq@numparts}%
2296   If #2 is empty, use \autoExamName
2297   \def\eqexamargii{#2}\ifx\eqexamargii\@empty
2298     \edef\eqexamargii{\autoExamName}\fi
2299   \xdef\thisexamlabel{\eqexamargii}\xdef\curr@quiz{\eqexamargii}%
2300   \def\eqexamargi{#1}\ifx\eqexamargi\@empty
2301     \edef\eqexamargi{\eqexamargii}\fi
2302   \expandafter\gdef\expandafter\thisUFexamlabel
2303   \expandafter{\eqexamargi}%
2304   \edef\eq@tmp{\the\partNames\string\{\eqexamargii}}%
2305   \global\partNames=\expandafter{\eq@tmp}%
2306   \csarg\ifx{NumberOfParts}\relax\else
2307     \ifx\eqeWrtExamTitleToSolns\eqe@YES
2308       \ifnum\nNumberOfParts=1\relax
2309         \else
2310           \def\eqe@argi{#1}%
2311           \ifx\eqe@argi\@empty
2312             \eqe@writetoSolns{\eqexamargii}\eqe@writetoAux{\string
2313               \csarg\string\gdef{userFriendly\eqexamargii}{\eqexamargii}}
2314             \else
2315               \eqe@writetoSolns{#1}\eqe@writetoAux{\string
2316                 \csarg\string\gdef{userFriendly#2}{#1}}%
2317             \fi
2318           \fi
2319         \fi
2320       \exambegdef
2321       \edef\temp@Exp{\noexpand\shortquiz\sqstar[\eqexamargii]\temp@Exp
2322 }{%
2323   \examenddef
2324   \vskip\eqeques@parsep\relax\kern0pt %dps88
2325   \endshortquiz
2326   \aftergroup\afterexamsepcode

```

```

2327   \par\penalty-100\vskip0pt
2328   \ifDoNotRecordThisExam % dpsf02
2329     \expandafter\eIFFalseWrtSolns\fi
2330   \global\DoNotRecordThisExamfalse
2331 }

```

`\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}`.

```

2332 \newcommand{\EQEcalculateAllTotals}{%
2333   \begingroup

```

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

```

2334     \count\z@=0\relax
2335     \def\##1{\csarg\ifx{##1total}\relax\else
2336       \advance\count\z@\csname##1total\endcsname

```

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

```

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

```

```

2337     \fi}\csname thePartNames\endcsname
2338     \xdef\eqeGrandTotal{\the\count\z@}%
2339     \ifnum\eqeGrandTotal=0 \else

```

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

```

2340     \def\##1{\eqe@calc@percent{##1}}%
2341     \csname thePartNames\endcsname\fi
2342   \endgroup
2343 }

```

`\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.

```

2344 \newcommand{\nPctDecPts}{1}
2345 \def\eqe@calc@percent#1{\@ifundefined{#1total}{%
2346   \csarg\gdef{#1percent}{??}}{%

```

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

```

2347     \csarg\ifx{FPdiv}\relax
2348     \count2=\tot@lForPart{#1}%
2349     \edef\expGT{\csname eqeGrandTotal\endcsname}%
2350     \multiply\count2by100\relax\divide\count2by\expGT\relax
2351     \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.

```

2352     \FPdiv{\eqe@pForPart}{\csname#1total\endcsname}%
2353         {\csname eqeGrandTotal\endcsname}%
2354     \FPmul{\eqe@pForPart}{\eqe@pForPart}{100}%
2355     \FPround{\eqe@pForPart}{\eqe@pForPart}{\nPctDecPts}%
2356     \csarg\xdef{#1percent}{\eqe@pForPart}\fi
2357 }%
2358 }

2359 \def\writetotalstoaux{\eqe@IW0\@auxout{\string
2360     \csarg\string\gdef{\thisexamlabel total}{\theeqpointvalue}}%
2361     \eqe@IW0\@auxout{\string\csarg
2362     \string\gdef{\thisexamlabel nQuestions}{\theeqquestionnoi}}%
2363 }
2364 \newcommand{\writelastpage}[1][\def\eqe@argi{#1}%
2365     \ifx\eqe@argi\empty\else\label{#1PageEnd}\fi
2366     \eqe@IW0\@auxout{\string\csarg
2367     \string\gdef{eqExamLastPage}{\arabic{page}}}%
2368 }
2369 \def\exlabel{}
2370 \def\sqlabel{}
2371 \def\exsolafter{\textit{Solution}:}
2372 \def\sqlsolafter{\textit{Solution}:}

```

The exercise labels in the body of the text. (2015/02/27) changed the command `\exlabelformatwp` to incorporate other parameters.

```

2373 %\def\exlabelformat{\textbf{\theeqquestionnoi.\ }}
2374 \def\exlabelformat{\textbf{%
2375     \theeqquestionnoi\eqe@decPointPrb\eqe@hspannerPrb}}
2376 \def\exlabelformatwp{\exlabelformat}

```

The exercise labels for solutions at the end of the document

```

2377 \def\exsllabelformat
2378     {\string\makebox[0pt][r]{\string\textbf{%
2379     \theeqquestionnoi\eqe@decPointPrb\eqe@hspannerSoln}}}
2380 \def\exsllabelformatwp
2381     {\string\makebox[0pt][r]{\string\textbf{%
2382     \theeqquestionnoi\eqe@decPointPrb\eqe@hspannerSoln}}%
2383     (\thepartno)\eqe@hspannerSoln}
2384 \ifanswerkey
2385     \def\exrtnlabelformat{}
2386     \def\exrtnlabelformatwp{}
2387     \def\eq@sqlrtnlabel{}
2388 \else
2389     \def\exrtnlabelformat{\$\square\$}
2390     \def\exrtnlabelformatwp{\$\square\$}
2391     \def\eq@sqlrtnlabel{\$\square\$}
2392 \fi
2393 \def\sqlrtnlabel{\eq@sqlrtnlabel}

```

(2010/08/21) Enable some localizations of strings

```

2394 \newcommand{\exsectitletext}{Solutions to \webtitle}
2395 \def\exsectitle{\normalsize\exsectitletext}
2396 %\def\exsectitle{\normalsize\hspace*
2397 %    {-\oddsidemargin}\exsectitletext}
2398 \ifpackageloaded{exerquiz}{%\{Solutions to \websubject}
2399     \renewcommand{\exsecrunhead}{}}{\newcommand{\exsecrunhead}{}}
2400 %\providecommand{\exsecrunhead}{Solutions to \websubject}%
2401 \def\eq@sqlsectitle{}
2402 \def\eq@sqlsecrunhead{}
2403 \def\eq@sqllabel{\string\llap{\string\textbf{\thequestionnoi.\ }}}
2404 \def\eq@sqllabel{}
2405 \let\include@quizzesolutions\relax
2406 \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.

```

2407 \newcommand{\optsFillIn}[1]{\def\eqe@optsFillIn{#1}}
2408 \let\eqe@optsFillIn\@empty
2409 \newcommand{\fillin}[3][u]{%
2410     \ifx#1u\let\@fillinFmt\underbar
2411     \else\ifx#1b\let\@fillinFmt\relax
2412     \else\let\@fillinFmt\relax\fi\fi
2413     \ifeq@proofing
2414         \@fillinFmt{\makebox[#2]{%
2415             \strut\hfil\bfseries\color{red}#3\hfil}}%
2416     \else
2417         \@fillinFmt{\makebox[#2]{\strut\hfil}}%
2418         \ifundefined{@quiz}{}{%
2419             \if\eq@online\eqe@YES\relax
2420                 \ifeq@nosolutions
2421                     \ifeq@solutionsafter\else
2422                         \ifx\eq@insertverticalspace\eqe@YES\relax
2423                             \stepcounter{@cntfillin}%
2424                             \edef\fieldName{%
2425                                 \if\probstar*eqexam.\curr@quiz.fillin.%
2426                                 \thequestionnoi.part\thepartno.%
2427                                 fi\the@cntfillin%
2428                             \else
2429                                 eqexam.\curr@quiz.fillin.%
2430                                 \thequestionnoi.fi\the@cntfillin%
2431                             \fi
2432                             }\makebox[0pt][r]{\textField[\BC{}
2433                             \presets{\eqe@optsFillIn}}]{%

```

```

2434 \fieldName}{#2}{11bp}}%
2435 \fi
2436 \fi
2437 \fi
2438 \fi
2439 }%
2440 \fi\space\ignorespaces}

```

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

```

2441 \newcommand\defaultTFwidth{30pt}
2442 \newcommand\TF[2][\defaultTFwidth]{%
2443 \def\eqe@next{\fillin{#1}{#2}}%
2444 \ifdim\eq@extralabelsep=0pt\relax\else
2445 \if\probstar*\relax\if\exerwparts@cols0
2446 \def\eqe@next{\makebox[0pt][r]{%
2447 \fillin{#1}{#2}}\ignorespaces}%
2448 \fi\fi\fi
2449 \eqe@next}
2450 \def\fillinWidth#1{%
2451 \if\probstar*
2452 \settowidth{\eq@tmplengthA}{\normalfont\ }%
2453 \addtolength{\eq@tmplengthA}{#1}%
2454 \edef\eq@extralabelsep{\the\eq@tmplengthA}%
2455 \fi
2456 }
2457 \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.

```

2458 \def\eqe@DoWarning#1{\PackageWarning{eqexam}{\string#1\space
2459 requires the first argument\MessageBreak of problem* to be <num>ea}}
2460 \def\eqe@DoNum{\textbf{??}\eqe@DoWarning{\DoNum}}
2461 \def\eqe@nDoNum{\textbf{??}\eqe@DoWarning{\nDoNum}}
2462 \let\DoNum\eqe@DoNum
2463 \let\nDoNum\eqe@nDoNum
2464 \def\makeDoNum#1{\xdef\nDoNum{#1}%
2465 \def\ifc@sewrap{\ifcase#1??\or}%
2466 \xdef\DoNum{\expandafter\ifc@sewrap\eqe@wordNums\else
2467 \eqe@wordNumsError\fi}}
2468 \def\makeOutOfNum#1{\xdef\nOutOfNum{#1}%
2469 \def\ifc@sewrap{\ifcase#1??\or}%
2470 \xdef\nOutOfNum{\expandafter\ifc@sewrap\eqe@wordNums\else
2471 \eqe@wordNumsError\fi}}
2472 \def\eqe@OutOfNum{\textbf{??}\eqe@DoWarning{\nOutOfNum}}
2473 \def\eqe@nOutOfNum{\textbf{??}\eqe@DoWarning{\nOutOfNum}}
2474 \let\nOutOfNum\eqe@OutOfNum
2475 \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.

```

2476 \newcommand{\eqe@wordNums}{one\or two\or three\or
2477   four\or five\or six\or seven\or eight\or nine\or ten}
2478 \newcommand{\eqe@wordNumsError}{\noexpand\PackageError{eqexam}%
2479   {Number out of range, 1--10}%
2480   {Use a smaller number, or redefine the command
2481   \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`.

```

2482 \let\priorPNPAction\relax % dps1
2483 \let\postPNPAction\relax
2484 \def\eqe@mkRoomPgBrk{\priorPNPAction\newpage\postPNPAction}
2485 \def\makeRoomForProb#1#2{\par %\endgraf % dps 11/11/10
2486   \bgroup\@nobreakfalse\addpenalty{-500}%
2487   \setlength{\@tempdimb}{#1}%
2488   \@tempdima \pagegoal \advance \@tempdima -\pagetotal
2489   \ifdim \@tempdima<\@tempdimb\ifnum\col@number>\@ne\columnbreak
2490     \else\aftergroup\eqe@mkRoomPgBrk\fi\fi\egroup
2491   \ifnum\@reportpoints>1
2492     \ifx\eqe@prevProbZero\eqe@YES
2493       \if\eqe@isPts0\else\vskip-\halfHtPtBox\relax\fi
2494     \else
2495       \ifl@stDispl@yPoints
2496         \if\eqe@isPts0\@checkSpacing{1}\else
2497           \if#21 \@checkSpacing{0}\fi
2498         \fi\fi
2499     \fi
2500   \fi
2501 }

```

`\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`.

```

2502 \newcommand{\emitMessageNearBottom}{\@ifstar
2503   {\def\eqe@emb{\ifnum\col@number>\@ne\columnbreak
2504     \else\newpage\fi\insertContAnnot}\eq@emitMessageNearBottom}
2505   {\let\eqe@emb\relax\eq@emitMessageNearBottom}}
2506 \newcommand{\eq@emitMessageNearBottom}[2][2][w1Vspace]{%
2507   \eq@emitMessageNearBottom{#1}{#2}}
2508 \def\eq@@emitMessageNearBottom#1#2{\par
2509   \bgroup\@nobreakfalse\addpenalty{-500}%
2510   \setlength{\@tempdimb}{#1}%
2511   \@tempdima \pagegoal \advance \@tempdima -\pagetotal

```

```

2512 \ifdim \@tempdima<\@tempdimb #2 \eq@emmb
2513 \xdef\eq@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`.

```

2514 \newcommand{\getSpaceLeftOnPage}{\par\bgroup
2515 \@tempdima \pagegoal \advance \@tempdima -\pagetotal
2516 \xdef\amtSpaceLeftOnPage{\the\@tempdima}\egroup}

```

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

```

2517 \newcommand{\pnpDflt}{\@fvsizeskip\textheight}
2518 \newcommand{\promoteNewPage}[1][\pnpDflt]{%
2519 \makeRoomForProb{#1}{0}}

```

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

```

2520 \def\pointsmarginparpush{3pt}
2521 \def\@checkSpacing#1{\bgroup\ifinner\else
2522 \@tempdima\lastPageTotal
2523 \@tempdimb\pagetotal
2524 \ifdim\@tempdima < \@tempdimb
2525 \advance\@tempdimb by-\@tempdima
2526 \ifdim\@tempdimb < \eq@pointboxtotalheight
2527 \if#1%
2528 \eq@pointboxtotalheight\halfHtPtBox\relax
2529 \advance\eq@pointboxtotalheight\dpPtBox\relax
2530 \fi
2531 \@tempdima=\eq@pointboxtotalheight
2532 \advance\@tempdima\pointsmarginparpush\relax
2533 \advance\@tempdima by-\@tempdimb
2534 \vspace*{\@tempdima}%
2535 \fi
2536 \fi\fi
2537 \egroup}
2538 \def\@checkSpacingi{\bgroup
2539 \@tempdima = \lastPageTotal
2540 \@tempdimb = \pagetotal
2541 \ifdim\@tempdima < \@tempdimb
2542 \advance\@tempdimb by-\@tempdima
2543 \eq@pointboxtotalheight\halfHtPtBox\relax
2544 \advance\eq@pointboxtotalheight\dpPtBox\relax
2545 \ifdim\@tempdimb < \eq@pointboxtotalheight
2546 \@tempdima=\eq@pointboxtotalheight
2547 \advance\@tempdima\pointsmarginparpush\relax
2548 \advance\@tempdima by-\@tempdimb

```

```

2549          \vspace*{\@tempdima}%
2550          \fi
2551      \fi
2552 \egroup}
\default@nbaselineskip is the default number of \baselineskips needed to place a new problem. While
\baselineskip          \baselineskip is the number of \baselineskips needed for a new problem (or
                        problem*).
2553 \newcommand{\setDefaultnbaselineskip}[1]{\def\default@nbaselineskip{#1}}
2554 \setDefaultnbaselineskip{6}
2555 \def\nbaselineskip#1{\def\@nbaselineskip{#1}}
2556 \def\nbaselineskipReset{\edef\@nbaselineskip{\default@nbaselineskip}}
2557 \baselineskipReset

```

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

```

2558 \def\eqe@gobbletoend#1\end{}
2559 \def\eqe@grabarg#1\end{\def\numpoints{#1}}
2560 \def\eqe@isPts{1}

```

We reset some `exerquiz` parameters for `eqexam`

```

2561 \aboveeskip{3pt}\beloweskip{3pt}
2562 \partstopsep{3pt}\partsitegap{3pt}\partsparep{0pt}
2563 \rowsepDefault{3pt}\partstabcolsep{1.5pt}%
2564 \partstabtopsep{3pt}\partstabrowsep{3pt}
2565 \renewcommand\belowesolnskip{{}}%
2566 \let\isProbEnv\eqe@NO
2567 \let\topofprohook\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.

```

2568 \def\eqe@p@gobnxtprDef{\@ifnextchar\par{\eqe@p@gobtop@rnext}{}}
2569 \def\eqe@p@gobtop@rnext#1\par{}
2570 \def\applyparfixesp{\let\eqe@p@gobnxtpr\eqe@p@gobnxtprDef}
2571 %\def\cancelparfixesp{\let\eqe@p@gobnxtpr\relax}
2572 \def\cancelparfixesp{\let\eqe@p@gobnxtpr\ignorespaces}

```

The `problem` environment now begins.

```

2573 \newcommand{\problem}[1][\lowercase{\def\@rgi{#1}}%

```

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

```

2574     \ifx\@rgi\eqe@h
2575         \def\hidden@ttr{#1}\let\numpoints\@empty\else
2576         \let\hidden@ttr\relax\def\numpoints{#1}\fi
2577     \problem@cont}

```

`\probV@luesInsert` This command writes the number of points to the AUX file; form of the macro name written is `value<exam-label>Prob<prob-no>`.

```

2578 \let\probV@luesInsert\relax

```

```

2579 \def\probV@luesInsert{\bgroup\@tempcnta\value{eqquestionnoi}\advance
2580 \@tempcnta\@ne % Jan21
2581 \eq@IW0\@auxout{\string\csarg\string
2582 \gdef{value\thisexamlabel Prob\the\@tempcnta}{\numpoints}}\egroup
2583 }

```

`\trackProblemsOn` This command (`\trackProblemsOn`) activates problem tracking (ie, write problems and values to the AUX file; `\trackProblemsOff` turns off tracking. Tracking is off by default. When tracing is on, the value of each problem is saved in the form `\csarg\gdef{value<exam-label>Prob<prob-num>}`. This feature was developed to support the skills package.

```

2584 \def\trackProblemsOn{\let\probV@luesInsert\probV@luesInsert}
2585 \def\trackProblemsOff{\let\probV@luesInsert\relax}
2586 \trackProblemsOff

```

`\numPtsOfProblem[<exam-label>]{<prob-num>}` The command `\numPtsOfProblem` typesets the number of points of problem `<prob-num>` from part `<label-name>`.

```

2587 \newcommand{\numPtsOfProblem}[2][\thisexamlabel]%
2588 {\@nameuse{value#1Prob#2}}

```

The problem environment continues.

```

2589 \newcommand{\problem@cont}[1][\let\isProbEnv\eq@YES
2590 \def\@rgi{#1}\ifx\hidden@ttr\relax % assume a number or empty
2591 \ifx\@rgi\@empty\else\def\hidden@ttr{[#1]}\fi\fi
2592 \if\eq@isPts0\global\let\eq@prevProbZero\eq@YES\else
2593 \global\let\eq@prevProbZero\eq@NO\fi
2594 \global\thereissolutionfalse
2595 \def\numpointsEmpty{0}%
2596 \ifx\numpoints\@empty\def\numpoints{0}\def\numpointsEmpty{1}%
2597 \gdef\eq@isPts{0}\else\gdef\eq@isPts{1}\fi
2598 \makeRoomForProb{\@nbaselineskip\baselineskip}{\eq@isPts}%
2599 \gdef\probstar{x}\let\afterlabelhskip\@empty
2600 \ifx\marginpoints\@empty\else
2601 \if\numpointsEmpty\eq@One\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.

```

2602 \def\@lti{\let\@isitstar\eq@One\eq@grabarg}%
2603 \def\@ltii{\let\@isitstar\eq@Zero\eq@gobbletoend}%
2604 \expandafter\@ifstar\expandafter\@lti
2605 \expandafter\@ltii\numpoints\end
2606 \probV@luesInsert % Jan21
2607 \if\@isitstar\eq@One\addtocounter{eqpointvalue}{\numpoints}%
2608 \@markttotalvalue
2609 \ifdisplayPoints
2610 \def\marginparafterhook{\PTs{\numpoints}\space}\else
2611 \def\marginparafterhook{\PTs*{\numpoints}\space}\fi\else
2612 \def\marginparpriorhook{\noindent
2613 \probvalue{\numpoints}{0}}\fi
2614 \fi
2615 \setcounter{eq@count}{\value{eqquestionnoi}}%

```

```

2616 \addtocounter{eq@count}{1}%
2617 \ifnum\value{eq@count}=1\relax
2618 \eqe@IW0\@auxout{\string
2619 \csarg\string\gdef{pagenofirstprob\thisexamlabel}{\thepage}}%
2620 \fi

```

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

```

2621 \topofprobhook
2622 \begin{equestions}%

```

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

```

2623 \edef\ctrl@exp@exercise{\noexpand
2624 \begin{exercise}[eququestionnoi]\hidden@ttr}%
2625 \ctrl@exp@exercise\ignorespaces\eqe@p@gobnxtpr
2626 }

```

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

```

2627 \def\endproblem{\end{exercise}}%
2628 \end{equestions}%
2629 \ifeqlocalversion\ifeqglobalversion
2630 \xdef\eqe@tmp{\noexpand\forVersion{\eq@selectedVersion}}%
2631 \aftergroup\eqe@tmp
2632 \fi\fi
2633 \global\eqlocalversionfalse
2634 \ifdispl@yPoints\global\l@stDispl@yPointstrue
2635 \else\global\l@stDispl@yPointsfalse\fi
2636 \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.

```

2637 \newcommand\itemPTsTxt[1]{\eqe@ptsFmt{#1}\,\text{%
2638 \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi}$}
2639 \newcommand\itemPTsEaTxt[1]{\eqe@ptsFmt{#1}\,\text{%
2640 \ifnum#1=1\relax\eqptLabel\else\eqptsLabel\fi\space\eqeachLabel}$}
2641 \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.

```

2642 \newif\ifObeyPTsStar \ObeyPTsStartrue
2643 \def\obeyPTsStar{\global\ObeyPTsStartrue}
2644 \def\ignorePTsStar{\global\ObeyPTsStarfalse}
2645 \def\PTs{\leavevmode@ifstar{\@PTs*}}{\@PTs{x}}
2646 \def\@PTs#1#2{%
2647     \if\@reportpoints\eqe@Zero\ignorespaces\else
2648         \if\eqe@pointsPartsId\eqe@One
2649             \addtocounter{eqpointvalue}{#2}\@marktotalvalue
2650             \addtocounter{eq@count}{#2}\fi

```

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

```

2651     \if#1*\ifObeyPTsStar\ignorespaces\else\expandafter\@gobble\fi
2652 \else
(2012/04/26) Wrapped \itemPTsFormatted{\itemPTsTxt{#2}} as the argument
of \eqe@movePTs. \eqe@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 \eqe@movePTs follows.
2653     \eqe@movePTs{\itemPTsFormatted{\itemPTsTxt{#2}}}\fi
2654 \fi
2655 }
2656 \def\eqe@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.

```

2657 \let\auto\relax
2658 \def\isItD@{\@ifnextchar\Do{\let\yest@D@\eqe@YES\y@st@Do}
2659     {\let\yest@D@\eqe@NO\@gobblelet@end}}
2660 \let\yest@D@\eqe@NO
2661 \let\D@Num@\empty
2662 \def\y@st@Do\Do#1\end{\gdef\D@Num{#1}}
2663 \def\@gobblelet@end#1\end{}
Test the argument to see if it is the \auto token
2664 \def\is@uto#1\auto#2\@nil{\def\eqe@argi{#1}\def\eqe@argii{#2}%
2665     \ifx\eqe@argii\empty\let\isit@uto\eqe@Zero\else
2666         \let\isit@uto\eqe@One\fi
2667 }

```

`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).

```

2668 \let\eqe@pointsPartsId\relax

```

```

2669 \def\leadinitemWarningStar{\PackageWarning{eqexam}
2670   {Using the star (*) in front of the points\MessageBreak
2671   designator is not allowed when there is a
2672   \MessageBreak\string\leadinitem}}
2673 \def\tableadinWarningStar{\PackageWarning{eqexam}
2674   {Using the star (*) in front of the points\MessageBreak
2675   designator is not allowed when there is a
2676   \MessageBreak\string\tableadin}}
2677 \let\isProbStarEnv\eqe@NO
2678 \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`

```

2679 \def\eqe@ps@gobtop@r#1\par{\eqe@isle@dinnext}
2680 \def\eqe@isle@dinnext{\@ifnextchar\leadinitem{\vskip-\baselineskip}}
2681 \def\restorele@dinprefixDef{\@ifnextchar\par{\eqe@ps@gobtop@r}{}}
2682 \let\eqe@ps@gobnxtpar\restorele@dinprefixDef
2683 \def\applyleadinfix{\let\eqe@gobnxtpar\restorele@dinprefixDef}
2684 \def\canceledinfix{\let\eqe@gobnxtpar\relax}
2685 \def\applyparfixes{\PackageInfo{eqexam}
2686   {Applying paragraph fixes to problem\MessageBreak
2687   and problem* environments}\applyparfixesp\applyleadinfix}
2688 \def\cancelparfixes{\PackageInfo{eqexam}
2689   {Cancelling paragraph fixes to problem\MessageBreak
2690   and problem* environments}\cancelparfixesp
2691   \canceledinfix}

```

The default behavior is to do nothing about blank lines.

```

2692 \cancelparfixesp\canceledinfix
2693 \let\nutralizeparfixes\cancelparfixes

```

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

```

2694 \csarg\def{problem*}{\let\isProbEnv\eqe@YES
2695   \let\isProbStarEnv\eqe@YES
2696   \global\thereissolutionfalse
2697   \@ifnextchar[{\pr@bl@m@star}{\pr@bl@m@star[]}}
2698 \def\pr@bl@m@star[#1]{%
2699   \@ifnextchar[{\pr@blem@star{#1}}{\pr@blem@star{#1}[]}}
2700 \def\pr@blem@star#1[#2]{%
2701   \if\eqe@isPts0\global\let\eqe@prevProbZero\eqe@YES\else
2702     \global\let\eqe@prevProbZero\eqe@NO\fi
2703   \def\numpoints{#1}\ifx\numpoints\@empty
2704     \gdef\eqe@isPts{0}\else\gdef\eqe@isPts{1}\fi
2705   \makeRoomForProb{\@nbaselineskip\baselineskip}{\eqe@isPts}%

```

Let the `\Do` commands to internal versions

```

2706   \let\DoNum\eqe@DoNum\let\nDoNum\eqe@nDoNum
2707   \let\OutOfNum\eqe@OutOfNum\let\nOutOfNum\eqe@nOutOfNum

```

```

2708 % \proofingsymbol{\ding{52}}%
2709 \gdef\probstar{*}%
2710 \gdef\pr@b@secondarg{#2}\setcounter{eq@count}{0}%
2711 \let\afterlabelhskip\@empty
2712 \global\let\probpointseach\@empty\def\numpoints{#1}%

Determine if the argument begins with *
2713 \@ifstar{\let\@isitstar\eqe@One\eqe@grabarg}%
2714 { \let\@isitstar\eqe@Zero\eqe@gobbletoend}#1\end

If this problem* environment has a \leadinitem, the * option is not allowed
2715 \if\@isitstar\eqe@One
2716 \@tempcnta=\thequestionnoi\relax
2717 \advance\@tempcnta1\relax
2718 \@ifundefined{leadinitem\thisexamlabel-\the\@tempcnta}{ }
2719 { \leadinitemWarningStar\let\@isitstar\eqe@Zero}%
2720 \@ifundefined{tableadin\thisexamlabel-\the\@tempcnta}{ }
2721 { \tableadinWarningStar\let\@isitstar\eqe@Zero}%
2722 \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.
2723 \expandafter\is@uto\numpoints\auto\@nil
2724 \if\isit@uto\eqe@One\let\eqe@pointsPartsId\eqe@One
2725 \global\let\probpointseach\relax

The author has requested \auto
2726 \def\eqe@next{\autocalcparts}%
2727 \else
2728 \ifx\pr@b@secondarg\@empty\else
2729 \let\eqe@pointsPartsId\eqe@Four
2730 \isItD@#2\end\fi

Not \auto so either <num>ea or <num>
2731 \def\eqe@next{\manualcalcparts{\numpoints}}%
2732 \fi\eqe@next

2012/11/30 added \leadinitem defined eqexam.def as a dummy command, which
we the \let here to \eqe@leadinitem.
2733 \let\leadinitem\eqe@leadinitem

2015/05/31 added \tableadin defined eqexam.def as a dummy command, which
we the \let here to \eqe@leadinitem.
2734 \let\tableadin\eqe@tableadin

\topofprobstarhook is a general purpose hook at the top of the problem* environment, before the
beginning of the nested eqequations environment.
2735 \topofprobstarhook
2736 \ifeqfortextbook
2737 \writeT@SolnFile{\protect\global
2738 \protect\frstProbNumShownfalse}\fi

```



```

2739 \begin{equestions}%
2740 \begin{exercise}[equestionnoi]*\eq@gobnxtpar}%
2741 \def\ftb@endprobstarCks{%
2742 \ifWithinANSGrp
2743 \PackageError{eqexam}{\string\bGrpANS\space is still open}
2744 {You need to match it with an \string\bGrpANS,
2745 or remove it.}%
2746 \fi
2747 }

```

\endproblem* begins here.

```

2748 \csarg\def\endproblem*}{%
2749 \eq@IW0\@auxout{\string\csarg\string
2750 \gdef{nPartsThisProb\thisexamlabel.\thequestionnoi}%
2751 {\arabic{partno}}}%
2752 \ifx\probpointseach\@empty\else
2753 \ifx\probpointseach\auto
2754 \eq@IW0\@auxout{\string\csarg\string
2755 \gdef{prob\thisexamlabel.\thequestionnoi}%
2756 {\theeq@count}}%
2757 \else
2758 \setcounter{eq@count}{\value{partno}}%
2759 \ifx\pr@b@secondarg\@empty\else
2760 \bgroup\toks0=\expandafter{\pr@b@secondarg}%
2761 \expandafter\isItD@\the\toks0 \end
2762 \ifx\yest@D@\eq@YES\ifx\eq@pointsEach\eq@YES
2763 \eq@IW0\@auxout{\string
2764 \csarg\string\gdef{DoNumThisProb\thisexamlabel.%
2765 \thequestionnoi}{\D@Num}}%
2766 \@tempcnta = \value{eq@count}%
2767 \advance\@tempcnta -\D@Num
2768 \global\advance\value{eq@count}-\@tempcnta
2769 % 3.0k
2770 \@tempcnta=\value{partno}%
2771 \advance\@tempcnta -\D@Num
2772 \multiply\@tempcnta by\argi
2773 \addtocounter{eqpointvalue}{-\@tempcnta}%
2774 \fi\fi
2775 \egroup
2776 \fi
2777 \multiply\value{eq@count}\argi
2778 \eq@IW0\@auxout{\string\csarg\string
2779 \gdef{prob\thisexamlabel.\thequestionnoi}{\theeq@count}}%
2780 \fi\fi
2781 \end{exercise}%
2782 \end{equestions}%
2783 \ifeqfortextbook\ftb@endprobstarCks\fi
2784 \ifeqlocalversion\ifeqglobalversion

```

```

2785     \xdef\eqe@tmp{\noexpand\forVersion{\eq@selectedVersion}}%
2786     \aftergroup\eqe@tmp
2787     \fi\fi
2788     \global\eqlocalversionfalse

2789     \@markttotalvalue
2790     \ifdispl@yPoints\global\l@stDispl@yPointstrue
2791     \else\global\l@stDispl@yPointsfalse\fi
2792 }

```

`\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  $4x^2 \geq (x+1)^2$  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\eq@leadinitemparts` 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`.

```
2793 \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.

```
2794 \def\eqe@leadinitem{%
```

```
2795     \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 exception.

```
2796         \def\eqe@next{\eqe@@leadinitem}%
```

```
2797         \ifeqfortextbook\writeT@SolnFile{\protect\bpartsmrk}\fi
```

```
2798     \else
```

```
2799         \def\eqe@next{\PackageError{eqexam}
```

```
2800         {The \string\leadinitem\space command may
```

```
2801         only be used\MessageBreak once per problem* environment}
```

```
2802         {Remove all but one of the \string\leadinitem\space
```

```
2803         commands.}}%
```

```
2804     \fi\eqe@next
```

```
2805 }
```

```
2806 \newif\ifisleadin \isleadinfalse
```

```
2807 \let\isparshapeExpanded\eqe@NO
```

```
2808 \def\eqe@@leadinitem{\if\itsforleadinitem\eqe@NO
```

```
2809     \setcounter{partno}{0}\fi\refstepcounter{partno}}%
```

After initializing the counter, we make various calculations

```
2810     \eqe@initializeContAnnot
```

```
2811     \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.

```
2812     \eqe@writetoAux{\string\csarg\string
```

```
2813     \gdef{leadinitem\thisexamlabel-\thequestionnoi}{}}%
```

```
2814     \settowidth{\eq@tmpdima}{\normalfont\parts@indent\eqe@prtsepPrb}%
```

```
2815     \xdef\leadinIndentPrtSep{\the\eq@tmpdima}%
```

```
2816     \setlength{\eq@tmpdima}{\eqemargin+\eq@tmpdima}%
```

```
2817     \xdef\leadinIndent{\the\eq@tmpdima}%
```

```
2818     \settowidth{\eq@tmplength}{\parts@indent}%
```

```
2819     \edef\partsleadinIndent{\the\eq@tmplength}%
```

```
2820     \setlength{\eq@tmplength}{\linewidth-\leadinIndentPrtSep}%
```

```
2821     \edef\leadinIndentLength{\the\eq@tmplength}%
```

Modify `\@listii` from core LaTeX: Increase `\leftmarginii` and `\linewidth` appropriately.

```
2822 \@ifundefined{@listii@SAVE}{\global\let\@listii@SAVE\@listii}{}%
```

```
2823     \expandafter\def\expandafter\@listii\expandafter{\@listii@SAVE
```

```

2824     \advance\leftmarginii\leadinIndentPrtSep\relax
2825     \leftmargin\leftmarginii \labelwidth\leftmarginii
2826     \advance\labelwidth-\labelsep
2827 %     \advance\linewidth\leadinIndentPrtSep\relax
2828     }%

```

The parshape for the first paragraph of the \leadinitem.

```

2829     \parshape=2 \eqmargin \linewidth \leadinIndent \leadinIndentLength

```

Here is the definition of \solutionparshape, we use the shape of subsequent paragraphs.

```

2830     \xdef\solutionparshape{%\noexpand\linewidth\leadinIndentLength
2831     \noexpand\parshape=1 \leadinIndent\space\leadinIndentLength}%

```

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

```

2832     \global\let\isparshapeExpanded\eqe@NO
2833     \everypar{\solutionparshape
2834     \global\let\isparshapeExpanded\eqe@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.

```

2835     \let\eq@item\eq@leadin@item\eqp@rtc@lcm@rk
2836     \def\currhideopt{x}\eq@hidesolutionfalse\eq@nolinkfalse
2837     \@ifnextchar[{\@ckhide}{\eq@item}%
2838 }

```

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

```

2839 \def\eq@leadin@item{\eq@ckglobalhide\ifeq@hidesolution\else
2840     \global\let\eqExerSolnHeader\eq@@writeexheaderlist\fi
2841     \ifeq@nosolutions\eq@nolinktrue\fi
2842     \ifeq@solutionsafter\eq@nolinktrue\fi
2843     \unskip\noindent\makebox[\partsleadinIndent]{\eqexlisttabheader}%
2844     \eqe@prtsepPrb\ignorespaces
2845 }

```

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

```

2846 \def\eqe@tableadin{\@tableadinitemtrue\eqe@writetoAux{\protect
2847     \csarg\protect\gdef{tableadin\thisexamlabel-\theequationnoi}{}}%
2848 }

```

\pushProblem There may be an occasion when a multi-part question needs to be broken between
\popProblem 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  $c$  make the function  $f$ 
continuous at  $x=2$ ?
\begin{solution}[.5in]\end{solution}
\end{parts}

```

2849 `\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.

```

2850 \newcommand\popProblem[1] [] {%
2851     \def\@argi{#1}\ifx\@argi\@empty
2852     \def\eqe@bParts{\begin{parts}}\else
2853     \def\eqe@bParts{\begin{parts}[#1]}\fi
2854     \eqe@bParts\setcounter{partno}{\nlastItem}}

```

2855 `\def\lastPageTotal{0pt}`

2856 `\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`.

2857 `\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>`.

2858 `\let\eqe@pointsEach\eqe@NO`

2859 `\def\manualcalcparts#1{%`

`\expandafter\prob@Arg#1ea\end`

2861 `\ifx\argii\@empty\edef\numpoints{#1}%`

2862 `\let\eqe@pointsEach\eqe@NO`

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

2863 `\let\eqe@pointsPartsId\eqe@Zero`

2864 `\else`

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

2865 `\gdef\probpointseach{x}\let\eqe@pointsPartsId\eqe@Two`

2866 `\let\eqe@pointsEach\eqe@YES`

Points each specified

2867 `\setcounter{eq@count}{\value{equationnoi}}%`

2868 `\addtocounter{eq@count}{1}\csarg`

```

2869     \ifx{prob\thisexamlabel.\theeq@count}\relax
2870     \def\numpoints{\argi}\else
2871     \def\numpoints{\expandafter
2872         \csname prob\thisexamlabel.\theeq@count\endcsname}%
2873     \fi

2874     \ifx\yest@D@\eqe@YES
2875     \setcounter{eq@count}{\value{eqquestionnoi}}%
2876     \addtocounter{eq@count}{1}%
2877     \@ifundefined{nPartsThisProb\thisexamlabel.\theeq@count}
2878     {\makeOutOfNum{0}\makeDoNum{0}}{%
2879     \expandafter\makeOutOfNum{%
2880     \csname nPartsThisProb\thisexamlabel.%
2881     \theeq@count\endcsname}%
2882     \expandafter\makeDoNum{%
2883     \csname DoNumThisProb\thisexamlabel.%
2884     \theeq@count\endcsname}}%
2885     \fi
2886     \fi

```

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

```

2887     \ifx\marginpoints\@empty
    No points for this exam

2888     \else
2889     \ifx\argi\@empty
    If \argi is empty, no points were specified, so we \let \marginpoints to \@empty
2890     \let\marginpoints\@empty
2891     \else
2892     \probV@luesInsert % Jan21
    Points are displayed in margins or inline
2893     \ifx\argii\@empty

```

Total points specified

```

2894     \ifx\marginpoints\@empty\else
2895     \probV@luesInsert % Jan21
2896     \if\@isitstar\eqe@0ne
    Points to appear “in-line” rather than in the margins
2897     \addtocounter{eqpointvalue}{#1}%
2898     \@markttotalvalue\ifdisplyPoints
2899     \def\marginparafterhook{\itemPTsFormatted{%
2900     \itemPTsTxt{\numpoints}}\space}\fi
2901     \else
    Points appear in the margins
2902     \def\marginparpriorhook{\noindent
2903     \probvalue{\numpoints}{0}}%
2904     \fi

```

```

2905             \fi
2906             \else

```

Points each specified

```

2907             \ifx\marginpoints\@empty\else
2908             \if\@isitstar\eqe@0ne\ifdisplyPoints
Points to appear “in-line” rather than in the margins
2909                 \def\marginparafterhook{%
2910                 \itemPTsFormatted{\itemPTsEaTxt{\argi}}\space}\fi
2911             \else
Points appear in the margins
2912                 \ifdisplyPoints
2913                 \def\marginparpriorhook{\noindent
2914                 \marginpoints{\numpoints}{\argi}}\fi
2915             \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`.

```

2916             \edef\eqp@rtc@lcm@rk{noexpand
2917             \addtocounter{eqpointvalue}{\argi}%
2918             \noexpand\@marktotalvalue}%
2919             \fi
2920         \fi
2921     \fi
2922 \fi
2923 \ifnum\value{eq@count}=1\relax
2924     \eqe@IW0\@auxout{\string\csarg\string
2925     \gdef{pagenofirstprob\thisexamlabel}{\thepage}}%
2926 \fi
2927 }

```

`\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`.

```

2928 \def\acp@mpah{\itemPTsFormatted{\itemPTsTxt{\numpoints}}\space}
2929 \def\acp@mpph{\noindent\marginpoints{\numpoints}{0}}
2930 \def\autocalcparts{%
2931     \setcounter{eq@count}{\value{eqquestionnoi}}%
2932     \addtocounter{eq@count}{1}%
2933     \csarg\ifx{prob\thisexamlabel.\theeq@count}\relax
2934     \def\numpoints{0}% assume zero points until we get the total
2935     \else
2936     \edef\numpoints{\@nameuse{prob\thisexamlabel.\theeq@count}}%
2937     \fi
2938 %     \addtocounter{eqpointvalue}{\numpoints}\@marktotalvalue
2939     \ifx\marginpoints\@empty\else
2940     \probV@luesInsert % Jan21

```

```

2941     \ifdisplyPoints
2942     \if\@isitstar\eqe@One
    If we have *\auto, the total is to appear inline.
2943         \def\marginparafterhook{\acp@mpah}\else
    Otherwise, the total will appear in the margin.
2944         \def\marginparpriorhook{\acp@mpph}\fi
2945     \fi
2946 \fi
2947 \setcounter{eq@count}{0}%
2948 }

```

`\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}`.

```

2949 \let\eqe@fpmrk\@empty
2950 \newcommand{\forproblem}[1]{\def\eqe@fpmrk{-\the@exno}%
2951     \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*}

```

```

2952 \newcommand{\foritem}[2][ ]{%
2953     \setcounter{partno}{0}\def\@rgi{#1}\ifx\@rgi\@empty
2954         \def\eqe@nextitem{\item}\else
2955         \def\eqe@nextitem{\item[#1]}\fi
2956     \foritem@cont{#2}}
2957 \newcommand{\forleadinitem}[2][ ]{%
2958     \setcounter{partno}{0}\def\@rgi{#1}\ifx\@rgi\@empty
2959         \def\eqe@nextitem{\leadinitem}\else
2960         \def\eqe@nextitem{\leadinitem[#1]}\fi
2961     \foritem@cont{#2}}

```

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

```

2962 \def\foritem@cont#1{\ifuseNumForParts
2963     \edef\fliPartNo{#1}\setcounter{partno}{#1-1}\else
2964     \lowercase{\def\eqe@selectedItem{#1}}%
2965     \let\eqe@initLoop\eqe@NO

```



```

2966 \loop
2967     \stepcounter{partno}\expandafter
2968     \if\alph{partno}\eq@selectedItem
2969         \let\eq@initLoop\eqe@YES\fi
2970 \ifx\eq@initLoop\eqe@NO\repeat
2971 \edef\fliPartNo{\the\c@partno}%
2972 \addtocounter{partno}{-1}\fi\let\itsforleadinitem\eqe@YES
2973 \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.

```

2974 \newcommand{\annotContStr}{%
2975     \textbf{Problem~{\eqeCurrProb} continued\strut}}
2976 \newcommand{\acvspace}[1]{\def\ic@vspace{#1}\setlength\iacvspace{#1}}

```

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

```

2977 \newlength\iacvspace
2978 \newcommand{\ic@vspacedefault}{1in} % changed .25in to 1in 2012/12/04
2979 \newcommand{\resetacvspace}{\setlength\iacvspace{\ic@vspacedefault}}
2980 \resetacvspace

```

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

```

2981 \def\@nnotContStrSkip{\vskip3pt}
2982 \newcommand{\eqe@insertContAnnot}[1][\iacvspace]{\promoteNewPage[#1]%
2983     \ifnum\arabic{page}>\eqe@currProbStartPage
2984     \xdef\eqe@currProbStartPage{\arabic{page}}%
2985     {\settowidth\eqe@tmplength{\parts@indent\eqe@prtsepPrb}%
2986     \xdef\eqe@partsIndent{\the\eqe@tmplength}}%
2987     \ifwithinparts

```

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

```

2988     \@ifundefined{eq@item@latex}{\item[]\hspace*{-\eqemargin}}%
2989     \ifx\solutionparshape\@empty\else\hspace*{-\eqe@partsIndent}\fi}
2990     {\eq@item@latex[]\hspace*{-\eqemargin}}%
2991     \hspace{-\eqe@partsIndent}}\else

```

If placed between problem sets, we indent as appropriate.

```

2992     \hspace*{-\eqemargin}\fi
2993     \annotContStr\@nnotContStrSkip % dpsj20
2994 \fi
2995 }

```

```
2996 \edef\eq@currProbStartPage{\arabic{page}}
```

`\turnContAnnotOff` Turn off and on this feature. The default is off.

```
\turnContAnnotOn 2997 \newif\ifcont@not \cont@notfalse
2998 %\newcommand{\@gobbloptone}[1] [] {}
2999 \newcommand{\turnContAnnotOff}{\global\cont@notfalse
3000   \global\let\eq@insertContAnnot\relax}
3001 \newcommand{\turnContAnnotOn}{\global\cont@nottrue
3002   \global\let\eq@insertContAnnot\eqe@insertContAnnot}
3003 \turnContAnnotOff
```

Used for manually inserting annot

```
3004 \def\insertContAnnot{\eq@insertContAnnot}
```

`\aNNewPage` A simple command for inserting `\newpage`, only if the `\answerkey` option has been taken.

`\qNewPage`

```
3005 \newcommand\aNNewPage{\ifanswerkey\newpage\fi}
3006 \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.

```
3007 \newcounter{backofpage}
3008 \newcommand\bopText{on the back of page~\boPage}
3009 \newcommand\bopCoverPageText{on the back of the cover page}
3010 \newcommand\OnBackOfPage[1] [\bopText]{%
3011   \refstepcounter{backofpage}\label{bop\thebackofpage}
3012   \begingroup
3013   \csarg\ifx{r@bop\thebackofpage}\relax
3014     \def\boPage{??}\else
3015     \edef\eqe@temp{\csname r@bop\thebackofpage\endcsname}%
3016     \ifx\hyper@anchor@undefined
3017       \edef\boPage{\expandafter\@secondoftwo\eqe@temp}\else
3018       \edef\boPage{\expandafter\@secondoffive\eqe@temp}\fi
3019     \c@eq@count\boPage
3020     \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.

```
3021   \edef\boPage
3022   {%
3023     \ifx\eqex@coverpage\relax
3024       \ifnum\value{eq@count}=0
3025         1%
3026       \else
3027         \the\value{eq@count}%
3028       \fi
3029     \else
```

```

3030             \ifnum\value{eq@count}=0
3031                 -1%
3032             \else
3033                 \the\value{eq@count}%
3034             \fi
3035         \fi
3036     }%
3037 \fi
3038 \csarg\ifx{r@bop\thebackofpage}\relax#1\else
3039 \ifnum\boPage=-1\relax\bopCoverPageText\else#1\fi\fi
3040 \endgroup
3041 }

```

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).

```

3042 \newcommand{\useFillerLines}{\ifx\vspaceFiller % dpsj8
3043 \vspaceFillerLines\else\@eqlinedfillertrue
3044 \let\vspaceFiller\vspaceFillerLines\fillTypeDefault
3045 \fi}

```

`\useFillerDefault` Resets the vertical space to the original white space.

```

3046 \newcommand{\useFillerDefault}{\@eqlinedfillerfalse
3047 \let\vspaceFiller\vspaceFillerDefault
3048 \let\eqWriteLine\hfill}

```

`\fillTypeHRule` Writes the line as a solid line (`\hrulefill`).

```

3049 \newcommand{\fillTypeHRule}{\let\eqWriteLine\eqWriteLineFill
3050 \let\makeVgrid\relax}

```

`\fillTypeDots` Writes the line as a dotted line (`\dotfill`).

```

3051 \newcommand{\fillTypeDots}{\let\eqWriteLine\eqWriteLineDots
3052 \let\makeVgrid\relax}

```

`\fillTypeDashLine` Writes the line as a dotted line (`\eqdashrulefill`).

```

3053 \newcommand{\fillTypeDashLine}{\let\eqWriteLine\eqWriteLineDashFill
3054 \let\makeVgrid\relax}

```

`\fillTypeBlankLine` Fills the line

```

3055 \newcommand{\fillTypeBlankLine}{\let\eqWriteLine\eqWriteLineBlankFill
3056 \let\makeVgrid\relax}

```

`\fillTypeDefault` Resets fill type back to the default, `\hrulefill`.
3057 `\newcommand{\fillTypeDefault}{\let\eqWriteLine\eqWriteLineFill`
3058 `\let\makeVgrid\relax}`

`\fillTypeGrid` Fills the space with a grid, horizontal and vertical lines.
3059 `\newcommand{\fillTypeGrid}{\ifx\vspaceFiller\vspaceFillerLines`
3060 `\let\eqWriteLine\eqWriteLineFill\let\makeVgrid\eqe@makeVgrid\fi}`

`\eqWriteLineColor` The color of the rule to use.
3061 `\newcommand{\eqWriteLineColor}[1]{\def\eq@WriteLineColor{#1}}`
3062 `\eqWriteLineColor{gray}`

`\eqWLSpacing` The line spacing between the rules.
3063 `\newlength\wlVspace`
3064 `\newcommand{\eqWLSpacing}[1]{\setlength\wlVspace{#1}}`
3065 `\eqWLSpacing{14.0pt}`

Fill Types. We have three types of line fill: `\hrulefill`, `\dotfill`, and a custom rule `\eqdashrulefill`. The commands three `\eqWriteLineFill`, `\eqWriteLineDots`, and `\eqWriteLineDashFill` implements these three types. They are `\let` to `\eqWriteLine`, which is used in `\vspaceFillerLines`.

```

3066 \newcommand{\eqWriteLineFill}{%
3067 \textcolor{\eq@WriteLineColor}{\leaders
3068 \hrule height \flfboxrule\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 `flexextended` and `answerkey` are in force, and when `\useFillerLines` and `\turnflanskeyOn` have been expanded.

```

3069 \newbox\eqe@tempbox
3070 \newbox\eqe@nskeyflsplit
3071 \newbox\eqe@nskeyfltop
3072 \def\eqdotrulefill{\leavevmode
3073 \cleaders\hb@xt@ .44em{\copy\eqe@tempbox\hss}\hfill} % dps16
3074 % \cleaders\hb@xt@ .44em{\unhcopy\eqe@tempbox\hss}\hfill} % dps8
3075 \newcommand{\eqWriteLineDots}{%
3076 \textcolor{\eq@WriteLineColor}{\eqdotrulefill}}
3077 \def\eqdashrulefill{\leavevmode
3078 \cleaders\hb@xt@ .44em{\rule{.22em}{\flfboxrule}\hss}\hfill\kern\z@}
3079 \newcommand{\eqWriteLineDashFill}{%
3080 \textcolor{\eq@WriteLineColor}{\eqdashrulefill}}
3081 %\newcommand{\eqWriteLineBlankFill}{\hfill}
3082 \newcommand{\eqWriteLineBlankFill}{\vphantom{\hrulefill}}

```

Vertical counterparts to those above, excepting blank fill.

```

3083 \newcommand{\eqWriteLineVFill}{\leaders\vrule width\flfboxrule\vfill}
3084 \def\eqdotruleVfill{\cleaders\vbox to .44em{\vss
3085 \hbox to0pt{\hss.\hss}}\vfill}
3086 \newcommand{\eqWriteLineVDots}{\eqdotruleVfill}

```

```

3087 \def\eqdashruleVfill{\cleaders\vbox to .44em{\vss
3088 \hbox to0pt{\hss\rule{\flfboxrule}{.22em}\hss}}\vfill}
3089 \newcommand{\eqWriteLineDashVFill}{\eqdashruleVfill}

```

When the solution environment contains both an `nLines` and a dimension, 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`.

```

3090 \newif\if@equsedim \@equsedimtrue
3091 \newcommand{\useVspaceDimen}{\@equsedimtrue}
3092 \newcommand{\usenLineDimen}{\@equsedimfalse}
3093 \newif\if@eqalignfilllinestoleft\@eqalignfilllinestoleftfalse

```

`\fillerLinesAlignDef` Use `\fillerLinesAlignDef` and `\fillerLinesOnLeftMargin` to adjust the alignment of the rule lines for this feature. The first is the default, the second one aligns rule lines to the left margin.

```

3094 \newcommand{\fillerLinesOnLeftMargin}{\@eqalignfilllinestolefttrue}
3095 \newcommand{\fillerLinesAlignDef}{\@eqalignfilllinestoleftfalse}
3096 \newcommand\priorPageBreakMsg[1]{\def\priorP@geBre@kMsg{#1}}
3097 \let\priorP@geBre@kMsg\empty
3098 \newcommand{\flPageBreakMsg}[1]{%
3099 \priorPageBreakMsg{\emitMessageNearBottom[\iacvspace]
3100 {\eqfititin{\Large\strut}#1}}}%
3101 }

```

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.

```

3102 \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.

```

3103 \define@choicekey+{eqefillLines}{numbers}{\val\nr}%
3104 {none,left,right}[none]{\ifcase\nr\relax
3105 \let\eqe@numLinesL\relax\let\eqe@numLinesR\relax\or
3106 \let\eqe@numLinesL\eqe@numLinesL\let\eqe@numLinesR\relax\or
3107 \let\eqe@numLinesL\relax\let\eqe@numLinesR\eqe@numLinesR
3108 \fi
3109 }{\PackageWarning{aeb}{Bad choice for numbers, permissible values
3110 are none, left, and right. Try again}}
3111 \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.

```

3112 \define@key{eqefillLines}{numbersep}[2pt]{%
3113   \setlength{\eqetmplengtha}{#1}%
3114   \edef\eqe@numbersep{\the\eqetmplengtha}}
3115 \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.
3116 \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.
3117 \define@boolkey{eqefillLines}{topline}[true]{ }
3118 \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.
3119 \let\eqe@usedeffboxrule\eqe@NO % dps20
3120 \let\gridtypeseselected\@empty % dps26
3121 \define@choicekey+{eqefillLines}{gridtype}[\val\nr]%
3122   {line,dash,dots}[line]{\edef\gridtypeseselected{\val}% dps26
3123   \ifcase\nr\relax
3124     \let\gridHLineFill\eqWriteLineFill
3125     \let\gridVLineFill\eqWriteLineVFill
3126   \or
3127     \let\gridHLineFill\eqWriteLineDashFill
3128     \let\gridVLineFill\eqWriteLineDashVFill
3129   \or
3130     \let\gridHLineFill\eqWriteLineDots
3131     \let\gridVLineFill\eqWriteLineVDots
3132     \let\eqe@usedeffboxrule\eqe@YES %dps20
3133   \fi
3134 }{\PackageWarning{aeb}{Bad choice for numbers, permissible values
3135   are line, dash, and dots. Try again}}
3136 \let\gridHLineFill\eqWriteLineFill
3137 \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.
3138 \define@choicekey+{eqefillLines}{fltype}[\val\nr]%
3139   {line,dash,dots,blank,grid}[line]{\ifcase\nr\relax
3140     \fillTypeHRule\or
3141     \fillTypeDashLine\or
3142     \fillTypeDots
3143     \let\eqe@usedeffboxrule\eqe@YES\or % dps20
3144     \fillTypeBlankLine\or
3145     \fillTypeGrid\fi
3146 }{\PackageWarning{aeb}{Bad choice for numbers, permissible values
3147   are line, dash, dots, blank, and grid. Try again}}

align=(left|default) Another convenience key, align simply executes the command

```

versions.

```
3148 \define@choicekey+{eqefillLines}{align}[\val\nr]%
3149   {default,left}[default]{\ifcase\nr\relax
3150     \fillerLinesAlignDef\or
3151     \fillerLinesOnLeftMargin\fi
3152 }{\PackageWarning{aeb}{Bad choice for numbers, permissible values
3153   are line, dash, dots, blank, and grid. Try again}}
3154 \define@choicekey*{eqefillLines}{equalcells}[\val\nr]%
3155   {true,false}[true]{\ifcase\nr\relax
3156     \equalCellSizesOn\or
3157     \equalCellSizesOff\fi
3158 }
```

`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.

```
3159 \define@boolkey{eqefillLines}{outlineonly}[true]{}
3160 \KV@eqefillLines@outlineonlyfalse
3161 \define@choicekey{eqefillLines}{outlineonly*}{true,false}[true]{% dps26
3162   \@nameuse{KV@eqefillLines@outlineonly#1}%
3163   \def\eqefillLines@outlineonlystar{#1}}%
3164 \let\eqefillLines@outlineonlystar\@empty
```

`bgonly*=(true|false)` colors the work area with a color of your choice, see `bgcolor` below.

```
3165 \define@boolkey{eqefillLines}{bgonly}[true]{} % dps11
3166 \KV@eqefillLines@bgonlyfalse
3167 \define@choicekey{eqefillLines}{bgonly*}{true,false}[true]{% dps26
3168   \@nameuse{KV@eqefillLines@bgonly#1}%
3169   \def\eqefillLines@bgonlystar{#1}}%
3170 \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.

```
3171 \define@key{eqefillLines}{bgcolor}[]{\def\eqe@BGColor{#1}} % dps11
3172 \let\eqe@BGColor\@empty
```

`\fillLinesNumFmt{<cmds>}` is used to specify the format for line numbers. Within the `<cmds>` argument, use `#1` to indicate the placement of the number. For example, the code `\fillLinesNumFmt{\textbf{#1}}` puts the line numbers in bold font.

```
3173 \def\fillLinesNumFmt#1{\def\eqe@fillLinesNumFmt##1{\def\flnum{##1}#1}}
3174 \fillLinesNumFmt{\flnum} % set to default values
3175 \define@key{eqefillLines}{numfmt}[]{\fillLinesNumFmt{#1}} % dps21
3176 \define@key{eqefillLines}{linegap}[14pt]{\def\fl@wlspace{#1}} % dps21
3177 \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`.

```
3178 \newcommand{\setFillLinesFmt}[1]{\def\@rgi{#1}}%
```

```

3179 \let\eqe@usedeffboxrule\eqe@NO %dps20
3180 \let\gridtypeseselected\@empty
3181 \ifx\@rgi\@empty
3182   \setkeys{eqefillLines}{numbers,numbersep,color,
3183     gridtype,fltype,align}%
3184 \else
3185   \let\is@outlineonly\eqe@NO\let\is@bgonly\eqe@NO
3186   \ifKV@eqefillLines@outlineonly\let\is@outlineonly\eqe@YES\fi
3187   \ifKV@eqefillLines@bgonly\let\is@bgonly\eqe@YES\fi
3188   \setkeys{eqefillLines}{#1}%
3189   \eqWLSpacing{\fl@wlspace}%
3190   \ifx\makeVgrid\relax
3191     \KV@eqefillLines@toplinefalse
3192 %   \forceEqualCellsfalse % dps21
3193     \equalCellSizesOff % dpsj15
3194     \ifx\is@bgonly\eqe@YES % carryover from prev prob
3195       \ifKV@eqefillLines@bgonly
3196         \KV@eqefillLines@bgonlyfalse
3197       \else
3198         \PackageWarning{eqexam}{You must specify fltype=grid for
3199           bgonly to work.\MessageBreak
3200           Ignoring this key for now}%
3201         \KV@eqefillLines@bgonlyfalse
3202         \let\is@bgonly\eqe@NO
3203       \fi
3204     \else % not continuation
3205       \ifKV@eqefillLines@bgonly
3206         \PackageWarning{eqexam}{You must specify fltype=grid for
3207           bgonly to work.\MessageBreak
3208           Ignoring this key for now}%
3209         \KV@eqefillLines@bgonlyfalse
3210         \let\is@bgonly\eqe@NO
3211       \fi
3212     \fi
3213 % same for outline
3214     \ifx\is@outlineonly\eqe@YES % carryover from prev prob
3215       \ifKV@eqefillLines@outlineonly
3216         \KV@eqefillLines@outlineonlyfalse
3217       \else % dps18
3218         \PackageWarning{eqexam}{You must specify fltype=grid for
3219           outlineonly to work.\MessageBreak
3220           Ignoring this key for now}%
3221         \KV@eqefillLines@outlineonlyfalse
3222         \let\is@outlineonly\eqe@NO
3223       \fi
3224     \else % not continuation
3225       \ifKV@eqefillLines@outlineonly
3226         \PackageWarning{eqexam}{You must specify fltype=grid for
3227           outlineonly to work.\MessageBreak
3228           Ignoring this key for now}%

```



```

3229         \KV@eqefillLines@outlineonlyfalse
3230         \let\is@outlineonly\eqe@NO
3231     \fi
3232 \fi
3233 \let\gridHLineFill\eqWriteLine
3234 \let\gridVLineFill\eqWriteLine
3235 \else
3236 \ifx\gridtypeselcted@empty % dps26
3237 \let\gridHLineFill\eqWriteLineFill
3238 \let\gridVLineFill\eqWriteLineVFill
3239 \fi
3240 \fi
3241 \fi
3242 }

```

The next two definitions place line numbers on the left and right, respectively.

```

3243 \def\eqe@numLinesL{\ifKV@eqefillLines@topline\ifnum\@tempcntb>0 % dps5
3244 \smash{\llap{\tiny
3245 \eqe@fillLinesNumFmt{\the\@tempcntb}\hglue\eqe@numbersep\relax}}\fi
3246 \else\smash{\llap{\tiny
3247 \eqe@fillLinesNumFmt{\the\@tempcntb}\hglue\eqe@numbersep\relax}}\fi}
3248 \def\eqe@numLinesR{\smash{\rlap{\tiny
3249 \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.

```

3250 \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`.

```

3251 \newif\ifgridpgrk\gridpgrkfalse % 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).

```

3252 \newif\ifwriteVertic@lFLines \writeVertic@lFLinestrue % dps16
3253 \def\fl@set@nnotContStrSkip{\def\@nnotContStrSkip{\vskip6pt}} % dpsj23
3254 \let\p@ssToFLs\relax
3255 \newif\iffl@firstpass % dps27
3256 \newcommand{\vspaceFillerLines}[1]{\begingroup\offinterlineskip %dps1
3257 \global\fl@firstpasstrue % dps27
3258 \fl@set@nnotContStrSkip % dpsj20
3259 \p@ssToFLs\if@eqlinedfiller
3260 \def\eqe@next{\vspaceFillerLines@i{#1}}\else
3261 \expandafter\def\expandafter\eqe@next
3262 \expandafter{\expandafter\endgroup\p@ssToFLs}\fi
3263 \global\let\p@ssToFLs\relax\eqe@next}
3264 \def\vspaceFillerLines@i#1{\advance\wlvspace-\flfboxrule % dps25
3265 \ifx\eqe@usedeffboxrule\eqe@YES\flfboxrule=.4pt\fi % dps20

```

```

3266 \ifKV@eqefillLines@outlineonly\KV@eqefillLines@toplinetrue %dps10
3267 \KV@eqefillLines@bgonlyfalse % dps11
3268 % \forceEqualCellsfalse % dpsj15
3269 \equalCellSizesOff\let\makeVgrid\eqe@makeVgrid\fi %dps10
3270 \ifKV@eqefillLines@bgonly\KV@eqefillLines@toplinetrue
3271 \KV@eqefillLines@outlineonlyfalse\flfboxrule=.4pt % dps20
3272 % \forceEqualCellsfalse % dpsj15
3273 \equalCellSizesOff\let\makeVgrid\eqe@makeVgrid\fi %dps11
3274 \parindentOpt\relax\parskipOpt
3275 \@tempdima\w1Vspace\eqetmplengthaOpt
3276 \@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.
3277 \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.

3278 \eqe@wrtLineKernal
3279 \setbox\eqe@tempbox\hbox{\hb@xt@0pt{\hss.\hss}}\ht\eqe@tempbox.33pt
3280 \dp\eqe@tempboxOpt % dps16
3281 \ifx\makeVgrid\relax
3282 % \forceEqualCellsfalse % dpsj15
3283 \equalCellSizesOff
3284 \else % dps2
3285 \let\eqWriteLine\gridHLineFill
3286 \forceEq@lCells % dps2
3287 \ifx\flEq@lLineWidth\@empty\else
3288 \linewidth=\flEq@lLineWidth\relax\fi
3289 \ifKV@eqefillLines@bgonly\else
3290 \ifKV@eqefillLines@topline\bgroup\@tempcntb=0% dps5
3291 \makebox[Opt][l]{\eqe@x
3292 \makebox[\eqe@fillwidth]{\eqWriteLine}}\egroup
3293 \fi\fi % dps5
3294 \fi
3295 \settowidth{\eqetmplengthb}{\eqe@decPointPrb}\def\vfGo@l{#1}%
3296 \ifx\makeVgrid\relax\else
3297 \def\priorPNPAction{\xdef\fl@nRows{\the\@tempcnta}% dps27
3298 \if@ealignfilllinestoleft\eqe@x\fi % dpsj12
3299 \makeVgrid\priorP@geBre@kMsg\global\fl@firstpassfalse}% dps4
3300 \def\postPNPAction{\global\gridpgbrktrue % dps27
3301 \setlength{\@tempdima}
3302 {(\vfGo@l+\w1Vspace)-\depthtoday}% dps25
3303 \edef\vfGo@l{\the\@tempdima}\@tempdima=\w1Vspace\relax
3304 \@tempcnta=0\relax}% dps18
3305 \fi
3306 \@whiledim\@tempdima<\vfGo@l\relax\do
3307 {\edef\depthtoday{\the\@tempdima}%
3308 \ifx\eqe@insert@more@content\relax\ifx\makeVgrid\relax

```

```

3309     \priorP@geBre@kMsg\fi\fi %dps15
3310     \eq@insertContAnnot\vskip\wVspace\eqe@x
    Draw the horizontal line: a rule, dotted line, dashed line
3311     \ifKV@eqefillLines@bgonly\vglue\flfboxrule\else % dpsj11 dpsj21
3312     \ifKV@eqefillLines@outlineonly\vglue\flfboxrule\else %dps10
3313         \makebox[0pt][l]{\eqe@numLinesL
3314         \makebox[\eqe@fillwidth]{\eqWriteLine}\eqe@numLinesR}\fi
3315     \fi
3316     \advance\@tempcnta\relax\advance\@tempcntb1\relax
3317     \edef\depthtodate{\the\@tempdima}%
3318     \addtolength{\@tempdima}{\wVspace}%\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.

```

3319 } \xdef\fl@nRows{\the\@tempcnta}\makeVgrid % dps27
3320 \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.

```

3321     \ifvoid\eqe@nskeyflsplit\else\PackageWarning{eqexam}
3322     {Some material from a solution environment does not\MessageBreak
3323     appear. Increase space allotted}{\normalcolor
3324     \llap{\smash{\rule{5pt}{5pt}}}}\aftergroup
3325     \reset@color\fi % dpsj24
3326 \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 $\langle dimen \rangle$ the dimension specified.

```

3327 \setlength{\@tempdima}{\vfGo@l-\depthtodate}\vskip\@tempdima
3328 % \vskip\flfboxrule\vskip3pt
3329 \dimen0=\sameVspace\relax
3330 \dimen2=\fboxrule\advance\dimen0by\dimen2
3331 \xdef\sameVspace{\the\dimen0 }\endgroup
3332 \global\flfrstsplittrue
3333 }

```

`\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`.

```

3334 \def\eq@linesXPgs{\@tempdimb\wVspace % dpsj20
3335 \advance\@tempdimb-\flfboxrule
3336 \@tempdimb=\soln@keys@nLines\@tempdimb
3337 \advance\@tempdimb\flfboxrule
3338 \expandafter\vspaceFillerLines\expandafter{\the\@tempdimb}}

```

`\eqe@wrtLineKernal`, based on a case analysis, sets the register `\eqetmplengha` and makes some horizontal glue adjustments as well.

```

3339 %\def\eq@linesXPgs{\@tempdimb\wLVspace
3340 % \@tempdimb=\soln@keys@nLines\@tempdimb
3341 % \expandafter\vspaceFillerLines\expandafter{\the\@tempdimb}}
3342 \def\eqe@wrtLineKernal{\let\eqe@x\relax\let\eqe@y\relax % dpsj9
3343 \ifx\solutionparshape\@empty % not lead-in
3344 \if@eqalignfilllinestoleft % align on left
3345 \setlength{\eqetmplengtha}{\labelwidth+\eqemargin}%
3346 \setlength{\eqetmplengtha}
3347 {\eqetmplengtha+\eqetmplengthb}%
3348 \ifwithinparts
3349 \ifx\istabularexer\eq@YES % tabular mode
3350 \setlength{\eqetmplengtha}{\prtsIndntSep}% dps27
3351 \eq@tmplengthA\eqetmplengtha
3352 \def\eqe@x{\hglue-\prtsIndntSep}% dps27
3353 \ifKV@eqefillLines@outlineonly\let\eqe@y\eqe@x\fi
3354 \else
3355 \settowidth{\eqetmplengtha}{\eqe@prtsepPrb}%
3356 \setlength{\eqetmplengtha}{\eqetmplengtha+\widthOfParts}% dps9
3357 \setlength{\eq@tmplengthA}{\eqetmplengtha}% dps9
3358 \def\eqe@x{\hspace*{-\eqetmplengtha}\hglue-\flfboxrule}% dps27
3359 \fi
3360 \else % not parts, still align left
3361 \def\eqe@x{\hspace*{-\eqemargin}}%
3362 \eqetmplengtha\eqemargin
3363 \eq@tmplengthA\eqemargin
3364 \fi
3365 \else % not lead-in, not align on left
3366 \ifwithinparts
3367 \setlength{\eqetmplengtha}{\labelwidth}%
3368 \setlength{\eqetmplengtha}
3369 {\eqetmplengtha+\eqetmplengthb}%
3370 \ifx\istabularexer\eq@YES % tabular mode
3371 \eqetmplengthaOpt
3372 \eq@tmplengthA\eqetmplengtha % dps27
3373 \else %dps2
3374 \def\eqe@x{\hglue-\flfboxrule}% dps4
3375 \eqetmplengthaOpt
3376 \eq@tmplengthA\eqetmplengtha
3377 \fi
3378 \else % not parts, not align left
3379 \eqetmplengthaOpt
3380 \eq@tmplengthA\eqetmplengtha
3381 \fi
3382 \fi
3383 \else % lead-in
3384 \if@eqalignfilllinestoleft
3385 \setlength{\eqetmplengtha}{\leadinIndentPrtSep}% dps27
3386 \eq@tmplengthA\eqetmplengtha
3387 \else
3388 \eqetmplengthaOpt

```

```

3389     \eq@tmplengthA\eqetmplengtha
3390     \fi
3391 \fi
3392 }

```

`\equalCellSizesOn` set the switch `\ifforceEqualCells` to true, while `\equalCellSizesOff` sets the same switch to false. Additionally, the latter command resets `\fillLinesLineWidth` to its default value of `\@empty`. It is therefore important not to use the switch directly, but to use these two convenience commands.

```

3393 \newif\ifforceEqualCells
3394 \def\equalCellSizesOn{\global\let\flEqu@lLineWidth\@empty
3395     \global\forceEqualCellstrue}
3396 \def\equalCellSizesOff{\global\let\flEqu@lLineWidth\@empty
3397     \global\forceEqualCellsfalse}
3398 \equalCellSizesOff

```

`\forceEqu@lCells` determines the value of `\fillLinesLineWidth`. It successively adds `\w1Vspace` until it exceeds `\linewidth`, it then subtracts off one `\w1Vspace` to get the value for `\fillLinesLineWidth`.

```

3399 \def\forceEqu@lCells{\bgroup
3400     \ifforceEqualCells
3401     \dimen6\w1Vspace % reduced from \vspaceFillerLines
3402     \advance\dimen6\flfboxrule
3403     \dimen2=\z@
3404     \dimen4\linewidth
3405     \advance\dimen4by-\flfboxrule
3406     \if@eqlalignfilllinestoleft % dps8
3407     \advance\dimen4\eq@tmplengthA
3408     \fi
3409     \@tempcnta0
3410     \@whiledim \dimen2 < \dimen4 \do{%
3411         \advance\@tempcnta1
3412         \advance\dimen2by\dimen6
3413     }%
3414     \advance\@tempcnta-1
3415     \advance\dimen2-\dimen6
3416     \advance\dimen2\flfboxrule
3417     \advance\dimen2by-\eq@tmplengthA
3418     \xdef\flEqu@lLineWidth{\the\dimen2}%
3419     \fi
3420 \egroup}
3421 \let\eqe@insert@more@content\relax
3422 \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.

```
3423 \def\dbMrk#1{\textsf{\footnotesize#1}}
3424 \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 of the work area. Use `\resetFillerCustomBg` to reset the method of coloring the background to its default method.

```
3425 \def\fillerCustomBg#1{\def\fillerBgIm@ge##1##2{#1}}
3426 \def\resetFillerCustomBg{\let\fillerBgIm@ge\@empty}
3427 \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.

```
3428 \def\gridIndentAdj#1{\def\@rgi{#1}\ifx\@rgi\@empty
3429 \let\gridIndent@dj\relax\else % dps4
3430 \setlength{\@tempdima}{#1}\edef\gridIndent@dj{\the\@tempdima}\fi}
3431 \let\gridIndent@dj\relax
3432 \def\@eqe@makeVgrid{\begingroup
3433 \@tempcnta=\fl@nRows\relax
3434 \advance\wLVspace\flfboxrule % dps25
3435 \setlength{\@tempdima}{\@eqe@fillwidth}%
3436 \setlength{\@tempdimb}{\wLVspace}%
3437 \edef\@eqe@tmp{\strip@pt\@tempdimb}%
3438 \expandafter\gobtodot\@eqe@tmp.\@nil
3439 \divide\@tempdima by\intPrt\relax
3440 \dimen2=\@tempdima
3441 \dimen4=1pt
3442 \advance\dimen2 by 0.5\dimen4
3443 \divide\dimen2 by \dimen4
3444 \multiply\dimen2 by \dimen4
3445 \edef\@eqe@tmp{\strip@pt\dimen2}%
3446 \expandafter\gobtodot\@eqe@tmp.\@nil
3447 \advance\@tempcnta-1\relax % dps21
3448 \multiply\@tempdimb\@tempcnta
3449 \edef\@eqe@HOfVrule{\the\@tempdimb}% dps7
3450 \ifKV@eqefillLines@topline % dps21
3451 \advance\@tempdimb\wLVspace
3452 \advance\@tempdimb\flfboxrule
3453 \edef\@eqe@HOfVruleFrstRow{\the\@tempdimb}%
3454 \def\@eqe@setVRule{\vbox
```

```

3455         to\eqe@HOfVruleFrstRow\relax{\gridVLineFill}}% dps8
3456     \else
3457         \def\eqe@setVRule{\vbox
3458             to\eqe@HOfVrule\relax{\gridVLineFill}}% dps8
3459     \fi
3460     \eqe@tempcnta=0
3461     \dimen2=\wLVspace
3462     \count4=\intPrt\relax
3463     \setlength{\dimen4}{\eqe@fillwidth}%
3464     \multiply\dimen2by\count4
3465     \@whiledim \dimen2 > \dimen4 \do{%
3466         \advance\count4by-1\relax
3467         \edef\intPrt{\the\count4 }%
3468         \dimen2\wLVspace
3469         \multiply\dimen2by\count4
3470     }%
3471     \dimen2\wLVspace
3472     \if@eqalignfilllinestoleft % dps27
3473         \let\fl@oSmash\smash\let\fl@iSmash\relax
3474     \else
3475         \let\fl@iSmash\smash\let\fl@oSmash\relax
3476     \fi
3477     \fl@oSmash{%
3478     \makebox[0pt][l]{\makebox[\eqe@fillwidth][l]{\fl@iSmash}% dps27
3479         \ifx\gridIndent@dj\relax\else
3480             \hglue\gridIndent@dj\relax\fi
3481         \eqe@y\ifeqedb\rlap{\dbMrk{D}}\fi % dpsj9 \eqe@y (fudge)
3482         \rlap{\ifeqedb\rlap{\dbMrk{B}}\fi
3483         \ifwriteVertic@lFLines % dps16
3484         \setlength{\@tempdima}{\eqe@fillwidth}%
3485         \ifKV@eqefillLines@bgonly % dps11
3486             \ifx\fillerBgIm@ge\@empty
3487                 \ifx\eqe@BGColor\@empty\else
3488                     \rlap{\textcolor{\eqe@BGColor}
3489                         {\rule{\eqe@fillwidth}{\eqe@HOfVruleFrstRow}}}\fi
3490                 \else
3491                     \rlap{\fillerBgIm@ge{\eqe@fillwidth}{\eqe@HOfVruleFrstRow}}\fi
3492     \else
3493         \dimen2=\wLVspace
3494         \dimen0=0pt
3495         \@whilenum\eqe@tempcnta<\count4\do{\rlap{\hglue\dimen0
3496             \ifKV@eqefillLines@outlineonly %dps10
3497                 \ifnum\eqe@tempcnta=0 \else\let\gridVLineFill\vfill\fi\fi
3498             \textcolor{\eq@WriteLineColor}{\eqe@setVRule % dps27
3499                 \ifeqedb\rlap{\advance\eqe@tempcnta by1\relax\scriptsize
3500                     \raisebox{1pt}{\kern1pt\the\eqe@tempcnta}}\fi}}%
3501             \advance\dimen0by\dimen2
3502             \advance\eqe@tempcnta by1
3503         }% dps27
3504     \fi % dps11

```

```

3505 \advance\dimen2by\flfboxrule
3506 \ifKV@eqefillLines@bgonly\else % dps11
3507 \rlap{\hglue\@tempdima
3508 \textcolor{\eq@WriteLineColor}{\eqe@setVRule
3509 \ifeqedb\rlap{\thinspace\dbMrk{R}}\fi}}%
3510 \ifKV@eqefillLines@outlineonly %dps10
3511 \makebox[Opt][l]{\makebox[\eqe@fillwidth]{\eqWriteLine}}%dps10
3512 \fi
3513 \ifKV@eqefillLines@topline% dps7
3514 \ifgridpgrk\@tempdima\eqe@HOfVruleFrstRow\relax % dps21
3515 \raise\@tempdima\hbox{\makebox[Opt][l]{%
3516 \makebox[\eqe@fillwidth]{\eqWriteLine}}}%
3517 \global\gridpgrkfalse
3518 \fi
3519 \fi
3520 \fi
3521 \ifx\eqefillLines@outlineonlystar\@empty\else % dps26
3522 \global\let\eqefillLines@outlineonlystar\@empty
3523 \global\KV@eqefillLines@outlineonlyfalse
3524 \fi
3525 \ifx\eqefillLines@bgonlystar\@empty\else % dps26
3526 \global\let\eqefillLines@bgonlystar\@empty
3527 \global\KV@eqefillLines@bgonlyfalse
3528 \fi
3529 \fi % ifwriteVertic@lFLines
3530 \eqe@insert@more@content
3531 }% rlap
3532 }% smash (fl@iSmash)
3533 }% makeboxes and \fl@oSmash
3534 \endgroup
3535 }% \eqe@makeVgrid

```

In preparation for the `flextended` option, we define several commands and one environment that emit a package warning if the `flextended` option is not taken.

```

3536 \def\turnfl@nskeyMsg{\PackageWarning{eqexam}
3537 {You must first take the flextended\MessageBreak
3538 option for this command to have any\MessageBreak effect}}
3539 \def\turnfl@nskeyOn{\turnfl@nskeyMsg}
3540 \def\priorw@Msg{\PackageWarningNoLine{eqexam}
3541 {The priorworkarea environment does nothing\MessageBreak
3542 without the flextended option. All such content\MessageBreak in these
3543 environments are absorbed}\global\let\priorw@Msg\@empty}
3544 \newenvironment{priorworkarea}{\priorw@Msg
3545 \setbox\eqe@nskeyflsplit\vbox\bgroup}
3546 {\egroup\setbox\eqe@nskeyflsplit\box\voidb@x}
3547 \let\turnfl@nskeyOff\turnfl@nskeyOn
3548 \newif\ifflfrstsplit \flfrstsplittrue
3549 \newif\ifeqe@flnosolns \eqe@flnosolnsfalse
3550 \def\turnflnosolnsOn{\eqe@flnosolnstrue\turnfl@nskeyOn}
3551 \def\turnflnosolnsOff{\eqe@flnosolnsfalse\turnfl@nskeyOn}

```



```

3552 %\let\turnflnosolnsOn\turnflanskeyOn
3553 %\let\turnflnosolnOff\turnflanskeyOn
3554 \newdimen\flboxrule \flboxrule=.4pt
3555 \def\tweakBreakPoint#1{\def\@rgi{#1}\ifx\@rgi\@empty
3556   \gdef\twe@kBre@kPoint{Opt}\else
3557   {\setlength{\@tempdima}{#1}%
3558     \xdef\twe@kBre@kPoint{\the\@tempdima}}\fi}
3559 \def\twe@kBre@kPoint{Opt}

```

When `fextended` option is not loaded, we make a minimal definition of `\eq@b@ddCodeSpecial`. This redefinition is needed when `bgonly` is in effect.

```

3560 \def\eq@b@ddCodeSpecial#1{\if@eqlinedfiller
3561   \ifKV@eqefillLines@bgonly\leavevmode
3562     \vskip-\baselineskip\kern\lineskip\fi\fi#1}

```

`\flSeparateCutNames` For debug, or whatever reason, `\flSeparateCutNames` gives each flx cut file a distinct name.

```

3563 \def\flSeparateCutNames{\def\fl@CutName{flx\fl@Cnt-\jobname.cut}}
3564 </package>
3565 <*fextended>

```

14.2 Utility commands used with the `fextended` option

```

3566 \def\fls@vebaselinelineskip{% dps22
3567   \xdef\flbaselineskip{\the\baselineskip}}
3568 \def\fl@Cnt{0}
3569 \def\fl@CutName{flx-\jobname.cut}
3570 \def\flwriteexsol@fter{% dpsj23
3571   \ifx\exsolafter\@empty
3572     \immediate\write\verbatim@out{\string\ignorespaces}% dpsj23
3573   \else
3574     \immediate\write\verbatim@out{\string\exsolafter
3575       \string\space\string\ignorespaces}%
3576   \fi
3577 }
3578 \def\fl@getContent#1{\begingroup
3579   {\count0=\fl@Cnt\relax
3580     \advance\count0by1\relax
3581     \xdef\fl@Cnt{\the\count0 }}}%
3582 % \let\save@wlog\wlog\let\wlog\@gobble
3583 % \let\save@message\message\let\message\@gobble
3584 \global\let\verbatim@out\CommentStream
3585 \immediate\openout\verbatim@out=#1
3586 \ifwithinparts
3587   \ifx\istabularexer\eq@YES
3588     \immediate\write\verbatim@out{\string\hfuzz\string\hsize}%
3589     \immediate\write\verbatim@out{\string\minipage[t]{\string\hsize}}}%
3590   \ifanswerkey
3591     \immediate\write\verbatim@out{\string\expandafter
3592       \string\noindent\string\eqSolnExCmds}% dps23

```

```

3593     \flwriteexsol@fter
3594   \else
3595     \immediate\write\verbatim@out{\string\expandafter
3596       \string\noindent\string\priorWorkAreaCmds
3597       \string\ignorespaces}% dps23
3598   \fi
3599   \else
3600     \ifanswerkey\flwriteexsol@fter\fi
3601   \fi
3602 \else
3603   \ifanswerkey\flwriteexsol@fter\fi
3604 \fi
3605 \verbatimwrite
3606 }
3607 \def\fl@vsplitandplace#1{%
3608   \iff@firstpass
3609   \global\setbox\eq@nskeyflsplit\color@vbox
3610     \normalbaselines
3611     \predisplaypenalty=-50
3612     \postdisplaypenalty=-50
3613     \setlength{\hspace}{\eq@fillwidth}%
3614     \expandafter\noindent % dpsj23
3615     \ifanswerkey\expandafter\eqSolnExCmds\else
3616       \expandafter\priorWorkAreaCmds\fi
3617     \ignorespaces\input{#1}\fls@vebaselinelineskip
3618     \color@endbox\global\fl@firstpassfalse\fi
3619   \@tempdima\eq@HofVrule\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.

```

3620   \advance\@tempdima\flfboxrule
3621   \advance\@tempdima\lineskip
3622   \iff@firstsplit\else\advance\@tempdima\w@Vspace\fi
3623   \advance\@tempdima\maxdepth
3624   \advance\@tempdima by\twe@kBre@kPoint\relax
3625   \eq@tmplengthB\@tempdima % target height
3626   \splittopskip\w@Vspace
3627   \splitmaxdepth\maxdepth
3628   \advance\splitmaxdepth by\twe@kBre@kPoint\relax
3629   \setbox\eq@pointbox=\copy\eq@nskeyflsplit
3630   \setbox\eq@nskeyfltop=\vsplit\eq@pointbox to \@tempdima
3631   \setbox\eq@nskeyfltop=\vbox{\unvbox\eq@nskeyfltop}%
3632   \ifvoid\eq@pointbox\else
3633     \setlength{\@tempdima}
3634       {\eq@tmplengthB-\dp\eq@nskeyfltop}%
3635   \ifdim\@tempdima<0pt
3636     \@tempdima=-\@tempdima
3637   \ifdim\@tempdima>\maxdepth
3638     \advance\@tempdima-\flbaselineskip\relax

```

```

3639     \else
3640     \fi
3641   \else
3642   \setbox\eq@pointbox=\copy\eqe@nskeyflsplit
3643   \advance\@tempdima by\flbaselineskip\relax
3644   \setbox\eqe@nskeyfltop=\vsplit\eq@pointbox to \@tempdima
3645   \setbox\eqe@nskeyfltop=\vtop{\unvbox\eqe@nskeyfltop}%
3646   \setlength{\@tempdimb}{\eq@tmplengthB-\dp\eqe@nskeyfltop}%
3647   \ifdim\@tempdimb<0pt % too much
3648     \advance\@tempdima-\flbaselineskip\relax
3649   \else
3650   \fi
3651 \fi
3652 \fi
3653 \splittopskip\wVspace
3654 \lineskip1pt
3655 \setbox\eq@pointbox=\copy\eqe@nskeyflsplit
3656 \global\setbox\eqe@nskeyfltop=\vsplit\eqe@nskeyflsplit to \@tempdima
3657 \global\setbox\eqe@nskeyfltop\vtop{\unvbox\eqe@nskeyfltop\vskipOpt}%
3658 \ifvoid\eqe@nskeyflsplit
3659   \ifeqedb\llap{Bot}\fi
3660   \ifeqedb\raise\eqe@HOfVrule\llap{Top}\fi
3661   \ifeqedb\llap{V\qqquad}\fi
3662   \ifflrstsplit\global\flrstsplitfalse
3663   \ifeqedb\llap{FP\hskip.5in}\fi
3664   \rlap{\raisebox{\eqe@HOfVrule+\flfboxrule+\lineskip}
3665     {\vtop{\vsize=\eqe@HOfVrule\relax
3666       \unvbox\eqe@nskeyfltop\vfil}}}\else
3667   \ifeqedb\llap{SP\hskip.5in}\fi
3668   \rlap{\raisebox{\eqe@HOfVrule+\flfboxrule+\wVspace+\lineskip}
3669     {\vtop{\vsize=\eqe@HOfVrule\relax
3670       \unvbox\eq@pointbox\vfil}}}%
3671   \global\setbox\eqe@nskeyflsplit\box\voidb@x
3672 \fi
3673 \else
3674   \ifeqedb\llap{Bot}\fi
3675   \ifeqedb\raise\eqe@HOfVrule\llap{Top}\fi
3676   \ifeqedb\llap{N\qqquad}\fi
3677   \ifflrstsplit\global\flrstsplitfalse
3678   \ifeqedb\llap{FP\hskip.5in}\fi
3679   \rlap{\raisebox{\eqe@HOfVrule+\flfboxrule+\lineskip}
3680     {\vtop{\vsize=\eqe@HOfVrule\relax
3681       \unvbox\eqe@nskeyfltop\vfil}}}\else
3682   \ifeqedb\llap{SP\hskip.5in}\fi
3683   \rlap{\raisebox{\eqe@HOfVrule+\flfboxrule+\wVspace+\lineskip}
3684     {\vtop{\vsize=\eqe@HOfVrule\relax

```

To have the filler lines/grid with prior and solution content to space more than one page, we unbox `\eqe@nskeyfltop` but do not void it.

```

3685   \unvbox\eqe@nskeyfltop\vfil}}}%

```

```

3686 %      \unvbox\eq@pointbox\vfil}}}%
3687 %      \global\setbox\eq@nskeyflsplit\box\voidb@x
3688   \fi
3689 \fi
3690 \setbox\eq@pointbox\box\voidb@x
3691 % \gdef\twe@kBre@kPoint{0pt}% dpsj21
3692 }

```

14.3 Filler lines with answerkey option

When using the `fextended` 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.

```

3693 \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.

```

3694 \def\turnflnosolnsOn{\eq@flnosolnstrue}
3695 \def\turnflnosolnsOff{\eq@flnosolnsfalse}
3696 \def\turnflanskeyOn{%
3697   \def\turnfl@nskeyOnOff{%
3698     \ifanswerkey
3699       \ifx\makeVgrid\relax % already in grid mode, do nothing
3700       \ifwriteVertic@lFLines\KV@eq@fillLines@toplinefalse\fi % dps16
3701       \writeVertic@lFLinesfalse
3702       \let\makeVgrid\eq@makeVgrid % dpsj5
3703     \fi % dps17
3704     \let\eq@insert@more@content\eq@insertSolns % dpsj5
3705     \eq@fextendedtrue % dps16
3706     \fillTypeGrid % dps16
3707     \global\let\eq@insertverticalspace\eq@YES
3708     \global\vspacewithsolnstrue
3709   \fi
3710 }%
3711 }

```

`\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.

```

3712 \def\turnflanskeyOff{%
3713   \def\turnfl@nskeyOnOff{%
3714     \ifanswerkey
3715       \writeVertic@lFLinestrue
3716       \eq@fextendedfalse
3717       \let\eq@insertverticalspace\eq@NO
3718       \@eqlinedfillerfalse
3719       \global\let\eq@insert@more@content\relax
3720       \global\let\eq@insertverticalspace\eq@YES
3721       \ifvspacewithsolns\else
3722       \global\vspacewithsolnsfalse\fi

```

```

3723     \global\displayworkareafalse
3724     \fi
3725   }%
3726 }
3727 \turnflanskeyOff

3728 \let\p@ssToSolns\relax % dps30
3729 \def\eqe@insertSolns{\clubpenalty=50
3730   \widowpenalty=50 \vbadness=10000
3731   \fl@vsplitandplace{s\fl@CutName}}
3732 \def\eqe@b@ddCodeSpecial#1{\global\gridpgbrkfalse % dpsj15
3733   \global\let\eqe@e@ddCodeSpecial\relax
3734   \gdef\eqe@flnexti{#1}% dpsj5
3735   \if@eqlinedfiller
3736     \ifKV@eqefillLines@bgonly\leavevmode
3737       \vskip-\baselineskip\kern\lineskip\fi % dpsj11
3738     \ifKV@eqefillLines@outlineonly\leavevmode
3739       \vskip-\baselineskip\kern\lineskip\fi
3740     \ifanswerkey %\turnfl@nskeyOnOff % dps16
3741       \ifeqe@fextended
3742         \ifcont@nnot
3743           \ifx\makeVgrid\relax
3744             \gdef\p@ssToFLs{\turnfl@nskeyOnOff
3745               \let\eqe@insert@more@content\eqe@insertSolns
3746               \writeVertic@lFLinesfalse
3747               \KV@eqefillLines@toplinefalse
3748               \let\makeVgrid\eqe@makeVgrid
3749             }%
3750           \else
3751             \gdef\p@ssToFLs{\turnfl@nskeyOnOff
3752               \let\eqe@insert@more@content\eqe@insertSolns
3753               \writeVertic@lFLinestrue}%
3754           \fi
3755         \gdef\eqe@flnexti{\turnfl@nskeyOnOff
3756           \global\fl@firstpasstrue % dpsj5
3757           \p@ssToSolns
3758           \global\gridpgbrkfalse
3759           \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 `\eqe@ddCodeSpecial`.

```

3760     \xdef\kdvsp@SAVE{\ifkeepdeclaredvspacing
3761       true\else false\fi}%
3762     \gdef\kdvsp@Restore{@nameuse
3763       {keepdeclaredvspacing\kdvsp@SAVE}%
3764     \gdef\twe@kBre@kPoint{Opt}\global
3765     \let\kdvsp@Restore\relax}% dpsj21
3766     \expandafter\global\vspacewithkeyOff
3767     \vbadness=10000
3768     \fl@getcontent{s\fl@CutName}}%

```

```

3769         \gdef\eq@e@ddCodeSpecial{\endverbatimwrite\endgroup
3770 %         \kdvsp@Restore % dpsj20
3771         \ifwithinparts\ifx\istabularexer\eq@YES
3772         \immediate\write\verbatim@out{\string\endminipage}\fi\fi
3773         \immediate\closeout\verbatim@out
3774         \global\let\eq@e@ddCodeSpecial\relax
3775     }%
3776 \else
3777     \PackageWarningNoLine{eqexam}{For solutions to appear
3778     using filler lines\MessageBreak
3779     (with flextexded and answerkey options),\MessageBreak
3780     you must first expand \string\turnContAnnotOn\MessageBreak
3781     prior to line \the\inputlineno.\MessageBreak
3782     Switching to \string\useFillerDefault}%
3783     \useFillerDefault
3784     \let\eq@insertverticalspace\eq@N0
3785 \fi
3786 \fi
3787 \fi
3788 \fi
3789 \eqe@flnexti
3790 }% 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.

```

3791 \def\eqe@priorw@content{%
3792     \clubpenalty=50 % dps15
3793     \widowpenalty=50
3794     \vbadness=10000
3795     \fl@vsplitandplace{p\fl@CutName}%
3796 }%
3797 \renewenvironment{priorworkarea}{\par
3798 \ifx\solutionparshape\@empty\else % dps28
3799     \pushEnvir
3800     \everypar{}\if@e@alignfilllinestoleft\else
3801     \parshape=1 \leadinIndent \linewidth\fi
3802 \popEnvir
3803 \fi
3804 \global\let\p@ssToFLs\relax
3805 \global\let\p@ssToSolns\relax
3806 \vspacewithkeyOff
3807 \ifeq@nosolutions\else\sloppy\fi
3808 \def\eqe@flnexti{\eqSavedComment}%
3809 \ifeqe@flnosolns\ifdisplayworkarea

```

```

3810   \def\eqe@flnexti{\fl@getcontent{p\fl@CutName}}\fi\fi % dpsj21
3811   \gdef\eqe@flnextii{\endeqSavedComment}%dps11
3812   \ifeqe@flnosolns\ifdisplayworkarea
3813     \gdef\eqe@flnextii{\endverbatimwrite\endgroup
3814     \ifwithinparts\ifx\istabularexer\eq@YES
3815       \immediate\write\verbatim@out{\string\endminipage}\fi\fi
3816     \immediate\closeout\verbatim@out}\fi\fi
3817   \eqe@flnexti
3818 }{%
3819   \eqe@flnextii
3820   \ifcont@not
3821     \ifeqe@flnosolns
3822       \ifdisplayworkarea
3823         \ifx\eq@insertverticalspace\eqe@YES
3824           \ifx\makeVgrid\relax
3825             \gdef\p@ssToFLs{%
3826               \let\eqe@insert@more@content\eqe@priorw@content
3827               \writeVertic@lFLinesfalse
3828               \KV@eqefillLines@toplinefalse
3829               \let\makeVgrid\eqe@makeVgrid}%
3830           \else
3831             \gdef\p@ssToFLs{%
3832               \let\eqe@insert@more@content\eqe@priorw@content
3833               \writeVertic@lFLinestrue}%
3834           \fi
3835           \gdef\kdvsp@Restore{\gdef\twe@kBre@kPoint{Opt}\global
3836             \let\kdvsp@Restore\relax}% dpsj21
3837         \fi
3838       \fi
3839     \else
3840       \global\setbox\eqe@nskeyflsplit\box\voidb@x
3841       \global\setbox\eqe@nskeyfltop\box\voidb@x
3842     \fi
3843   \else
3844     \PackageWarningNoLine{eqexam}{For priorworkarea to appear
3845     using filler lines\MessageBreak
3846     (with flextexded and nosolutions options),\MessageBreak
3847     you must first expand \string\turnContAnnotOn\MessageBreak
3848     prior to line \the\inputlineno.\MessageBreak
3849     Switching to \string\useFillerDefault}%
3850     \gdef\p@ssToSolns{\useFillerDefault}%
3851   \fi
3852 }

3853 % end of segment
3854 </flextended>
3855 <*package>

Set the eqexam page style, if not otherwise indicated.
3856 \ifeqfortextbook\else
3857   \if@bypasseqexamheading\else

```

15 Insertion of figures into a problem or solution

Over the years, there have been many requests for “easy” methods of inserting images (or other content) into a problem. In this section, we provide some basic tools for doing just that.

There are two methods for inserting content (without disturbing the layout of the document): (1) using a minipage; and (2) using a wrap-figure package, such as `picins` (recommended).

The demo file for these features is `ins-fig.tex`

15.1 Enclose problem in a minipage

The idea is to include the `problem` environment in a `minipage` with a width smaller than `\linewidth`; then place the figure in another `minipage` next to the question. To use this methods, precede the enclosing `minipage` with the command `\probInMinipage`; illustration follows.

```
\def\PROB{The problem ...}
\def\FIG{\includegraphics[width=4.5cm]{figura1}}
\probInMinipage
\begin{minipage}{\linewidth}
\begin{minipage}[t]{\textwidth-4.5cm-11pt}\kern0pt
\begin{problem}[10]\PROB
\begin{solution}[1in]
This is the solution to ‘‘\PROB’’
\end{solution}
\end{problem}
\end{minipage}
\begin{minipage}[t]{4.5cm}\kern0pt\FIG
\end{minipage}\end{minipage}
```

The same technique is used for multi-part problems.

```
\probInMinipage
\begin{minipage}[t]{\linewidth-4.5cm-11pt}\kern0pt\parindent15pt
\item \PROB
\begin{solution}[1in]
This is the solution to ‘‘\PROB’’
\end{solution}
\begin{workarea}{\sameVspace}
\end{minipage}\hfill
\begin{minipage}[t]{4.5cm}\kern0pt\FIG\end{minipage}
```

`\probInMinipage` Place in front of a `minipage` environment that encloses side-by-side content: the problem on the left and the figure on the right. The big problem here is inserting the figure without disturbing the right margin totals. The `\eqTWSave` is the saved

value of the `\textwidth` in effect when the `exam` environment is opened. The right totals box is placed in the right margin based on this value.

```

3859 \def\probInMinipage{%
3860   \ifanswerkey
3861     \ifkeepdeclaredvspacing
3862       \global\let\w@sKeyOn\eqe@YES
3863       \vspacewithkeyOff
3864     \else
3865       \global\let\w@sKeyOn\eqe@NO
3866     \fi
3867   \fi
3868   \let\probInMinip@ge\eqTWSave\noindent}
3869 \let\probInMinip@ge\relax

```

`vadjForSolnInBx{width}` When the `problem` environment is enclosed in a `minipage`, the space left for the solution is not correct when `\vspacewithkeyOn` is in effect.

This is companion environment to the `\probInMinipage` command whose intention is to get the spacing correct. It is placed after all `minipages` are closed that enclosed the whole `problem`. The argument, `<width>`, is the same width of `minipage` that encloses the problem. The content of the environment should be the same content as the `solution` environment. `vadjForSolnInBx` calculates the total height of the typeset solution. It then calculates the difference between `\sameVspace`, which should hold the value of the optional argument of `solution`, and the actual height of the solution. It then executes a `\vspace` for that amount calculated.

```

3870 \newenvironment{vadjForSolnInBx}[1]{%
3871   \setlength\@tempdima{#1}%
3872   \setbox\z@\vbox\bgroup\hsize\@tempdima
3873 }{%
3874   \egroup
3875   \ifanswerkey
3876     \ifx\w@sKeyOn\eqe@YES
3877       \global\vspacewithkeyOn\fi
3878     \setlength{\@tempdima}{\ht\z@+\dp\z@}%
3879     \setlength{\@tempdima}{\sameVspace-\@tempdima}%
3880     \par\vspace{\@tempdima}\fi
3881     \setbox\z@\box\voidb@x
3882 }

```

15.2 Insertions using a wrapping package

A technique for wrapping text around the figure has been developed, and is an ugly hack. The example below is typical: define `\insertFig` (for convenience). Prior to the opening of `problem`, expand `\bProbInsert{\insertFig}`.

`\bProbInsert{fig}` Use this command above the `problem` (or `problem*`) environment only. Prior to opening the `problem` use `\bProbInsert` with its argument. Place `\eProbInsert` `\eProbInsert` at the end of the statement of the question, as illustrated below.

`\bProbInsert` puts the question into a group, `\eProbInsert` gets us out of the group.

```
\def\insertFig{\parpic[r]{\FIG}}
\bProbInsert{\insertFig}
\begin{problem}[10]\PROB
\PROB\space\PROB\eProbInsert
\begin{solution}[1in]
This is the solution: \PROB
\end{solution}
\end{problem}
```

```
3883 \def\bProbInsert#1{\def\@rgi{#1}\ifx\@rgi\@empty
3884 \global\let\insE@rlyAtQues\@empty\else
3885 \gdef\insE@rlyAtQues{\begingroup#1}\fi}
```

`\bItemInsert{fig}{content}\eProbInsert`

The `\bItemInsert` command is the `\item`-version of `\bProbInsert`. Place `\bItemInsert` with its argument just after the `\item`, as illustrated below. At the end of the question place `\eProbInsert`.

```
\begin{problem*}[5ea]
Solve each without error.
\begin{parts}
\item \bItemInsert{\insertFig} \PROB
\PROB\space\PROB\space\PROB\eProbInsert
\begin{solution}[1in]
This is the solution. \PROB
\end{solution}
...
\end{parts}
\end{problem*}
```

```
3886 \def\bItemInsert#1{\begingroup#1\noindent\ignorespaces}
3887 \def\eProbInsert{\par\endgroup\@restorepar\par}
```

16 Insertions in the solution section

Its a bit trickier to include a wrapped graphic in the solution section. The primary `\probInsertSoln` command for doing this is `\probInsertSoln`

```
\bProbInsert{\insertFig}
\probInsertSoln[\protect\vskip3pt]{\insertFig}
\begin{problem}[10]\begin{cq*}
\PROB\space\PROB

\PROB\space\PROB\space\PROB\space\PROB\cqQS{}{\RESTOREPAR}

\PROB\cqQS{\eProbInsert}{}
\end{cq*}
```

```

\begin{solution}[1in]
\ifcqsA\else\textit{Question}: \cqCopiedQues
\par\medskip\noindent\textit{Solution}:\space\fi
This is the solution: \PROB
\end{solution}
\end{problem}

```

`\probInsertSoln[opt]{cmd}` The command `\cmd` is a expands to the figure insertion; for example, `\gdef\insertFig{\parpic[r]{\FIG}}`, the second argument would read `\insertFig`. The first argument was designed for vertical adjustment, as needed. In the sample file `ins-fig.tex`, we insert an extra 3pts, see the verbatim listing above.

```

3888 \newcommand{\probInsertSoln}[2] [] {\writeToSolnFile{#1%
3889 \protect\def\protect
3890 \priorexlabelheader{\protect#2\protect\noindent
3891 }}}

```

Sometimes you need to insert `\RESTOREPAR` to make the paragraphs layout as they should. This is illustrated above.

```

3892 \def\RESTOREPAR{\let\par\@par}

```

```

3893 </package>
3894 <*textbook>
3895 \ProvidesFile{eqtextb.def}
3896 [2016/01/18 v4.6 Cmds used by the fortextbook option (dps)]

```

17 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.

17.1 Setting options with `\textbookOpts`

```
\textbookOpts We set up a command for setting the options for the fortextbook option.
marginans 3897 \define@boolkey{eqe@tbopts}[is]{instred}[true]{}
inlineans 3898 \define@boolkey{eqe@tbopts}[is]{studented}[true]{%
marginsonleft 3899   \ifisstudented
                ssols
                lsols
```

Now let's try to filter out the even-numbered problems for the student edition.

```
3900 \tbfilterOutEvenNums
```

The above command is normally `\let` to `\@gobble`.

```
3901 \fi
3902 }
```

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.

```
3903 \def\tb@beginexam@code{%
```

```
3904 \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.

```
3905 \newcommand{\tbfilterOutEvenNums}{%
```

exerquiz changed `\eqEXT` to two variables, so we make the same change here

```
3906 \def\eqEXT##1##2{\ifodd##1\let\eqe@next\relax\else
```

```
3907 \def\eqe@next{\gobbletoEndEXT}\fi\eqe@next}%
```

```
3908 }
```

```
3909 \newcommand{\tballowAllNums}{%
```

exerquiz changed `\eqEXT` to two variables, so we make the same change here

```
3910 % \let\eqEXT\@gobble
```

```
3911 \let\eqEXT\@gobbletwo
```

```
3912 \let\endeqEXT\relax
```

```
3913 }
```

```
3914 \define@boolkey{eqe@tbopts}[is]{marginans}[true]{}
```

```
3915 \define@boolkey{eqe@tbopts}[is]{inlineans}[true]{}
```

```
3916 \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.

```
3917 {\@mparswitchfalse\reversemarginpar}
```

```
3918 \define@boolkey{eqe@tbopts}[show]{ssols}[true]{}
```

```
3919 \define@boolkey{eqe@tbopts}[show]{lsols}[true]{%
```

```
3920 \ifshowlsols\let\tb@soln@choice\tb@showlsols\fi}
```

The default settings are true for `studented` and false for `instred`.

```
3921 \newcommand{\textbookOpts}[1]{\setkeys{eqe@tbopts}{#1}}%
```

We do not allow both `instred` and `studented` to be true.

```
3922 \ifisinstred\global\isstudentedfalse\else
```

```
3923 \ifisstudented\global\isinstredfalse
```

```
3924 \fi\fi
```

Added this part in in case `\textbookOpts` comes after `\marparboxwidth`.

```
3925 \ifdim\tbmarparboxwidth=1sp\else
```

```
3926 \expandafter\tbMakeFinalCalcs\fi
```

```
3927 }
```

As mentioned above, the default settings are `true` for `studented` and `false` for `instred`.

```
3928 \isstudentedtrue
3929 \isinstredfalse
3930 \ismarginansfalse
3931 \isinlineansfalse
```

The command is available only in the preamble.

```
3932 \@onlypreamble{\textbookOpts}
```

`\turnOffMarAnsOnAnsInline` `\turnOnMarAnsOffAnsInline` `\toggleInstrAns` These three command may not be useful in the creation of a textbook, but you never know, I used them in my demo doc `fortextbook.tex` to turn off and on the display of the answers (change margin to inline, change inline to margin, and toggle margin and inline).

```
3933 \newcommand{\turnOffMarAnsOnAnsInline}{%
3934   \global\ismarginansfalse\global\isinlineanstrue
3935   \insMidMarg{\global\ismarginansfalse
3936     \global\isinlineanstrue}%
3937 }
3938 \newcommand{\turnOnMarAnsOffAnsInline}{%
3939   \global\ismarginanstrue\global\isinlineansfalse
3940   \insMidMarg{\global\ismarginanstrue
3941     \global\isinlineansfalse}%
3942 }
3943 \newcommand{\toggleInstrAns}{%
3944   \ifisinstred\ifismarginans
3945     \global\ismarginansfalse\global\isinlineanstrue
3946     \insMidMarg{\global\ismarginansfalse
3947       \global\isinlineanstrue}%
3948   \else
3949     \global\ismarginanstrue\global\isinlineansfalse
3950     \insMidMarg{\global\ismarginanstrue
3951       \global\isinlineansfalse}%
3952   \fi\fi
3953 }
```

17.2 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 3954 \newif\ifWithinANSGrp\WithinANSGrpfalse
3955 \newif\iffb@isANSListOpen\fb@isANSListOpenfalse
```

```

3956 \newcommand{\bGrpANS}{%
3957   \if\probstar*\else
3958     \PackageError{eqexam}{Use of \string\bGrpANS\space
3959       only applies\MessageBreak to the problem* environment}{Please
3960       remove this \string\bGrpANS.}%
3961   \fi
3962   \ifWithinANSGrp
3963     \global\WithinANSGrpfalse
3964     \let\tb@next\relax
3965     \PackageError{eqexam}{\string\bGrpANS\space already open}
3966     {You issued an earlier \string\bGrpANS,
3967       but did not close it.}%
3968   \else
3969     \global\WithinANSGrptrue
3970     \global\ftb@isANSListOpenfalse
3971     \def\tb@next{\ANS}%
3972   \fi
3973   \tb@next
3974 }
3975 \newcommand{\eGrpANS}{%
3976   \if\probstar*\else
3977     \PackageError{eqexam}{Use of \string\eGrpANS\space
3978       only applies\MessageBreak to the problem* environment}{Please
3979       remove this \string\eGrpANS.}%
3980   \fi
3981   \ifWithinANSGrp
3982     \global\WithinANSGrpfalse
3983     \def\tb@next{\ANS}%
3984   \else
3985     \let\tb@next\relax
3986     \PackageError{eqexam}{\string\eGrpANS\space already closed}
3987     {You've issued two consecutive \string\eGrpANS\space
3988       commands,\MessageBreak either remove this one
3989       or the previous one.}%
3990   \fi
3991   \tb@next
3992 }

```

\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.

```

3993 \newcommand{\ANS}{\@ifstar{\let\tb@istart=1\tb@ANS}
3994   {\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 \eGrpANS. We keep this to revert to this definition if this new feature causes problems.

```

\newcommand{\tb@ANS}[1]{%
  \ifinstred

```

```

\ifisinlineans\if\tb@istart0\ANSFmt{\theequationnoi}{#1}\fi\fi
\ifismarginans
  \edef\eqe@prehold{\noexpand\par\kern0pt\noindent
    \if\probstar*%
      \noexpand\begin{eqeList}[\tb@wparts@len]{%
        \noexpand\eqedsplyOnlyFrst{\theequationnoi}%
        {\thepartno}\noexpand\eqe@hspannerMrg
        \noexpand\makebox[\noexpand\tb@mrpartwidth]{%
          \noexpand\tb@mrpartFmt{\thepartno}}}%
      \else
        \noexpand\begin{eqeList}%
          {\noexpand\tb@mrpartDigitFmt{%
            \theequationnoi\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.

```

3995 \def\ftb@defineInsSpan#1{\def\ftb@argi{#1}\ifx\ftb@argi\@empty
3996   \def\ftb@InsSpan{}\else\ftb@spanPrts{#1}%
3997   \def\ftb@InsSpan{\noexpand\hspace{-\labelsep}}%
3998   \noexpand\textcolor{MRGPARTcolor}{--}\noexpand
3999   \makebox[\noexpand\tb@mrpartwidth]{\noexpand
4000   \tb@mrpartFmt{\ftb@EndSpanPrts}}\eqe@hspannerMrg}\fi
4001 }

```

`\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`.

```

4002 \def\ftb@spanPrts#1{#{%
4003   \advance\value{partno}by#1\relax
4004   \xdef\ftb@EndSpanPrts{\thepartno}}%
4005 }

```

`\ftb@EqeListPrtsFmt` is the internal formatting used within the `eqeList` for the part letter.

```

4006 \def\ftb@EqeListPrtsFmt{\noexpand
4007   \makebox[\noexpand\tb@mrpartwidth]{\noexpand
4008   \tb@mrpartFmt{\thepartno}}\nobreak
4009 }

```

`\ftb@OpenEqeListPrts` opens an `eqeList` environment, and displays the question number (optionally) and the part number.

```

4010 \def\ftb@OpenEqeListPrts{\noexpand
4011   \begin{eqeList}[\tb@wparts@len]{\noexpand
4012   \eqedsplyOnlyFrst{\theequationnoi}%
4013   {\thepartno}\noexpand\eqe@hspannerMrg\ftb@EqeListPrtsFmt}%
4014 }

```


`\ftb@CloseEqeList` closes the `eqeList` after inserting `\qe@prehold` and the content, #1.

```

4015 \def\ftb@CloseEqeList#1{\expandafter\insMidMarg%
4016   \expandafter{\qe@prehold#1\end{eqeList}}}%
4017 }

```

`\grpANSDelimiter` delimits the parts when `\bGrpANS/\eGrpANS` is used. May be redefined.

```

4018 \newcommand{\grpANSDelimiter}{\textcolor{MRGPARTcolor}{,}\space}

```

`\tb@ANS` does the main work of `\ANS`.

```

4019 \newcommand{\tb@ANS}[2][\%
4020   \ifisinstred
4021     \ifisinlineans
4022       \if\tb@istart0\ANSFmt{\theeqquestionnoi}{#2}\fi
4023     \fi
4024     \ifismarginans
4025       \ftb@defineInsSpan{#1}%

```

We create the code that we will introduce into `\insMidMarg`, this will be introduced prior to #2.

```

4026   \edef\qe@prehold{%
4027     \if\probstar%

```

If this question is one with parts...

```

4028     \ifftb@isANSListOpen

```

If the list is already open (`\ifftb@isANSListOpen`), we just add content to the `eqeList` environment.

```

4029     \ftb@EqeListPrtsFmt\noexpand\qe@hspannerMrg
4030   \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.

```

4031     \noexpand\par\kernOpt\noindent
4032     \ftb@OpenEqeListPrts\ftb@InsSpan
4033   \fi
4034   \else

```

This is a question without parts.

```

4035     \noexpand\begin{eqeList}%
4036     {\noexpand\tb@mrgDigitFmt%
4037     \theeqquestionnoi\qe@decPointMrg}}%
4038   \fi
4039   }%

```

We have finished constructing `\qe@prehold`. We next set `\ftb@isANSListOpen` to true, if `\WithinANSGrp` is true.

```

4040   \ifWithinANSGrp\global\ftb@isANSListOpentru\fi

```

If we are within an open group, we emit `\insMidMarg` with the `\qe@prehold`, followed by #2, and a comma-space combo, but *we do not close* the `eqeList` environment.

```

4041   \ifWithinANSGrp

```

```

4042             \expandafter\insMidMarg\expandafter
4043                 {\eqe@prehold#2\grpANSDelimiter}%
4044         \else
         This is the normal case, we insert \eqe@prehold, #2, and close the eqeList envi-
         ronment.
4045             \expandafter\insMidMarg%
4046             \expandafter{\eqe@prehold#2\end{eqeList}}%
4047             \global\ftb@isANSListOpenfalse
4048         \fi
4049     \fi % \ifismarginans
4050     \fi %\ifisinstded
4051 }
End (10/13)

```

eqeList An environment used to format the answers in the margins, when marginans is in effect.

```

4052 \newenvironment{eqeList}[2][\tb@woparts@len]{\begin{list}{#2}{%
4053     \def\argi{#1}\setlength{\labelwidth}{#1}%
4054     \ifx\argi\tb@woparts@len
4055     \settowidth{\labelsep}{\eqe@prtsepMrg}\else
4056     \settowidth{\labelsep}{\eqe@hspannerMrg}\fi
4057     \setlength{\leftmargin}{\labelwidth+\labelsep}%
4058     \setlength{\parskip}{0pt}\setlength{\partopsep}{0pt}%
4059     \setlength{\topsep}{1pt}\setlength{\parsep}{0pt}%
4060     \setlength{\itemindent}{0pt}\setlength{\itemsep}{3pt}%
4061 } \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)}
4062 \newcommand{\mrgDigitFmt}[1]{\def\tb@mrgDigitFmt##1{#1}}
4063 \mrgDigitFmt{#1}

```

\mrgPartFmt Format of the part (including possibly the parentheses), example give above.

```

4064 \definecolor{MRGPARTcolor}{named}{black}
4065 \newcommand{\mrgPartFmt}[1]{\def\tb@mrgPartFmt##1{#1}}
4066 \mrgPartFmt{\textcolor{MRGPARTcolor}{(\hfil#1\hfil)}}

```

ANScolor The default color of the answers that appear in the margins or inline.

```

4067 \definecolor{ANScolor}{rgb}{0,0,.8}

```

\ANSFmt The command that sets the format, may be redefined as needed. Used in the \ANS command above.

```

4068 \newcommand{\ANSFmt}[2]{\textcolor{ANScolor}{#2}}

```

We have two environments that we use in three different situations:

- `eqequestions` 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.”
- `eqeList` 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 4069 \def\eqe@decPointPrb{.} % decimal point of prob number
\prbNumPrtsep 4070 \def\eqe@prtsepPrb{\ } % prob with parts, space after part
4071 \def\eqe@hspannerPrb{\ } % space after prob number
4072 \providecommand{\prbDecPt}[1]{\def\eqe@decPointPrb{#1}}
4073 \providecommand{\prbPrtsep}[1]{\def\eqe@prtsepPrb{#1}}
4074 \providecommand{\exPrtsep}[1]{\def\eqe@prtsepPrb{#1}}
4075 \providecommand{\prbNumPrtsep}[1]{\def\eqe@hspannerPrb{#1}}

\solDecPt Basic parameters for the problems in the solution sets.
\solPrtsep 4076 \def\eqedecPointSoln{.} % decimal point of prob number
\solNumPrtsep 4077 \def\eqe@prtsepSoln{\ } % prob with parts, space after part
4078 \def\eqe@hspannerSoln{\ } % space after prob number
4079 \newcommand{\solDecPt}[1]{\def\eqedecPointSoln{#1}}
4080 \newcommand{\solPrtsep}[1]{\def\eqe@prtsepSoln{#1}}
4081 \providecommand{\solNumPrtsep}[1]{\def\eqe@hspannerSoln{#1}}

\mrgDecPt Basic parameters for the problems in the margins.
\mrgPrtsep 4082 \def\eqe@decPointMrg{.} % decimal point of prob number
\mrgNumPrtsep 4083 \def\eqe@prtsepMrg{\ } % prob with parts, space after part
4084 \def\eqe@hspannerMrg{\ } % space after prob number
4085 \newcommand{\mrgDecPt}[1]{\def\eqe@decPointMrg{#1}}
4086 \newcommand{\mrgPrtsep}[1]{\def\eqe@prtsepMrg{#1}}
4087 \newcommand{\mrgNumPrtsep}[1]{\def\eqe@hspannerMrg{#1}}

\setMarIndents Sets some dimensions used by the eqeList environment. \tb@woparts@len is
calculated and is used as the default gutter width in eqeList. \tb@wparts@len
is used for the gutter width for the gutter width when there is a problem with
parts. Finally, \tbmrgpartwdth in \ANS and is used for the width of a \makebox
that enclosed the part letter.
4088 \newcommand{\setMarIndents}[3][\normalsize\normalfont]{%

```

```

4089 \settowidth{\@tempdima}{#1#2\eqe@decPointMrg}%
4090 \xdef\tb@woparts@len{\the\@tempdima}%
4091 \settowidth{\@tempdima}%
4092     {#1#2\eqe@decPointMrg\eqe@hspannerMrg#3}%
4093 \xdef\tb@wparts@len{\the\@tempdima}%
4094 \settowidth{\@tempdima}{#1#3}%
4095 \xdef\tb@mrpartwidth{\the\@tempdima}%
4096 }}
4097 \setMarIndents{00}{(d)}

```

`\setSolnIndent` Used to set the some parameters used by `eqequestions`, in the solutions file.

```

4098 \newdimen\solnGutter
4099 \newcommand{\setSolnIndent}[3][\normalsize\normalfont\bfseries]{%
4100     {\settowidth{\@tempdima}{#1#2\eqe@decPointSoln\eqe@hspannerSoln}%
4101     \global\solnGutter\@tempdima
4102     \settowidth{\@tempdima}{#1#3}%
4103     \xdef\tb@solnpartwidth{\the\@tempdima}%
4104 }}
4105 \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 `0pt`, we set the length of `\eqemargin` using the current value; otherwise, we use the value determined by `\setSolnIndent`, above.

```

4106 \renewcommand{\setSolnMargins}[1]{%
4107     \ifdim\solnGutter=0pt \setlength\eqemargin{#1}\else
4108     \setlength\eqemargin{\solnGutter}\fi\ignorespaces}
4109 %\newcommand{\defaultSolnIndent}{\gdef\solnGutter{0pt}}
4110 \newcommand{\defaultSolnIndent}{\global\solnGutter=0pt}
4111 \defaultSolnIndent

```

`\prbNumFmt` We redefine `\exlabeledformat`, `\exsllabeledformat`, and `\exsllabeledformatwp`.
`\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)}

```

```

4112 \renewcommand{\exlabeledformat}{%
4113     \tbprbNumFmt{\theequationnoi\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.

```

4114 \newcommand{\prbNumFmt}[1]{\def\tbprbNumFmt##1{#1\eqe@hspannerPrb}}
4115 \prbNumFmt{\textbf{#1}}

```

Redefine `\exsllabeledformat`, and `\exsllabeledformatwp`

```

4116 \renewcommand{\exsllabeledformat}{\string\tbsolWoPrtsFmt{%
4117     \theequationnoi\string\eqe@decPointSoln}}
4118 \renewcommand{\exsllabeledformatwp}{\string\tbsolWPrtsFmt%
4119     {\string\eqedsplyOnlyFrst{\theequationnoi}{\thepartno}}}%

```

```

4120   {\thepartno}%
4121 }

\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.

4122 \newcommand{\solWoPrtsFmt}[1]{\def\tbsolWoPrtsFmt##1{%
4123   \makebox[Opt][r]{#1\eqe@hspannerSoln}}
4124 \solWoPrtsFmt{\textbf{#1}}
4125 \newcommand{\solWPrtsFmt}[2]{\def\tbsolWPrtsFmt##1##2{%
4126   \makebox[Opt][r]{#1\eqe@hspannerSoln}%
4127   \makebox[\tbsolnpartwidth][l]{#2}\eqe@prtsepSoln%
4128 }}
4129 \solWPrtsFmt{\textbf{#1}}{(\hfil#2\hfil)}

```

An alternate definition for `\solWPrtsFmt`, used by `\hangSolWPrtsFmt`.

```

4130 \newcommand{\solWPrtsFmt@hang}[2]{%
4131   \def\tbsolWPrtsFmt##1##2{%
4132     \makebox[Opt][r]{#1\eqe@prtsepSoln}%
4133     \makebox[\tbsolnpartwidth][l]{#2}\eqe@hspannerSoln}%
4134 }}

```

`\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.

```

4135 \let\bpartsmrk\relax
4136 \let\epartsmrk\relax
4137 \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.

```

4138   \def\prior@parts@hook{%
4139     \ifisleadin\else
4140       \writeT@SolnFile{^^J\protect\bpartsmrk}\fi}%
4141   \def\post@parts@hook{%
4142     \writeT@SolnFile{\protect\epartsmrk^^J}}%
4143   \def\bpartsmrk{\global\firstitemtrue\begin{eqepartsquestions}}%
4144   \def\epartsmrk{\end{eqepartsquestions}\global\firstitemfalse}%

```

We must also redefine `\solWPrtsFmt` by letting it to `\solWPrtsFmt@hang`, then executing it using the parameters passed.

```

4145   \let\solWPrtsFmt\solWPrtsFmt@hang
4146   \solWPrtsFmt{#1}{#2}%
4147 }
4148 %\@onlypreamble\hangSolWPrtsFmt

```

Define some switches, token registers, and boxes for managing the answers and marginal notes.

```

4149 \newif\ifexamenv \examenvfalse

```

```

4150 \newif\iffirstemit \firstemittrue
4151 \newtoks\txtbkt@ks \txtbkt@ks={}
4152 \newtoks\txtbkt@ksi \txtbkt@ksi={}
4153 \newbox\txtbkb@xb@t
4154 \newbox\txtbkb@xt@p
4155 \newbox\txtbkb@xh@ld
4156 \let\tbTopMargin\relax
4157 \let\tbBotMargin\relax
4158 \long\def\tb@addtoTopMargin#1{\txtbkt@ksi=#1}%
4159 \edef\eqe@tmphold{\the\txtbkt@ksi\the\txtbkt@ks}%
4160 \global\txtbkt@ks=\expandafter{\eqe@tmphold}%
4161 }
4162 \newcommand{\tb@addtoMargin}[1]{%
4163 \edef\eqe@tmphold{\the\txtbkt@ks}%
4164 \global\txtbkt@ks=\expandafter{\eqe@tmphold#1}%
4165 }

```

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 4166 \newcommand{\tb@showlsols}{\includecomment{lsol}\excludecomment{ssol}}
```

`ssol` Place short solutions in this environment.

```
\tb@showssols 4167 \newcommand{\tb@showssols}{\includecomment{ssol}\excludecomment{lsol}}
```

The default is to show the short solutions.

```

4168 \let\tb@soln@choice\tb@showssols
4169 %\let\tb@sols@choice\tb@showssols

```

17.3 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

```

4170   \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.

```

4171   \newcommand{\setTopMargin}[1]{%
4172     \gdef\tbTopMargin{\color@begingroup\normalcolor
4173       #1\color@endgroup}%
4174     \gdef\tbSaveTopMargin{\color@begingroup\normalcolor
4175       #1\color@endgroup}}
4176   \let\tbTopMargin\@empty
4177   \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

```

4178   \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.

```

4179   \newcommand{\setBotMargin}[1]{\gdef\tbBotMargin{#1}%
4180     \gdef\tbSaveBotMargin{#1}}
4181   \let\tbBotMargin\@empty
4182   \let\tbSaveBotMargin\@empty

4183   \newcommand{\restoreLastTopMargin}{\expandafter\setTopMargin
4184     \expandafter{\tbSaveTopMargin}}
4185   \newcommand{\restoreLastBotMargin}{\expandafter\setBotMargin
4186     \expandafter{\tbSaveBotMargin}}

```

`\insMidMarg` `\insMidMarg` is a `\parbox` that will hold the material in the margin.

`\MarParBoxFmt` The formatting for the marginal `\parbox`

```

4187 \newcommand{\MarParBoxFmt}{\normalsfcodes
4188   \normalfont\normalsize\normalbaselines\parindent0pt

```

```

4189 \vbadness\Mi \hbadness5000 \tolerance9999
4190 \parskipOpt\raggedright %\spaceskip=Opt\xspaceskip=Opt
4191 \setlength{\linewidth}{\tbmarparboxwidth}%
4192 }

```

`\tbmarparboxwidth` The width of the margin box. Initial value of 1sp, if the user does not reset the value, it is a package error.

```

4193 \newlength\tbmarparboxwidth
4194 \setlength\tbmarparboxwidth{1sp}

```

`MidMargcolor` The default color of text of the middle level

```

4195 \definecolor{MidMargcolor}{rgb}{0,0,.8}
4196 \newcommand{\midMargFmt}[1]{%
4197   \def\tb@midMargFmt{\normalfont\normalsize\normalcolor#1}}
4198 \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`.

```

4199 %\def\eqe@MarParBox#1{\parbox[b][\textheight][t]{%
4200 %   {\tbmarparboxwidth}{\color{MidMargcolor}#1}}
4201 \def\eqe@MarParBox#1{\lower\footskip\hbox{%
4202   \leavevmode\parbox[b][\textheight+\footskip][t]{%
4203     {\tbmarparboxwidth}{\tb@midMargFmt#1}}} %

```

Finally, we get to the `\insMidMarg`, this is used to write to the middle level.

```

4204 \newcommand{\insMidMarg}[1]{%
4205   \let\eqe@margininsert\@empty
4206   \expandafter\tb@addtoMargin\expandafter{\eqe@margininsert#1}%
4207 }

```

`\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

```

4208 \newcommand{\tbPreMarginHeader}{\par\penalty0 \kern3pt}
4209 \newcommand{\tbPostMarginHeader}{\par\nobreak}
4210 \definecolor{HEADERcolor}{named}{black}
4211 \newcommand{\cngMargHeadColorTo}[1]{\insMidMarg{\gdef\tb@MHC{#1}}}
4212 \newcommand{\resetMargHeadColor}{\insMidMarg{\gdef\tb@MHC{HEADERcolor}}}
4213 \resetMargHeadColor
4214 \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`.

```

4215 \newcommand{\insMargHead}[2] [] {% dps
4216   \insMidMarg{\tb@marginHeader{#1}{#2}}
4217 \newcommand{\insProbHead}[2] [] {%
4218   \def\tb@argi{#1}\ifx\tb@argi\@empty
4219     \protected@xdef\currProbHead{#2 \tbcontinued}\else
4220     \protected@xdef\currProbHead{#1}\fi
4221   \ifisinstred\ifismarginans
4222     \insMidMarg{\tb@marginProbHeader{#1}{#2}}\fi\fi
4223 }
4224 \newcommand{\tb@marginProbHeader}[2] {%
4225   \def\tb@argi{#1}\ifx\tb@argi\@empty
4226     \tb@marginHeader{#2 \tbcontinued}{#2}\else
4227     \tb@marginHeader{#1}{#2}\fi
4228 }
4229 \newcommand{\tb@marginHeader}[2] {\tbPreMarginHeader
4230   \tbMarginHeaderFmt{#2}\def\tb@argi{#1}\ifx\tb@argi\@empty
4231   \mark{#2}\else\mark{#1}\fi\tbPostMarginHeader
4232 }

```

`\tbcontinued` The continue annot that appear when a problem set flows over to the next page.

```

4233 \newcommand{\tbcontinued}{(cont.)}

```

`\tbplaceMargins` Redefine this macro to set the locations of the margins we are writing to.

```

4234 \newcommand{\marparboxwidth}[1] {%
4235   \setlength\tbmarparboxwidth{#1}%
4236   \setlength{\marginparwidth}{\tbmarparboxwidth}%
4237   \tbMakeFinalCalcs
4238 }
4239 \@onlypreamble\marparboxwidth
4240 \newcommand{\chkmarginboxwidth}{%
4241   \ifdim\tbmarparboxwidth=1sp \PackageError{eqexam}%
4242   {You have not set the value of\MessageBreak
4243     \string\marparboxwidth}%
4244   {Define the \string\marparboxwidth\space command}\fi
4245 }

```

`\ifmarginsonleft` A Boolean switch, if true, all margins are on the left; otherwise, they alternate

```

4246 \newif\ifmarginsonleft \marginsonleftfalse

```

`\tbSetupForMargins` We compute `\oddsidemargin`, `\evensidemargin`, and `\textwidth`

```

4247 \newcommand{\tbSetupForMargins}{%
4248   \ifmarginsonleft
4249     \setlength{\oddsidemargin}{\tbmarparboxwidth+\marginparsep}%

```

```

4250     \setlength{\evensidemargin}{\oddsidemargin}%
4251     \setlength{\textwidth}{\paperwidth-2in-\oddsidemargin}%
4252   \else
4253     \setlength{\oddsidemargin}{0pt}%
4254     \setlength{\evensidemargin}{\tbmarparboxwidth+\marginparsep}%
4255     \setlength{\textwidth}{%
4256       \paperwidth-2in-\oddsidemargin-\evensidemargin}%
4257   \fi
4258 }

```

`\tbplaceMargins` We calculate the coordinates of the lower left hand corner of the margin `\parbox` depending on the value of `\ifmarginsonleft`.

```

4259 \newcommand{\tbplaceMargins}{%
4260   \setlength{\@tempdima}{%
4261     \paperheight-1in-\topmargin-\headheight-\headsep-\textheight}%
4262   \xdef\@evenlly{\strip@pt\@tempdima}%
4263   \xdef\@oddlly{\@evenlly}%
4264   \setlength{\@tempdima}{1in}%
4265   \xdef\@evenllx{\strip@pt\@tempdima}%
4266   \ifmarginsonleft\else
4267     \setlength{\@tempdima}{1in+\textwidth+\marginparsep}\fi
4268   \xdef\@oddlx{\strip@pt\@tempdima}%
4269 }}

```

`\tbMakeFinalCalcs` Executed by `\marparboxwidth`

```

4270 \newcommand{\tbMakeFinalCalcs}{%
4271   \tbSetupForMargins
4272   \tbplaceMargins
4273 }

```

`\tbminskipbtnlayers` `\tbminskipbtnlayers` is the minimum skip between layers (top, middle, bottom) Executed by `\marparboxwidth`

```

4274 \newlength\tbminskipbtnlayers
4275 \setlength{\tbminskipbtnlayers}{6pt}

4276 \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.

```

4277 \newenvironment{carryOverFmt}[1]{#1\let\tb@carryoverFmt\@empty
4278   \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.

```

4279 \def\carryoverFmt#1{%
4280     \ifx\tb@carryoverFmt\@empty
4281         \global\let\tb@carryoverFmt\@empty
4282         \xdef\tb@co@page{\thepage}%
4283         \gdef\tb@carryoverFmt{\ifnum\thepage>\tb@co@page\relax
4284             #1\global\let\tb@carryoverFmt\@empty\fi}%
4285     \fi
4286 }
4287 \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.

```

4288 \let\tb@rest@reMarginFmt\relax
4289 \def\tb@insertCarryOver#1{%
4290     \let\tb@rest@reMarginFmt\relax
4291     \ifiscarryover\ifx\tb@carryoverFmt\@empty\else
4292         \let\tb@rest@reMarginFmt\tb@midMargFmt
4293         \tb@carryoverFmt\fi\fi
4294     #1 \tb@rest@reMarginFmt
4295 }

```

`\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.

```

4296 \newif\iftb@shipoutPermitted \tb@shipoutPermittedtrue
4297 \newcommand{\turnOnFTBShipout}{\global\tb@shipoutPermittedtrue}
4298 \newcommand{\turnOffFTBShipout}{\global\tb@shipoutPermittedfalse}

```

`\turnOnFTBShipout` `\turnOnFTBShipout` turns on the shipout, the default, and `\turnOffFTBShipout` turns it off.

```

4299 \newcommand{\eqe@tb@shipout}{\iftb@shipoutPermitted
4300     \ifnum\arabic{page}=\tblastpageshipped
4301     \let\tb@so@next\relax\else
4302     \xdef\tblastpageshipped{\arabic{page}}%
4303     \def\tb@so@next{\eqe@tb@ship@ut}\fi
4304     \expandafter\tb@so@next\fi
4305 }
4306 \def\tblastpageshipped{-100}

```

Here is the actual shipout code for writing to the margins.

```

4307 \newcommand{\eqe@tb@ship@ut}{%
4308     \fboxsep=0pt\setlength{\unitlength}{1pt}%
4309     \global\setbox\txtbkb@xb@t=\vbox\bgroup
4310         \color@bebingroup

```

```

4311         \hsize=\tbmarparboxwidth
4312         \vsize=\textheight
4313         \MarParBoxFmt
4314         \csname tbTopMargin\endcsname
4315         \vskip\tbminskipbtnlayers
4316 \set@typeset@protect
4317         \the\txtbkt@ks
4318         \color@endgroup\vfil
4319     \egroup
4320     \global\setbox\txtbkb@xt@p=\vsplit\txtbkb@xb@t to\textheight
4321     \ifvoid\txtbkb@xb@t\global\iscarryoverfalse
4322     \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.

```

4323     \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.

```

4324         \bgroup\setbox2=\vbox{%
4325             \color@begingroup\normalcolor
4326             \hsize=\tbmarparboxwidth\kern0pt
4327             \MarParBoxFmt\csname tbBotMargin\endcsname
4328             \color@endgroup
4329             \kern0pt
4330         }%

```

We reduce `\textheight` by the height of `\tbBotMargin`

```

4331         \dimen0=\textheight
4332         \advance\dimen0-\ht\txtbkb@xh@ld
4333         \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`

```

4334         \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`.

```

4335         \global\setbox\txtbkb@xb@t=\vbox{%
4336             \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`).

```

4337         \global\setbox\txtbkb@xt@p=\vbox{\unvbox0
4338             \vfill\vskip\tbminskipbtnlayers
4339             \vfil\unvbox2\relax}\egroup
4340     \fi
4341     \ifodd\value{page}%
4342     \put(\@oddlx,\@oddily){%

```

```

4343         \eqe@MarParBox{\unvbox\txtbkb@xt@p}}\else
4344         \put(\@evenllx,\@evenlly){%
4345         \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.
4346     \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.
4347     \if!\splitbotmark!\global\let\tb@sbm@exp\relax\else
4348         \xdef\tb@sbm@exp{\noexpand\tbPreMarginHeader
4349         \noexpand\tbMarginHeaderFmt{\splitbotmark}}%
4350         \noexpand\tbPostMarginHeader
4351         \noexpand\par\kern3pt}%
4352     \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).
4353         \global\txtbkt@ks=\expandafter{\tb@sbm@exp
4354         \tb@insertCarryOver{\unvbox\txtbkb@xb@t}}%
4355     \fi
4356 }

```

\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.

```

4357 \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.
4358     \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.
4359     \ifiscarryover\def\eqeifnext{\csname iftrue\endcsname}%
4360     \PackageInfo{eqexam}{Carry over of content in margin
4361     from page \thepage.\MessageBreak Creating a blank page}\else
4362     \def\eqeifnext{\csname iffalse\endcsname}\fi\eqeifnext#1\fi}

```

\setFullWidthHeader Makes the running header full width.

```

4363 \newcommand{\setFullWidthHeader}{%
4364     \setlength{\@tempdima}{%
4365         \evensidemargin+\tbmarparboxwidth+\marginparsep}%
4366     \edef\@headoffset{\the\@tempdima}%
4367     \def\@evenhead{\makebox[0pt]{\makebox[0pt][1]
4368         {\thepage}\hspace{\@headoffset}}\hfil\slshape\leftmark}%
4369     \ifmarginsonleft
4370         \def\@oddhead{\makebox[0pt]{\makebox[0pt][1]

```

```

4371         {\slshape\rightmark}\hspace{\@headoffset}}\hfil\thepage}%
4372     \else
4373         \def\@oddhead{\slshape\rightmark}\hfil\makebox[0pt]
4374             {\hspace{\@headoffset}\makebox[0pt][r]{\thepage}}}%
4375     \fi
4376 }

```

17.4 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

```

4377 \newif\ifchapterexercises \chapterexercisesfalse
4378 \let\chaptersolutions\relax
4379 \def\tb@EndOfChapterExercises{%
4380     \let\include@solutions@chapter\include@solutions
4381     \def\includeexersolutions{%
4382         \include@solutions@chapter
4383         \global\let\include@solutions\relax
4384     }%

```

`\chaptersolutions` is redefined from `\relax`. Input current solutions, close stream, open stream.

```

4385     \def\chaptersolutions{%
4386         \includeexersolutions
4387         \immediate\closeout\ex@solns
4388         \newwrite \ex@solns \global\let\quiz@solns\ex@solns
4389         \immediate\openout \ex@solns \jobname.sol
4390         \ifvspacewithsolns\writeAllAnsAtEnd\fi
4391     }%
4392 }

```

```

4393 \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.

```

4394 \excludecomment{afterChapSolns}
4395 %\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

```

4396 \def\NewCommentCutFile{\def\CommentCutFile{solnsAtEnd.cut}}
4397 \def\RestoreCommentCutFile{\def\CommentCutFile{comment.cut}}
4398 \@ifundefined{BeforeIncludedComment}{%
4399 \long\def\solnsAtEndcomment
4400 #1{\message{Special comment '#1'}}%
4401   \csarg\def{#1}{\endgroup \message{Processing '#1' comment.}}%
4402   \NewCommentCutFile\SetUpCutFile
4403   % #2 before SetUp, so we can do renaming.
4404   \message{Comment '#1' writing to \CommentCutFile.}%
4405   \ProcessComment{#1}}%
4406   \csarg\def{After#1Comment}{\immediate\closeout\CommentStream
4407   \RestoreCommentCutFile\input{solnsAtEnd.cut}\relax}%
4408   \CommentEndDef{#1}}
4409 }{\long\def\solnsAtEndcomment
4410 #1{\message{Special comment '#1'}}%
4411   \csarg\def{After#1Comment}{\immediate\closeout\CommentStream
4412   \RestoreCommentCutFile\input{solnsAtEnd.cut}\relax}%
4413   \csarg\def{#1}{\NewCommentCutFile\BeforeIncludedComment\relax
4414   \ProcessComment{#1}}%
4415   \CommentEndDef{#1}}
4416 }
4417 \solnsAtEndcomment{solnsAtEnd}
4418 \newcommand{\exercisesAtEndOfChapter}{%
4419   \ifeq@nosolutions\else
4420     \typeout{^^J!!!!Executing in chapter solutions!!!!^^J}
4421     \chapterexercisestruetb@EndOfChapterExercises
4422     \ifchapterexercises
4423     \csarg\let{solnsAtEnd}\@gobble
4424     \excludecomment{solnsAtEnd}%
4425     \csarg\let{AftersolnsAtEndComment}\relax
4426     \includecomment{afterChapSolns}\else
4427     \excludecomment{afterChapSolns}\fi
4428   \fi
4429 }
4430 \@onlypreamble\exercisesAtEndOfChapter

```

17.5 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.

```

4431 \newcommand{\setFullWidthLayout}{%
4432   \saveBasicLayoutParams
4433   \setlength{\oddsidemargin}{0in}%
4434   \setlength{\evensidemargin}{\oddsidemargin}%
4435   \setlength{\textwidth}{\paperwidth-2in}%

```

```

4436 \setlength{\linewidth}{\paperwidth-2in}%
4437 \setlength{\columnseprule}{0pt}%
4438 \def\@evenhead{\thepage\hfil\slshape\leftmark}%
4439 \def\@oddhead{\slshape\rightmark\hfil\thepage}%
4440 }

```

`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.

```

4441 \newenvironment{fullwidthtext}{%
4442 \begin{list}{}{%
4443 \setlength{\labelwidth}{0pt}\setlength{\labelsep}{0pt}%
4444 \setlength{\itemindent}{0pt}\setlength{\itemsep}{0pt}%
4445 \setlength{\topsep}{0pt}\setlength{\parsep}{0pt}%
4446 \setlength{\listparindent}{\parindent}%
4447 \setlength{\leftmargin}{0pt}\setlength{\rightmargin}{0pt}
4448 } \item\relax} \end{list}}

```

`\restorePageLayout` Restore the last saved page parameters.

```

4449 \newcommand{\restorePageLayout}{\newpage
4450 \setlength{\oddsidemargin}{\tb@osms}
4451 \setlength\evensidemargin{\tb@esms}
4452 \setlength{\textwidth}{\tb@tws}
4453 \setlength{\linewidth}{\tb@lws}
4454 \setlength{\columnseprule}{\tb@csr}
4455 }

```

Used by `\setFullWidthLayout` just before the page layout parameters are changed.

```

4456 \newcommand{\saveBasicLayoutParams}{%
4457 \xdef\tb@osms{\the\oddsidemargin}%
4458 \xdef\tb@esms{\the\evensidemargin}%
4459 \xdef\tb@tws{\the\textwidth}%
4460 \xdef\tb@lws{\the\linewidth}%
4461 \xdef\tb@csr{\the\columnseprule}%
4462 }

```

`\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}

4463 \newcommand{\initChapAfterSolns}{\newpage
4464   \clearTopMargin\clearBotMargin
4465   \setFullWidthLayout
4466 }

```

`\restoreFromChapAfterSolns` Restores the saved parameters at the end of the chapter solutions, see above for an example.

```

4467 \newcommand{\restoreFromChapAfterSolns}{\newpage
4468   \restorePageLayout\setFullWidthHeader
4469 }

4470 % End of textbook segment
4471 </textbook>
4472 <*package>

```

17.6 We shipout in support of fortetbook

We shipout `\eqe@tb@shipout` to be placed in the margins on every page.

```

4473 \ifeqfortetbook
4474 \AtBeginDocument{\tb@soln@choice
4475   \ifeqwritetomargins\chkmarginboxwidth
4476   \AddToShipoutPicture{\eqe@tb@shipout}\fi}
4477 \fi
4478 </package>
4479 <*textbook>

```

17.7 Modify eqequestions environment

We adjust the `eqequestions` environment to minimize spacing between problems.

```

4480 \eqequestopsep{0pt}
4481 \eqequesparsep{0pt}
4482 \eqequesitemsep{0pt}
4483 \eqequeslistparindent{0pt}
4484 \renewenvironment{eqequestions}{%
4485   \begin{list}{-}{%
4486     \ifwithinsoldoc\let\solnItemMngt\eqeSolnItemMngt\fi
4487     \setlength{\labelwidth}{\eqemargin}%
4488     \setlength{\parsep}{\eqeques@parsep}%
4489     \setlength{\itemsep}{\eqeques@itemsep}
4490     \setlength{\topsep}{\eqeques@topsep}%

```

```

4491 \setlength{\itemindent}{0pt}%
4492 \setlength{\listparindent}{\eqques@listparindent}%
4493 \ifwithinsoldoc\settowidth{\labelsep}{\eqe@hspannerSoln}\else
4494 \settowidth{\labelsep}{\eqe@hspannerPrb}\fi
4495 \setlength{\leftmargin}{\labelwidth}%
4496 }\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.

```

4497 \newcommand{\eqepquestopsep}[1]{\def\eqepques@topsep{#1}}
4498 \newcommand{\eqepquesparsep}[1]{\def\eqepques@parsep{#1}}
4499 \newcommand{\eqepquesitemsep}[1]{\def\eqepques@itemsep{#1}}
4500 \eqepquestopsep{\eqeques@itemsep}
4501 \eqepquesparsep{\eqeques@parsep}
4502 \eqepquesitemsep{\eqeques@itemsep}
4503 \newenvironment{eqepartsquestions}{%
4504 \begin{list}{}{%
4505 \settowidth{\labelwidth}{\eqe@prtsepSoln\hspace{\tbsolnpartwidth}}
4506 \setlength{\parsep}{\eqepques@parsep}%
4507 \setlength{\itemsep}{\eqepques@itemsep}%
4508 \setlength{\topsep}{\eqepques@topsep}%
4509 \setlength{\itemindent}{0pt}%
4510 \settowidth{\labelsep}{\eqe@hspannerSoln}
4511 \setlength{\leftmargin}{\labelwidth}%
4512 }\item\relax}{\end{list}}

```

17.8 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.

```

4513 \long\def\gobbletoEndEXT#1\endeqEXT{\@gobbletwo}
4514 \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`).

```

4515 \if\load@exerquiz\eqe@NO\DoNotFitItIn\fi
4516 \let\fillInFormatDefault@empty
4517 \def\fbInsSolnsStyle{\def\exerSolnsHeadnToc{}}
4518 \def\autoInsSolns{\let\fbInsSolnsStyle\relax}
4519 \AtBeginDocument{\fbInsSolnsStyle}
4520 \renewcommand{\exerSolnInput}{%
4521 \global\let\webnewpage\relax
4522 \ifsolutionsonly\else\immediate\closeout\ex@solns\fi
4523 \ifeq@nosolutions\else\newpage % 2012-03-14
4524 \iftherearesolutions\eq@solutionshook\eqsolutionshook}

```

```

4525         \ifsolutiononly\else\newpage\eq@solutionshook
4526         \eqsolutionshook\fi
4527         \ifx\webnewpage\relax
4528             \gdef\webnewpage{\global\let\webnewpage\newpage}%
4529         \fi
4530         \priorexsectitle\exerSolnsHeadnToc\priorexslinput
4531         \InputIfFileExists{\jobname.sol}{\PackageWarning{exerquiz}
4532             {!!! Solutions to exercises not found}}%
4533     \fi
4534 \fi
4535 }

```

`\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.

```

4536 \setcounter{partno}{1}\edef\firstPartLtr{\thepartno}
???? 6/2/11 When part (a) is hidden we need to generate the questions number
for the the first non-hidden part. Created \iffirstProbNumShown to help but it
not working yet.
4537 \newif\iffirstProbNumShown\firstProbNumShownfalse
4538 \def\tb@insertDecPoint{\ifwithinsoldoc\eqedecPointSoln\else
4539     \eqedecPointMrg\fi}
4540 \newcommand{\eqedsplyOnlyFrst}[2]{\def\thisPart{#2}%
4541     \ifx\thisPart\firstPartLtr\global\firstProbNumShowntrue
4542         \tb@mrgDigitFmt{#1}\tb@insertDecPoint\else
4543         \iffirstProbNumShown\tb@GenProbNum{#1}\else
4544         \global\firstProbNumShowntrue\tb@mrgDigitFmt{#1}%
4545         \tb@insertDecPoint\fi\fi\global\eqeGenProbNumfalse
4546 }

```

`\displayProbNumOnce` If a part is carried over to the next page, it may be necessary to manually force the display of the first digit.

```

\insMidMarg{\displayProbNumOnce}
4547 \newif\ifeqeGenProbNum \eqeGenProbNumfalse
4548 \newcommand{\displayProbNumOnce}{\global\eqeGenProbNumtrue} %
4549 %\def\tb@GenProbNum#1{\ifeqeGenProbNum#1\eqedecPointMrg\else
4550 \def\tb@GenProbNum#1{\ifeqeGenProbNum#1\tb@insertDecPoint\else
4551     \phantom{#1\tb@insertDecPoint}\fi}%
4552 \def\sq@priorhook{\medskip}
Adjustments of spacing between problems \eqexerskip, and the check for enough
room for the next problem.
4553 \def\default@fvsizekip{.1}
The skip prior to the beginning of an exercise
4554 \aboveexskip{Opt}
The skip after the end of an exercise
4555 \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.)

```
4556 \renewcommand\belowexsolnskip{}
```

We remove the `\mark` from this definition, see original definition in `eqexam.def`

```
4557 \renewcommand\exerSolnHeader [3]{%
4558     \ifeqforpaper\else\webnewpage\fi%\par
4559     \noindent\@ifundefined{hypertarget}
4560     {#3}{\hypertarget{#2}{#3}\relax}\solnhspace
4561 }
```

This causes the `eqexam` environment to write the user friendly name of the exam even if there is only one exam.

```
4562 \def\nNumberOfP@rts{0}
```

17.9 Some Convenience/Formatting Commands

`\preExamSolnHead` `\examSolnHeadFmt` `\postExamSolnHead` 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 as the formatting.

```
4563 \renewcommand{\preExamSolnHead}{\medbreak\noindent}
4564 \renewcommand{\examSolnHeadFmt}[1]{\textbf{#1}}
4565 \renewcommand{\postExamSolnHead}{\smallskip}
```

`\wrtChapSolnHead` Writes a chapter heading to the solution file, usage,

```
\wrtChapSolnHead{The New {\eqexam}}
```

```
4566 \newcommand{\wrtChapSolnHead}[1]{%
4567     \writeT@SolnFile{%
4568         \protect\preChapSolnHead
4569         \protect\chapHeadSolnFmt{\protect\ftbFmtChapter{\thechapter}#1}%
4570         \protect\postChapSolnHead
4571 }}
```

`\preChapSolnHead` Same as above, except for chapter headings.

```
\chapHeadSolnFmt 4572 \newcommand{\preChapSolnHead}{\bigbreak\noindent}
\postChapSolnHead 4573 \newcommand{\chapHeadSolnFmt}[1]{\large\textbf{#1}}
4574 \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.

```
4575 \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.

```
4576 \newcommand{\convertChapHeadToChapters}{%
4577     \let\preChapSolnHead\relax
4578     \let\chapHeadSolnFmt\chapter
4579     \let\ftbFmtChapter@gobble
4580     \let\postChapSolnHead\relax
4581 }
```

`\probset` A simple command to announce the problem set.

```
\subsection*{\probset{\thesection}}
```

See also the definition for the `probset` environment below.

```
4582 \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}
```

```
4583 \newcommand{\annotPage}{\protect\annotThePage{\thepage}}
```

```
4584 \newcommand{\annotThePage}[1]{\space(page\protect~#1)}
```

17.10 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.

```
4585 \def\noProbHeader{NPH}
4586 \newenvironment{probset}[2][\probset{\thesection}]{%
4587     \exam[#2]{\autoExamName}\ifx#1\noProbHeader\else
4588     \protected@edef\ftb@tmp@exp{\noexpand\insProbHead{#1}}%
4589     \ftb@tmp@exp\fi}{\endexam}
```

```
4590 \newcounter{exampleno}[section]
```

```
4591 \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}
```

`\examplename` is the label placed on the example.

```
4592 \newcommand{\examplename}{Example}
```

example A simple example environment, based on the `exercise` environment.

```
4593 \newenvironment{example}{%\medskip
4594   \belowexskip{\medskipamount}\aboveexskip{\medskipamount}%
4595   \makeRoomForProb{\@nbaselineskip\baselineskip}{}%
4596   \renewcommand\exlabel{Example}%
4597   \renewcommand\exlabelformat{\textbf{\exlabel~\theexampleno.}}%
4598   \let\eq@fititin\eqfititin
4599   \renewcommand\exrtnlabelformat{\square}%
4600   \def\eqexheader@wrapper{\eqexheader}%
4601   \SolutionsAfter
4602   \begin{exercise}[exampleno]}\end{exercise}}
```

example* An example environment with parts.

```
4603 \newenvironment{example*}{%\medskip
4604   \belowexskip{\medskipamount}\aboveexskip{\medskipamount}%
4605   \makeRoomForProb{\@nbaselineskip\baselineskip}{}%
4606   \renewcommand\exlabel{Example}%
4607   \renewcommand\exlabelformat{\textbf{\exlabel~\theexampleno.}}%
4608   \let\eq@fititin\eqfititin
4609   \renewcommand\exrtnlabelformat{\square}%
4610   \def\eqexheader@wrapper{\eqexheader}%
4611   \SolutionsAfter
4612   \begin{exercise*}[exampleno]}\end{exercise*}}
```

We set some parameters, to values better suited for the option.

```
4613 \setDefaultfvssizeskip{.1}
4614 \nbaselineskip{4}
```

17.11 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`.

```

4615 \newcommand{\ftbInputBookAux}[1]{%
4616   \filename@parse{#1}\@ifundefined{filename@ext}{%
4617     {\def\filename@ext{tex}}{}}%
4618   \xdef\tbBaseName{\filename@base}%
4619   \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.)

```

4620   \let\save@writefile\@writefile
4621   \let\@writefile\@gobbletwo
4622   \let\save@setckpt\@setckpt
4623   \let\@setckpt\@gobbletwo
4624   \makeatletter
4625   \InputIfFileExists{\tbBaseName.aux}{%
4626     \PackageInfo{eqexam}
4627     {Inputting auxiliary files of\MessageBreak\tbSourceFile}%
4628   }{%
4629     \PackageError{eqexam}
4630     {Auxiliary files for \tbSourceFile\space were not found}
4631     {Compile the source file \tbSourceFile\space
4632       three times\MessageBreak%
4633       to create the required auxiliary files.}%
4634   }%
4635   \makeatother

```

The solution files really shouldn't have a label, but if we do we'll save the L^AT_EX 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.

```

4636   \global\let\ftblabel\label
4637   \let\label\@gobble
4638   \let\@writefile\save@writefile
4639   \let\@setckpt\save@setckpt
4640 }
4641 \@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`.

```

4642 \newcommand{\restorelabel}{\global\let\label\ftblabel}
4643 \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.

```

4644 \newcommand{\ftbInputSolnFiles}[1][\tbBaseName.sol]{%
4645   \filename@parse{#1}\@ifundefined{filename@ext}%
4646     {\def\filename@ext{sol}}{}}%
4647   \xdef\tbBaseName{\filename@base}%
4648   \xdef\tbSourceFile{\filename@base.\filename@ext}%
4649   \InputIfFileExists{\tbBaseName.sol}{%
4650     \PackageInfo{eqexam}
4651     {Inputting solutions file \tbBaseName.sol\MessageBreak}}%
4652   }{%
4653   \PackageError{eqexam}
4654     {Solutions file for \tbSourceFile\space was not found}%
4655     {Compile the source files three times}%
4656   }%
4657 }

4658 % End of textbook segment
4659 </textbook>
4660 <*ftbsty>

```

18 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.

```

4661 \NeedsTeXFormat{LaTeX2e}
4662 \ProvidesPackage{fortexbook}
4663 [2012/03/14 v1.0 A fortexbook Package (dps)]
4664 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{eqexam}}
4665 \ProcessOptions
4666 \RequirePackage[%
4667   ftbsolns,fortexbook,usecustomdesign,
4668   forcolorpaper,noseparationrule,usexkv
4669 ]{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.


```

4670 % End of ftbsty segment
4671 </ftbsty>
4672 <*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.

```

4673 \edef\ftbInputEqTextb{\ifeqfortextbook\noexpand
4674   \InputIfFileExists{eqtextb.def}{\fi}
4675 \ftbInputEqTextb

```

19 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.

```

4676 %\IfFileExists{xkeyval}{%
4677 %   \if\eq@usexkeys\eqe@YES\RequirePackage{xkeyval}\else
4678 %   \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^AT_EX 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.

```

4679 \define@boolkey{eqFillin}{underline}[true]{}
4680 \define@key{eqFillin}{u}[]{\KV@eqFillin@underlinetrue}
4681 \define@key{eqFillin}{b}[]{\KV@eqFillin@underlinefalse}
4682 \define@boolkey{eqFillin}{boxed}[true]{}
4683 \define@key{eqFillin}{boxpretext}[]{\def\eq@fillintext{#1}}
4684 \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.

4685 \define@key{eqFillin}{parbox}[\relax]{\def\eq@fillinparbox{#1}}
4686 \let\eq@fillinparbox\@empty
4687 \define@key{eqFillin}{hiddenbox}[]{%
4688   \def\eq@fillinhiddenbox{%
4689     \setlength{\fboxrule}{0pt}\setlength{\fboxsep}{0pt}}
4690 \let\eq@fillinhiddenbox\@empty
4691 \define@boolkey{eqFillin}{enclosesoln}[true]{}
4692 \define@choicekey+{eqFillin}{boxsize}{tiny,scriptsize,footnotesize,%
4693   small,normalsize,large,Large,LARGE,huge,Huge}[normalsize]{%
4694   \def\eq@eqFillin@boxsize{\text{\csname#1\endcsname\strut}}}%
4695 }{\PackageWarning{eqexam}{Bad choice for boxsize, permissible values
4696   are tiny, scriptsize, footnotesize, small, normalsize,
4697   large, Large, LARGE, huge and Huge. Try again}}
4698 \def\eq@eqFillin@boxsize{\text{\normalsize\strut}}

4699 \define@key{eqFillin}{fboxsep}[3pt]{\def\eq@fillin@fboxsep{#1}}
4700 \def\eq@fillin@fboxsep{3pt}
4701 \define@choicekey+{eqFillin}{fontsize}{tiny,scriptsize,footnotesize,%
4702   small,normalsize,large,Large,LARGE,huge,Huge}[normalsize]{%
4703   \def\eq@eqFillin@fontsize{\csname#1\endcsname}%
4704 }{\PackageWarning{eqexam}{Bad choice for boxsize, permissible values
4705   are tiny, scriptsize, footnotesize, small, normalsize,
4706   large, Large, LARGE, huge and Huge. Try again}}
4707 \def\eq@eqFillin@fontsize{\ifmmode\else\normalsize\fi}
4708 \define@key{eqFillin}{color}[\eq@fillinColor]{\edef\eq@fillin@color{#1}}

4709 \define@choicekey+{eqFillin}{align}[\val\nr]%
4710   {l,r,c}[\eq@eqFillin@align@default]{%

```

```

4711 \def\eq@eqFillin@align{#1}%
4712 \ifcase\nr\relax
4713 \def\eqe@align@hfill{}\or
4714 \def\eqe@align@hfill{\hfill}\or
4715 \def\eqe@align@hfill{\hfil}\fi
4716 }{%
4717 \PackageWarning{eqexam}{Bad choice for align, permissible values
4718 are l, r, and c. Try again}}
4719 \let\eqe@align@hfill\relax

```

`defaultalign` is used to change the values of the default macros `\eq@eqFillin@align@default` and `\eqe@align@hfill@default` for the `align` key together.

```

4720 \define@choicekey+{eqFillin}{defaultalign}[\val\nr]{l,r,c}[c]{%
4721 \def\eq@eqFillin@align@default{#1}%
4722 \ifcase\nr\relax
4723 \def\eqe@align@hfill@default{}\or
4724 \def\eqe@align@hfill@default{\hfill}\or
4725 \def\eqe@align@hfill@default{\hfil}\fi
4726 }{%
4727 \PackageWarning{eqexam}{Bad choice for defaultalign, permissible
4728 values are l, r, and c. Try again}}
4729 \setkeys{eqFillin}{defaultalign=c}

```

`\fillInFormatDefault` is the default fill-in format

```

4730 \renewcommand{\fillInFormatDefault}{\normalfont}
4731 \define@key{eqFillin}{format}[\fillInFormatDefault]{%
4732 \def\eq@fillin@format{#1}}
4733 \def\eq@fillin@format{\bfseries}
4734 \def\eqe@fbox@corr#1{#1-2\fboxsep-2\fboxrule}
4735 \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.

```

4736 \@ifundefined{boxed}{%
4737 \def\eq@fillin@defaultbox{\fbox}%
4738 \def\eq@fillin@boxcmd{\fbox}%
4739 }{%
4740 \def\eq@fillin@defaultbox{\boxed}%
4741 \def\eq@fillin@boxcmd{\boxed}%
4742 }

```

`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`.

```

4743 \define@key{eqFillin}{boxcmd}{%
4744 [\eq@fillin@defaultbox]{\def\eq@fillin@boxcmd{#1}}
4745 \@ifundefined{underbar}{%
4746 \def\eq@fillin@defaultul{underline}%
4747 \def\eq@fillin@ulcmd{\underline}%

```

```

4748 }{%
4749     \def\eq@fillin@defaultul{underbar}%
4750     \def\eq@fillin@ulcmd{\underbar}%
4751 }
(2017/01/28) Added custom underline option, the author must define \ulcustom
and specify ulcmd=custom.
4752 \edef\ulcustom{\expandafter\noexpand\eq@fillin@ulcmd}
4753 \edef\temp@exp{\noexpand
4754 \define@choicekey+{eqFillin}{ulcmd}
4755     {underbar,underline,custom}[\eq@fillin@defaultul]}
4756 \temp@exp{\def\@rgi{#1}\def\eq@custom{custom}\ifx\@rgi\eq@custom
4757     \def\eq@fillin@ulcmd{\@nameuse{ulcustom}}\else
4758     \def\eq@fillin@ulcmd{\@nameuse{#1}}\fi
4759 }{\PackageWarning{aeb}{Bad choice for ulcmd, permissible values
4760 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.

```

4761 \define@key{eqFillin}{lift}[-1sp]{\def\eq@fillin@lift{#1}}
4762 \define@boolkey{eqFillin}{autolift}[true]{} %
4763 \define@key{eqFillin}{addtoautolift}[0pt]{}%
4764     \def\eq@fillin@addtoautolift{#1}}

```

The macro \eqe@getiiiOpts 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@getiiiOpts is used with the parbox option of \fillin. The macro \eqe@getiiiOpts has syntax:

```

\eqe@getiiiOpts[pos][height][inner-pos]

```

```

4765 \def\eqe@getiiiOpts{%
4766     \@ifnextchar[%]
4767     \i@eqe@getiiiOpts
4768     {\iii@eqe@getiiiOpts{c}{\relax}[s]}}
Get pos
4769 \def\i@eqe@getiiiOpts[#1]{%
4770     \@ifnextchar[%]
4771     {\ii@eqe@getiiiOpts{#1}}%
4772     {\iii@eqe@getiiiOpts{#1}{\relax}[s]}}
Get height
4773 \def\ii@eqe@getiiiOpts#1[#2]{%
4774     \@ifnextchar[%]
4775     {\iii@eqe@getiiiOpts{#1}{#2}}%
4776     {\iii@eqe@getiiiOpts{#1}{#2}[#1]}}

```

Get *inner-pos*

```
4777 \def\iii@eqe@getiiiOpts#1#2[#3]{%
4778   \def\eqe@opts@argi{#1}%
4779   \def\eqe@opts@argii{#2}%
4780   \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.

```
4781 \newcommand{\setfillinDefaults}[1]{\def\eqe@argi{#1}\ifx\eqe@argi\@empty
4782   \let\eqe@setfillinDefaults\@empty\else
4783   \def\eqe@setfillinDefaults{#1}\fi}
4784 \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.

```
4785 \AtBeginDocument{\@ifpackageloaded{collectbox}
4786   {\let\eqe@cb\eqe@YES}{\let\eqe@cb\eqe@NO}}
4787 \def\fillineolTooLongMsg{\PackageWarning{eqexam}{%
4788   The 'phrase' you are measuring is longer than\MessageBreak
4789   \string\linewidth. Changing length to Opt in hopes you\MessageBreak
4790   can fix things}}
4791 \def\fillineolNoCMsg{\PackageWarning{eqexam}{%
4792   For the \string\fillineol*\space form, the
4793   collectbox package\MessageBreak
4794   is required, but not loaded at this time.\MessageBreak
4795   Switching to \string\fillineol\space in hopes you
4796   load collectbox\MessageBreak
4797   next time or you remove the star-option if not\MessageBreak
4798   really needed}}
4799 \newcommand\fillineol{\@ifstar{\ifx\eqe@cb\eqe@YES
4800   \let\eqe@next\cbfillineol}\else
4801   \def\eqe@next{\fillineolNoCMsg\fillineol@i}\fi}
```

```

4802 \eqe@next}{\fillineol@i}}
4803 \newcommand{\fillineol@i}[1]{\bgroup\setbox\z@\hbox{#1}\fillineol@ii}
4804 \newcommand{\fillineol@ii}[2][]{\%
4805 \setlength\eqetmplengthb{\linewidth-\wd\z}%
4806 \ifdim\eqetmplengthb<0pt\eqetmplengthbOpt\fillineolTooLongMsg\fi
4807 \unhbox\z@\fillin[#1]{\eqetmplengthb}{#2}\egroup}

```

Allow the third argument to have verbatim text as well.

```

4808 \newcommand\cbfillineol{\bgroup\collectboxto{\@tempboxa}
4809 {\setlength{\eqetmplengthb}{\linewidth-\wd\@tempboxa}%
4810 \ifdim\eqetmplengthb<0pt\eqetmplengthbOpt\fillineolTooLongMsg\fi
4811 \unhbox\@tempboxa\cbfillineol@ia}}
4812 \newcommand\cbfillineol@ia[1][]{\def\@rgi{#1}\cbfillineol@ib}
4813 \newcommand\cbfillineol@ib{\collectboxto{\@tempboxa}%
4814 {\cbfillineol@i[\@rgi]}}
4815 \newcommand\cbfillineol@i[1][]{\fillin[#1]{\eqetmplengthb}%
4816 {\unhbox\@tempboxa}\egroup}

```

`\fillin` We finally begin the `\fillin` command.

```
4817 \renewcommand{\fillin}[3][]{\begingroup
```

`\ifsp@expand` is defined in `spdef` package. This is a version if `\ifsp` that expands correctly in an `\edef`.

```
4818 \expandafter\let\expandafter\ifsp\csname ifsp@expand\endcsname
```

Get the keys indicated by the document author.

```

4819 \setkeys{eqFillin}{boxsize,underline=true,boxed=false,%
4820 boxpretext,color,format,enclosesoln=false,fitwidth=false,lift,%
4821 autolift=false,addtoautolift}%
4822 \ifx\eqe@setfillinDefaults\@empty\else
4823 \protected@edef\eq@temp@exp{\noexpand
4824 \setkeys{eqFillin}{\eqe@setfillinDefaults}}%
4825 \eq@temp@exp
4826 \fi
4827 \protected@edef\eq@temp@exp{\noexpand\setkeys{eqFillin}{#1}}%
4828 \eq@temp@exp
4829 \fboxsep\eq@fillin@fboxsep\relax

```

If boxed, we turn underlining off

```
4830 \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`.

```
4831 \ifKV@eqFillin@autolift\def\eq@fillin@lift{-2sp}\fi
```

Get the second parameter.

```

4832 \edef\eqe@argii{#2}\ifx\eqe@argii\@empty\else
4833 \ifdim\eqe@argii=0pt\let\eqe@argii\@empty\fi\fi

```

We reset `\fboxrule` and `\fboxsep` as needed.

```
4834 \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.

```
4835 \ifx\eq@fillinhiddenbox\@empty\else
4836 \KV@eqFillin@boxedtrue\fi
```

If the `parbox` option is taken, we define the third parameter to be wrapped in a `\parbox`.

```
4837 \ifx\eq@fillinparbox\@empty\def\eqe@argiii{#3}\else
```

If `parbox` is specified, we make `align=1` the default.

```
4838 \ifx\eqe@align@hfill\relax
4839 \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.

```
4840 \expandafter\eqe@getiiiOpts\eq@fillinparbox\relax
```

Now, if this is to be `boxed`, we reduce the height of the box (`\boxed` increases the height by $2\text{\fboxrule}+2\text{\fboxrule}$)

```
4841 \ifKV@eqFillin@boxed
```

`\eqe@opts@argii` has a value of `\relax` if the document author did not specify a height for the box.

```
4842 \expandafter\ifx\eqe@opts@argii\relax\else
4843 \edef\eqe@opts@argii{\expandafter
4844 \eqe@fbox@corr\expandafter{\eqe@opts@argii}}\fi
4845 \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.

```
4846 \edef\eqe@parboxOptArgs{[\eqe@opts@argi]%
4847 \expandafter\ifx\eqe@opts@argii\relax\else
4848 [\eqe@opts@argii][\eqe@opts@argiii]\fi}%
```

Now we build the third parameter, `\eqe@argiii`.

```
4849 \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.

```
4850 {\eqe@align@hfill\ifKV@eqFillin@boxed\eq@fillintext\fi#3}}%
4851 \fi
```

If `\eqe@align@hfill` is still equal to `\relax`, give it the default value.

```
4852 \ifx\eqe@align@hfill\relax
4853 \def\eq@eqFillin@align{c}%
4854 \edef\eqe@align@hfill{\eqe@align@hfill@default}\fi
4855 \ifmmode\let\@eqmath\ensuremath\else\let\@eqmath\text\fi
```

We re-calculate the width of the formatted box

```
4856 \ifx\eq@fillinparbox\@empty
4857 \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

```

4858         \settowidth{\eqetmplengthb}{\@eqmath{\eq@eqFillin@fontsize
4859         \eq@fillin@format\ifKV@eqFillin@boxed\eq@fillintext\fi
4860         \eqe@argiii}}%
4861         \ifKV@eqFillin@boxed
4862             \setlength{\eqetmplengthb}{%
4863             \eqetmplengthb+2\fboxsep+2\fboxrule}%
4864         \fi
4865     \else
    If #2 is nonempty, we use this value.
4866         \setlength{\eqetmplengthb}{#2}%
4867     \fi
4868 \else
    parbox option with empty second argument, use \linewidth. for width
4869     \ifx\eqe@argii\@empty
4870         \setlength{\eqetmplengthb}{\linewidth}%
4871         \PackageWarning{eqexam}{Parameter \#2
4872         is empty with parbox option,\MessageBreak
4873         using \string\linewidth\space for width}%
4874     }%
4875 \else
    parbox option with second argument, use #2 for width
4876         \setlength{\eqetmplengthb}{#2}%
4877     \fi
4878 \fi
    Return \ifsp to its default definition.
4879     \expandafter\let\expandafter\ifsp\csname ifsp@default\endcsname
    Save the final calculated width as \eqe@bw.
4880     \edef\eqe@bw{\the\eqetmplengthb}%
    Set the underline option, ...
4881     \ifKV@eqFillin@underline\let\@fillinFmt\eq@fillin@ulcmd
4882     \else\let\@fillinFmt\relax\fi
    however, if parbox is specified, we remove the underlining, if any.
4883     \ifx\eq@fillinparbox\@empty\else
4884         \ifKV@eqFillin@underline\let\@fillinFmt\relax
4885 %         \ifx\@fillinFmt\underbar\let\@fillinFmt\relax
4886         \PackageInfo{eqexam}{Removing underline option, not permissible
4887         \MessageBreak with parbox option}%
4888     \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).


```

4889     \ifeq@proofing
4890         \ifKV@eqFillin@fitwidth
    If the fitwidth option is taken, we measure the width of the box. Ignored when
    the parbox option is used.
4891         \settowidth{\eqetmplengthb}{\@eqmath{\eq@fillin@format
4892             \ifx\eq@fillin@parbox\@empty\ifKV@eqFillin@boxed
4893                 \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.
4894         \ifKV@eqFillin@boxed
4895             \setlength{\eqetmplengthb}{%
4896                 \eqetmplengthb+2\fbboxsep+2\fbboxrule}%
4897         \fi
4898         \edef\eqe@bw{\the\eqetmplengthb}%
4899     \fi
    We build the fill-in field for the case of boxed.
4900     \ifKV@eqFillin@boxed
4901         \ifmmode\let\@eqmath\ensuremath\else\let\@eqmath\text\fi
4902     %         \mbox{\eq@fillin@format\ensuremath{\boxed{%
4903         \setbox\z@\hbox{\@eqmath{\eq@fillin@boxcmd{%
4904     %         \eq@fillin@format\ensuremath{\eq@fillin@boxcmd{%
4905         \eq@eqFillin@boxsize
4906         \@fillinFmt{\eq@eqFillin@fontsize % dpsf0214
4907             \ifKV@eqFillin@boxed
4908                 \edef\eqe@bw{\eqe@fbox@corr{\eqe@bw}}%
4909         \fi
    When the boxed option is taken, we adjust the width of the \makebox to get the
    desired width #2.
4910         \makebox[\eqe@bw][\eq@eqFillin@align]{\strut
4911             \@eqmath{\eq@fillin@format\color{\eq@fillin@color}}%
4912             \ifx\eq@fillin@parbox\@empty
4913                 \eq@fillintext\fi\eqe@argiii}}%
4914         }% end \@fillinFmt
4915     }% end \mbox
4916     \setlength{\@tempdima}{\ht0+\dp0}%
4917     \xdef\fillinTotalHeight{\the\@tempdima}%
4918     \mbox{\unhbox\z@}%
4919     \else
    The content is not to be boxed.
4920         \ifmmode\let\@eqmath\ensuremath\else\let\@eqmath\relax\fi
4921         \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.
4922         \ifdim\eq@fillin@lift=-1sp\relax
4923             \@fillinFmt{\makebox[\eqe@bw][\eq@eqFillin@align]{\strut
4924                 \color{\eq@fillin@color}}%

```

```

4925             \eqmath{\eq@fillin@format\eqe@argiii}}}%
4926         \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.
4927             \ifKV@eqFillin@autolift
4928                 \setbox2=\hbox{\@eqmath{\eq@fillin@format\eqe@argiii}}}%
4929                 \setlength{\@tempdima}{\dp2+\eq@fillin@addtoautolift}%
4930                 \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.
4931             \raisebox{-\eq@fillin@lift}{\@fillinFmt{%
4932                 \makebox[\eqe@bw][\eq@eqFillin@align]{\strut
4933                 \color{\eq@fillin@color}}%
4934                 \raisebox{\eq@fillin@lift}{\@eqmath{\eq@fillin@format
4935                 \eqe@argiii}}}}}%
4936             \fi
4937             \setlength{\@tempdima}{\ht0+\dp0}%
4938             \xdef\fillinTotalHeight{\the\@tempdima}%
4939             \mbox{\unhbox\z0}%
4940         \fi
4941     \else
We begin the case of not \ifeq@proofing, that is, the document author is not
compiling with the answerkey option.
4942         \ifKV@eqFillin@boxed
4943 %             \mbox{\eq@fillin@format\ensuremath{\boxed{%
4944 %             \eq@fillin@format\ensuremath{\eq@fillin@boxcmd{%
4945             \setbox\z0\hbox{\eq@eqFillin@fontsize % dpsf0214
4946             \@eqmath{\eq@fillin@format
4947             \eq@fillin@boxcmd{\eq@eqFillin@boxsize\ifeq@nosolutions
4948             \rlap{\@eqmath{\eq@eqFillin@fontsize % dpsf0214
4949             \eq@fillintext}}}\fi
4950             \ifvspacewithsolns % dpsf0214 \eq@eqFillin@fontsize
4951             \rlap{\@eqmath{\eq@eqFillin@fontsize\eq@fillintext}}}\fi
4952             \@fillinFmt{%
We do a similar thing if proofing is not active (nosolutions is taken).
4953             \makebox[\eqe@fbox@corr{\eqe@bw}]{%
4954                 \ifKV@eqFillin@enclosesoln\phantom
4955                 {\setlength\eqetmplengthb{\eqe@bw-2\fboxsep}%
4956                 \edef\eqe@bw{\the\eqetmplengthb}%
4957                 \@eqmath{\eqe@argiii}}\else
4958                 \strut\hfill\fi
4959             }%
4960             }%end \@fillinFmt
4961             }%end \hbox
4962             \setlength{\@tempdima}{\ht0+\dp0}%
4963             \xdef\fillinTotalHeight{\the\@tempdima}%

```

```

4964         \mbox{\unhbox\z@}%
4965     \else
    This is the case where the field is not boxed.
4966         \setbox\z@\hbox{\eq@eqFillin@fontsize % dpsf0214
4967         \ifdim\eq@fillin@lift=-1sp\relax
4968             \@fillinFmt{\makebox[\eq@bw]{%
4969 %             \phantom{\@eqmath{\eq@argiii}}
4970             \strut\hfil}}%
4971     \else
4972         \ifKV@eqFillin@autolift
4973             \setbox2=\hbox{\@eqmath{\eq@fillin@format
4974             \eq@argiii}}%
4975             \setlength{\@tempdima}%
4976             {\dp2+\eq@fillin@addtoautolift}%
4977             \edef\eq@fillin@lift{\the\@tempdima}\fi
4978             \raisebox[-\eq@fillin@lift]{\@fillinFmt{%
4979             \makebox[\eq@bw]{\raisebox{\eq@fillin@lift}
4980             {\phantom{\@eqmath{\eq@fillin@format
4981             \eq@argiii}}\strut\hfil}}}}%
4982     \fi
4983     }\setlength{\@tempdima}{\ht0+\dp0}%
4984     \xdef\fillinTotalHeight{\the\@tempdima}%
4985     \mbox{\unhbox\z@}%
4986 \fi

```

Online Code. If the quiz environment is defined, and the user has asked for `online` option we build a text field.

```

4987     \@ifundefined{@quiz}{}{%
4988         \if\eq@online\eq@YES\relax
4989             \ifeq@nosolutions
4990                 \ifeq@solutionsafter\else
4991                 \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).
4992                 \stepcounter{@cntfillin}%
4993                 \edef\fieldName{%
4994                     \if\probstar*eqexam.\curr@quiz.fillin.%
4995                     \theeqquestionnoi.part\thepartno.%
4996                     fi\the@cntfillin%
4997                 \else
4998                     eqexam.\curr@quiz.fillin.%
4999                     \theeqquestionnoi.fi\the@cntfillin%
5000                 \fi
5001                 }\ifx\eq@fillinparbox\@empty

```

If the `parbox` option is not taken, we build a text field with height 11bp

```

5002         \raisebox{-1bp}{\makebox[0pt][r]{%

```

```

5003             \textField[\BC{} \presets{\eqe@optsFillIn}]
5004             {\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.
5005             \setlength{\@tempdima}{%
5006             {\eqe@opts@argii+2\fbboxrule+2\fbboxsep}}%
5007             \raisebox{-1bp}{\makebox[0pt][r]{%
5008             \textField[\BC{} \presets{\eqe@optsFillIn}
5009             \Ff{\FfMultiline}]{\fieldName}{#2}
5010             {\fillinTotalHeight}}}\fi
5011             \fi
5012             \fi
5013             \fi
5014             \fi
5015             }%
    (2018/02/02) Removed \space\ignorespaces from \fillin
5016     \fi\endgroup}

```

`\TF` The `\TF` command depends on `\fillin`, so we make the appropriate changes.

```

5017 \renewcommand\TF[2][\defaultTFwidth]{%
5018     \def\eqe@next{\fillin[underline]{#1}{#2}}%
5019     \ifdim\eqe@extralabelsep=0pt\relax\else
5020         \if\probstar*\relax\if\exerwparts@cols0
5021             \def\eqe@next{\makebox[0pt][r]{%
5022             \fillin[underline]{#1}{#2}\ignorespaces}}%
5023         \fi\fi\fi\eqe@next
5024 }

```

This marks the end of the `eqexam` package. `dps`

```
5025 </package>
```

20 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	
<code>\#</code>	662, 666, 4871
<code>\%</code>	1714, 1890, 2001, 2008
<code>\@Ans@sq@f@Defaults</code>	1018, 1080
<code>\@EmailCourseName</code>	917, 943, 944
<code>\@EmailExamName</code>	919, 945, 946
<code>\@EmailInstr</code>	915, 941, 942
<code>\@EmailSubject</code>	921, 947, 948
<code>\@Mi</code>	4189
<code>\@PTs</code>	2645, 2646
<code>\@ServerRetnMsg</code>	923, 949, 950
<code>\@SubmitButtonLabel</code>	925, 933
<code>\@actionsAtPageBreak</code>	2156, 2226
<code>\@altTitle</code>	1334, 1337, 1338
<code>\@argi</code>	2851
<code>\@auxout</code>	1247, 1665, 1702, 1704, 1717, 1731, 1734, 1739, 1750, 1754, 1758, 2113, 2129, 2141, 2288, 2291, 2359, 2361, 2366, 2581, 2618, 2749, 2754, 2763, 2778, 2924
<code>\@bypasseqexamheadingfalse</code>	204
<code>\@bypasseqexamheadingtrue</code>	205
<code>\@checkSpacing</code>	<i>82</i> , 2165, 2177, 2496, 2497, 2521
<code>\@checkSpacingi</code>	2538
<code>\@ckhide</code>	2837
<code>\@date</code>	1333, 1421
<code>\@emitPartTotalsBox</code>	2166, 2168, 2178, 2180
<code>\@emitSepRule</code>	2171, 2172, 2183, 2184, 2191, 2192
<code>\@eqCommentsColor</code>	534, 2097
<code>\@eqCommentsColorBody</code>	536, 2101
<code>\@eqalignfilllinestoleftfalse</code>	3093, 3095
<code>\@eqalignfilllinestolefttrue</code>	3094
<code>\@eqeuseclassmaketitlefalse</code>	202
<code>\@eqeuseclassmaketitletrue</code>	203
<code>\@eqlinedfillerfalse</code>	2030, 3046, 3718
<code>\@eqlinedfillertrue</code>	3043
<code>\@eqmath</code>	4855, 4858, 4891, 4901, 4903, 4911, 4920, 4925, 4928, 4934, 4946, 4948, 4951, 4957, 4969, 4973, 4980
<code>\@equsedimfalse</code>	3092
<code>\@equsedimtrue</code>	3090, 3091
<code>\@evenfoot</code>	472
<code>\@evenhead</code>	470, 4367, 4438
<code>\@evenllx</code>	4265, 4344
<code>\@evenlly</code>	4262, 4263, 4344
<code>\@examAnsKeyLabel</code>	871, 873
<code>\@examEmailLabel</code>	896, 900, 902
<code>\@examNameLabel</code>	873, 880, 883
<code>\@examSIDLabel</code>	887, 890, 892
<code>\@fillinFmt</code>	2410– 2412, 2414, 2417, 4881, 4882, 4884, 4885, 4906, 4914, 4923, 4931, 4952, 4960, 4968, 4978
<code>\@fvsizekip</code>	2228, 2260, 2262, 2263, 2284, 2517
<code>\@gatherTitleText</code>	618, 620, 634, 661, 665
<code>\@getsecondOf</code>	1270, 1271, 1275
<code>\@gobblet@end</code>	2659, 2663
<code>\@gobbloptone</code>	2998
<code>\@headoffset</code>	4366, 4368, 4371, 4374
<code>\@ifl@ter</code>	1119
<code>\@insertPointsBoxPDF</code>	1586, 1602
<code>\@insertTotalsBoxPDF</code>	1593, 1603
<code>\@instructionsColor</code>	532, 2057
<code>\@isitstar</code>	2602, 2603, 2607, 2713–2715, 2719, 2721, 2896, 2908, 2942
<code>\@linkcolor</code>	546, 985, 987, 988, 1049, 1051, 1052, 1150, 1212
<code>\@listii</code>	2822, 2823
<code>\@listii@SAVE</code>	2822, 2823
<code>\@lti</code>	2602, 2604
<code>\@ltii</code>	2603, 2605
<code>\@markerTotalFmt</code>	1720, 1727
<code>\@marktotalvalue</code>	1545, 1553, 2608, 2649, 2789, 2898, 2918, 2938
<code>\@minus</code>	2055, 2095, 2102
<code>\@mparswitchfalse</code>	3917
<code>\@nbaselineskip</code>	2555, 2556, 2598, 2705, 4595, 4605
<code>\@nnotContStrSkip</code>	2981, 2993, 3253
<code>\@nobreakfalse</code>	2486, 2509
<code>\@nolinkcolor</code>	548, 985, 987, 1049, 1051
<code>\@oddf脚</code>	471, 472
<code>\@oddhead</code>	467, 470, 4370, 4373, 4439
<code>\@oddllx</code>	4268, 4342
<code>\@oddlly</code>	4263, 4342
<code>\@onlypreamble</code>	516, 1572, 3932, 4148, 4239, 4430, 4641
<code>\@pdfcreator</code>	379
<code>\@pdfviewparams</code>	394
<code>\@plus</code>	2055, 2095, 2102
<code>\@proofingsymbol</code>	1154, 1252
<code>\@proofingsymbolColor</code>	530, 1154
<code>\@reportpoints</code>	153, 487, 488, 490, 2165, 2177, 2491, 2647

<code>\@reporttotals</code>	495, 500, 501, 505	<code>\afterexamsepcode</code>	2208, 2216, 2326
<code>\@restorepar</code>	3887	<code>\aftergroup</code>	1906, 2161, 2326, 2490, 2631, 2786, 3324
<code>\@rgi</code>	573, 1780, 1784, 2573, 2574, 2590, 2591, 2953, 2958, 3178, 3181, 3428, 3555, 3883, 4756, 4812, 4814	<code>\afterInstrSkip</code>	2046, 2056
<code>\@secondoffive</code>	1271, 3018	<code>\afterlabelhskip</code>	2599, 2711
<code>\@secondoftwo</code>	1270, 3017	<code>\align</code>	<u>4679</u>
<code>\@setckpt</code>	4622, 4623, 4639	<code>align (key)</code>	<u>102</u>
<code>\@spacetobreak</code>	2156–2158	<code>allowcirc4mc (option)</code>	7
<code>\@sssolution</code>	1831	<code>\allowcircmcfalse</code>	113
<code>\@sssolution</code>	1829, 2032	<code>\allowcircmctrue</code>	114
<code>\@startsection</code>	2042, 2054, 2094	<code>\AllowFitItIn</code>	512
<code>\@subject</code>	828, 829	<code>allowrandomize (option)</code>	8
<code>\@tableadinitetrue</code>	2846	<code>\allowZeroTotals</code>	503, 504
<code>\@tempboxa</code>	4808, 4809, 4811, 4813, 4816	<code>\Alph</code>	629, 631, 717, 733, 737, 739, 742, 745, 797, 801, 803, 806, 953, 955
<code>\@templocalversionfalse</code>	676, 813	<code>\alt@SetSolnMargins</code>	305, 306, 310, 312
<code>\@templocalversiontrue</code>	775	<code>\altSetSolnMargins</code>	14, 305
<code>\@undefined</code>	2074, 3016	<code>\altTitle</code>	<u>1330</u>
<code>\@web@title</code>	819, 820	<code>\amtSpaceLeftOnPage</code>	82, 2516
<code>\@whiledim</code>	3306, 3410, 3465	<code>\aNewPage</code>	<u>3005</u>
<code>\@writefile</code>	4620, 4621, 4638	<code>\annotContStr</code>	<u>2974</u>
<code>\{</code>	1890	<code>\annotPage</code>	<u>4583</u>
<code>\}</code>	1896	<code>\annotThePage</code>	4583, 4584
<code>_</code>	430, 432, 434, 438, 840, 880, 890, 900, 1530, 2047, 2090, 2233, 2373, 2403, 2452, 4070, 4071, 4077, 4078, 4083, 4084	<code>\ANS</code>	<u>3954</u>
A		<code>\Ans@choice</code>	981, 1016, 1045, 1078, 1251
<code>\A</code>	933, 982, 1016, 1046, 1078	<code>\Ans@ck@sq@f</code>	1108
<code>\AA</code>	1597	<code>\Ans@ck@sq@l</code>	1107
<code>\AACalculate</code>	1597	<code>\Ans@proofing</code>	989, 1022, 1053, 1084, 1214, 1220, 1249
<code>\aboveexskip</code>	2561, 4554, 4594, 4604	<code>\Ans@r@l@Defaults</code>	957, 983, 1047
<code>\abovesqskip</code>	2266	<code>\Ans@sq@f</code>	1106
<code>\acp@mpah</code>	2928, 2943	<code>\Ans@sq@f@Actions</code>	1018, 1080
<code>\acp@mpph</code>	2929, 2944	<code>\Ans@sq@f@driver</code>	1002, 1005, 1064, 1067, 1216
<code>\acvspace</code>	<u>2974</u>	<code>\Ans@sq@l</code>	1105
<code>\addpenalty</code>	2486, 2509	<code>\Ans@sq@l@driver</code>	1212
<code>\addtoautolift</code>	<u>156</u>	<code>\ANScolor</code>	<u>4067</u>
<code>\addtocounter</code>	668, 1551, 1662, 1706, 1707, 2213, 2222, 2223, 2607, 2616, 2649, 2650, 2773, 2868, 2876, 2897, 2917, 2932, 2938, 2972	<code>\ANSFmt</code>	4022, <u>4068</u>
<code>\AddToShipoutPicture</code>	4476	<code>\AnswerKey</code>	82, 84
<code>\aeb@sectioncolor</code>	554	<code>answerkey (option)</code>	6
<code>\aebc@end</code>	365, 366, 682, 701, 797	<code>\answerkeyfalse</code>	70, 342, 517
<code>\aebshowgraylettersfalse</code>	190	<code>\answerkeyifSave</code>	346
<code>\aebshowgrayletterstrue</code>	192	<code>\answerkeytrue</code>	84, 185, 281, 3904
<code>afterChapSolns (environment)</code>	<u>4394</u>	<code>\answers</code>	287
<code>\afterCommentSkip</code>	2089, 2096, 2098, 2099	<code>\answers@sq</code>	287
		<code>\applyleadinfix</code>	2683, 2687
		<code>\applyparfixes</code>	87, 2685
		<code>\applyparfixesp</code>	2570, 2687
		<code>\ARG</code>	333, 335
		<code>\arg@i</code>	17
		<code>\argi</code>	1653, 1657, 1661, 1663, 2772, 2777, 2857, 2870, 2889, 2910, 2914, 2917, 4053, 4054
		<code>\argii</code>	1653, 1654, 1658, 1660, 1664, 2857, 2861, 2893

`\AtBeginDocument` 175, 371, 569,
 572, 685, 1126, 1291, 1544, 1699, 4474, 4519, 4785
`\AtEndDocument` 372, 1701
`\AtEndOfPackage` 20, 29, 33–39, 44, 46,
 53, 54, 69, 101, 112, 135, 183, 192, 263, 266, 751
`\author` 816
`\authorColor` 530
`\auto` 2657, 2664, 2723, 2753
`\autocalcparts` 95, 2726, 2930
`\autoExamName` 2264, 2297, 4587
`\autoInsSolns` 146, 4518
`\autolift` 156

B

`\baselineskip` 1809, 2598,
 2680, 2705, 3562, 3567, 3737, 3739, 4595, 4605
`\Bbox` 1148, 1173, 1176, 1177,
 1181–1183, 1190, 1192, 1193, 1196, 1197, 1218
`\BC` 910, 959, 1101, 1590, 1596, 2432, 5003, 5008
`\beforeCommentSkip` 2088, 2095, 2102
`\BeforeIncludedComment` 4413
`\beforeInstrSkip` 2045, 2055, 2062
`\belowexskip` 2561, 4555, 4594, 4604
`\belowexsolnskip` 2565, 4556
`bgcolor` (key) 103
`bgonly*` (key) 103
`\bGrpANS` 2743, 2744, 3954
`\bIFFalseWrtSolns` 2274, 2283
`\bigbreak` 4572
`\bigskip` 1396, 2191
`\bigskipamount` 1440, 1483, 1510
`\bItemInsert` 3886
`\boPage` 3008, 3014, 3017–3019, 3021, 3039
`\bopCoverPageText` 3009, 3039
`\bopText` 3008, 3010
`\botmark` 1656
`\box` 3546, 3671, 3687, 3690, 3840, 3841, 3881
`\boxcmd` 4736
`\boxed` 4679, 4740, 4741, 4902, 4943
`\boxpretext` 4679
`\boxsize` 4679
`\bpartsmrk` 2797, 4135, 4140, 4143
`\bProbInsert` 3883
`\bslash\textsluu{\textttuu{Mrk1}}End` 60
`\bslash\textsluu{\textttuu{Mrk1}}nQs` 60
`\bslash\textsluu{\textttuu{Mrk1}}Start` 60
`\btwnExamSkip` 319, 328
`\btwnExamSkipAmt` 318
`\bWebCustomize` 248

C

`\c@eq@count` 3019, 3020
`\c@partno` 2971
`\c@rryoverFmt` 4278, 4279
`\CA` 933
`\calcFromMarkers` 1720
`\calcQsBtwnMarkers` 1729
`\calcTextField` 1590, 1596
`\cancelleadinfix` 2684, 2691, 2692
`\cancelparfixes` 87, 2688, 2693
`\cancelparfixesp` 2571, 2572, 2690, 2692
`carryoverFmt` (environment) 4277
`\cbfillineol` 4800, 4808
`\cbfillineol@i` 4814, 4815
`\cbfillineol@ia` 4811, 4812
`\cbfillineol@ib` 4812, 4813
`\centering` 1433, 1439, 1445
`\centerWidget` 1140, 1170, 1217
`cfg` (option) 5
`\cfooteqe` 1309
`\chapHeadSolnFmt` 4569, 4572
`\chapter` 4578
`\chapterexercisefalse` 4377
`\chapterexercisetrue` 4421
`\chaptername` 4575
`\chaptersolutions` 4377, 4393
`\chead` 1298
`\cheadeqe` 1292
`\cheadSol` 1316
`\check@Box` 1044, 1076
`\chkmarginboxwidth` 4240, 4475
`\chgToNoSolns` 517
`\circ@Glyph` 1166, 1169, 1171
`\circProofingForCirc` 1170, 1187
`\ckbox@Color` 574, 575, 1146
`\ckboxColor` 22, 564, 573, 575
`\ckcir@Color` 577, 578, 1167, 1169
`\ckcirColor` 22, 565, 576, 578
`\cleaders` 3073, 3074, 3078, 3084, 3087
`\clearBotMargin` 135, 4178, 4464
`\clearpage` 1706
`\clearTopMargin` 135, 4170, 4464
`\clubpenalty` 3729, 3792
`\cngMargHeadColorTo` 4208
`\col@number` 2489, 2503
`\collectboxto` 4808, 4813
`\color` 575, 578, 1432, 1438, 1444,
 2101, 2415, 4198, 4200, 4679, 4911, 4924, 4933
`color` (key) 102
`\color@begingroup` 4172, 4174, 4310, 4325

<code>\color@endbox</code>	3618	<code>\cpUsefbox</code>	<u>1453</u>
<code>\color@endgroup</code>	4173, 4175, 4318, 4328	<code>\cqqsfalse</code>	300, 303
<code>\color@vbox</code>	3609	<code>\csarg</code>	331, 332, 334, 680, 682, 694, 695, 699, 701, 733, 797, 1261, 1262, 1273, 1274, 1279, 1479, 1508, 1522, 1529, 1665, 1702, 1704, 1717, 1731, 1733, 1734, 1737, 1739, 1749, 1750, 1753, 1755, 1757, 1758, 2113, 2118, 2120, 2122, 2125, 2129, 2141, 2209, 2288, 2291, 2305, 2312, 2315, 2335, 2346, 2347, 2351, 2356, 2360, 2361, 2366, 2581, 2619, 2694, 2748, 2749, 2754, 2764, 2778, 2812, 2847, 2868, 2924, 2933, 3013, 3038, 4401, 4406, 4411, 4413, 4423, 4425
<code>\columnbreak</code>	2489, 2503	<code>\CT@arc@</code>	267
<code>\columnseprule</code>	417, 4437, 4454, 4461	<code>\CT@cell@color</code>	266
<code>\comment</code>	411	<code>\ctrl@exp@exercise</code>	2623, 2625
<code>\CommentCutFile</code>	4396, 4397, 4404	<code>\curr@quiz</code>	973, 975, 1008, 1011, 1037, 1040, 1070, 1073, 1095, 1098, 1588, 1595, 1598, 2076, 2078, 2081, 2085, 2298, 2425, 2429, 4994, 4998
<code>\CommentEndDef</code>	4408, 4415	<code>\CurrentOption</code>	75, 79, 226, 4664
<code>\CommentStream</code>	3584, 4406, 4411	<code>\currExamName</code>	2196, 2199, 2200, 2276, 2285–2287, 2290, 2292
<code>\cont@notfalse</code>	2997, 2999	<code>\currhideopt</code>	1791, 1810, 1875, 1880, 1914, 1917, 2836
<code>\cont@nottrue</code>	3001	<code>\currProbHead</code>	4219, 4220
<code>\convertChapHeadToChapters</code>	<u>4576</u>	<code>\customNaming</code>	<u>49</u> , 1403
<code>\copy</code>	3073, 3629, 3642, 3655		
<code>\copyrightyears</code>	837		
<code>coverpage (option)</code>	6		
<code>\coverpage@subject</code>	1370–1373		
<code>\coverpageCID</code>	1385, 1393		
<code>\coverpagesubject</code>	<u>1370</u>		
<code>\coverpageSubjectFmt</code>	<u>1370</u>		
<code>coverpagesumry (option)</code>	6		
<code>\coverpageTitleFmt</code>	<u>1370</u>		
<code>\coverpageUniversityFmt</code>	<u>1370</u>		
<code>\cp@@CID</code>	1383, 1384, 1388, 1391, 1419		
<code>\cp@CID</code>	1388, 1419		
<code>\cp@EnclNameAndID</code>	1408, 1409, 1417		
<code>\cp@HghtFrstLn</code>	1410–1412		
<code>\cp@IsertNaming</code>	1465, 1486		
<code>\cp@setHghtFrstLn</code>	1412, 1418		
<code>\cp@SetNameAndIDWidth</code>	1406, 1414–1417		
<code>\cp@SetSumryWidth</code>	1459, 1473, 1474, 1502, 1503		
<code>\cp@sumryWdth</code>	1473–1475, 1502–1504		
<code>\cp@Usefbox</code>	1461, 1463, 1471, 1472, 1495, 1500, 1501, 1540		
<code>\cpCID</code>	1383		
<code>\cpCID@argi</code>	1385, 1387, 1389, 1391, 1416		
<code>\cpEnclNameAndID</code>	1408		
<code>\cpNameAndID</code>	1413, 1448		
<code>\cpNofbox</code>	<u>1453</u>		
<code>\cprulelength</code>	1464, 1477, 1506		
<code>\cpSetCIDWidth</code>	1386, 1392		
<code>\cpSetHghtFrstLn</code>	1410		
<code>\cpSetNameAndIDWidth</code>	1404, 1407		
<code>\cpSetSumryWidth</code>	<u>1453</u>		
<code>\cpSumrybypages</code>	51, 62, 1499		
<code>\cpSumrybyparts</code>	50, 61, 1470		
<code>\cpSumryGrade</code>	<u>1453</u> , 1493, 1538		
<code>\cpSumryHeader</code>	<u>1453</u> , 1483, 1510		
<code>\cpSumryPage</code>	<u>1453</u> , 1526		
<code>\cpSumryPts</code> .	<u>1453</u> , 1488, 1492, 1529, 1531, 1532, 1537		
<code>\cpSumryTotal</code>	<u>1453</u> , 1490, 1535		
		D	
		<code>\D@Num</code>	2661, 2662, 2765, 2767, 2771
		<code>\date</code>	<u>854</u>
		<code>\day</code>	866
		<code>\dbMrk</code>	3423, 3481, 3482, 3509
		<code>\DeclareOption</code>	4664
		<code>\DeclareOptionX</code>	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
		<code>\default@fvsizekip</code>	2259, 2261, 2262, 4553
		<code>\default@nbaselineskip</code>	83, 2553, 2556
		<code>\default@align</code>	155
		<code>\default@instructions</code>	<u>2043</u>
		<code>\default@soln@indent</code>	4109–4111
		<code>\default@TFwidth</code>	2441, 2442, 5017
		<code>\define@boolkey</code>	3117, 3159, 3165, 3897, 3898, 3914– 3916, 3918, 3919, 4679, 4682, 4691, 4735, 4762
		<code>\define@choicekey</code> ..	58, 3103, 3121, 3138, 3148, 3154, 3161, 3167, 4692, 4701, 4709, 4720, 4754
		<code>\define@color</code>	4064, 4067, 4195, 4210
		<code>\depth@to@date</code>	3302, 3307, 3317, 3327

<code>\dimen</code> ..	3329–3331, 3401–3405, 3407, 3410, 3412, 3415–3418, 3440–3445, 3461, 3463–3465, 3468, 3469, 3471, 3493–3495, 3501, 3505, 4331–4334	<code>\endanswers</code>	288
<code>\ding</code>	286, 1207, 1210, 2708	<code>\endanswers@sq</code>	288
<code>\displ@yPointsfalse</code>	524	<code>\endcomment</code>	412
<code>\displ@yPointstrue</code>	522, 523	<code>\enddlcomment</code>	412, 414
<code>\displayPointsOff</code>	20, 524	<code>\endeqEXt</code>	3912, 4513
<code>\displayPointsOn</code>	20, 523	<code>\endeqSavedComment</code>	414, 3811
<code>\displayProbNumOnce</code>	4547	<code>\endexam</code>	4589
<code>\displayworkareafalse</code>	85, 118, 180, 186, 194, 196, 283, 3723	<code>\endexamtotal@color</code>	550, 2166, 2178
<code>\displayworkareaifSave</code>	348	<code>\endexercise@parts@tabular</code>	1548
<code>\displayworkareaOff</code>	196	<code>\endgraf</code>	1985, 2485
<code>\displayworkareaOn</code>	195	<code>\endinput</code>	248, 4678
<code>\displayworkareatrue</code> ...	98, 171, 195, 343, 515, 519	<code>\endlongTitleText</code>	597, 637, 647, 660, 663
<code>\divide</code>	770, 789, 2350, 3439, 3443	<code>\endmanswers</code>	290
<code>\dlcomment</code>	411, 413	<code>\endmanswers@sq</code>	290
<code>\Do</code>	2658, 2662	<code>\endminipage</code>	2034, 2035, 2037, 3772, 3815
<code>\documentclass</code>	273	<code>\endpanel</code>	1894
<code>\DoNotFitItIn</code>	513, 4515	<code>\endparts</code>	1548
<code>\DoNotRecordThisExamfalse</code>	2272, 2330	<code>\endproblem</code>	2627
<code>\DoNotRecordThisExamtrue</code>	2273	<code>\endproblem*</code>	89
<code>\DoNum</code>	2460, 2462, 2466, 2706	<code>\endshortquiz</code>	2325
<code>\dpPtBox</code>	1646, 2151, 2529, 2544	<code>\endshortTitleText</code>	597, 651, 666
<code>\duedate</code>	854, 855, 856	<code>\endsolution</code>	1830, 1832
<code>\DV</code>	941, 943, 945, 947, 949	<code>\endsplitsolution</code>	1835
<code>dvipdfm</code> (option)	11	<code>\endtitleMarker</code>	617, 622, 637, 660
<code>dvipdfmx</code> (option)	11	<code>\endverbatimwrite</code>	1895, 3769, 3813
<code>dvips</code> (option)	11	<code>\ensuremath</code> .	4855, 4901, 4902, 4904, 4920, 4943, 4944
<code>dvipsone</code> (option)	11	environments:	
		<code>afterChapSolns</code>	4394
		<code>carryoverFmt</code>	4277
		<code>eq@numparts</code>	19
		<code>eqComments</code>	2088
		<code>eqeList</code>	4052
		<code>eqepartsquestions</code>	4497
		<code>eqpointsofar</code>	19
		<code>eqpointsthispag</code>	19
		<code>exam</code>	2259
		<code>example</code>	4593
		<code>example*</code>	4603
		<code>fullwidthtext</code>	4441
		<code>instructions</code>	2045
		<code>lsol</code>	4166
		<code>priorworkarea</code>	3791
		<code>problem</code>	2558
		<code>problem*</code>	2668
		<code>probs</code>	4585
		<code>splitsolution</code>	63, 1815
		<code>ssol</code>	4167
		<code>vadjForSolnInBx</code>	3870
		<code>workarea</code>	1774
		<code>\eoeTotalOff</code>	508
E			
<code>\eachLabel</code>	1563		
<code>\ef@Bbox</code>	1176, 1181, 1192, 1196		
<code>\egroup</code>	885, 894, 904, 1165, 1387, 1391, 1406, 1459, 1777, 2034, 2035, 2174, 2229, 2490, 2513, 2516, 2537, 2552, 2582, 2775, 3292, 3420, 3546, 3874, 4319, 4339, 4807, 4816		
<code>\eGrpANS</code>	3954		
<code>\eIFFalseWrtSolns</code>	2275, 2329		
<code>\email</code>	816		
<code>email</code> (option)	7		
<code>\EmailCourseName</code>	33, 917, 918		
<code>\EmailExamName</code>	34, 919, 920		
<code>\EmailSubject</code>	34, 921, 922		
<code>\emitMessageNearBottom</code>	2502, 3099		
<code>\encloseProblemsWith</code>	526		
<code>\enclosesoln</code>	4679		
<code>\end@ssSolution</code>	1832		
<code>\end@sssolution</code>	1830, 2036		

<code>\eoeTotalOn</code>	509	<code>\eq@hspanner</code>	990, 1023, 1054, 1085
<code>\epartsmrk</code>	4136, 4142, 4144	<code>\eq@initializeContAnnot</code>	2810
<code>\eProbInsert</code>	121, 3887	<code>\eq@initLoop</code>	2965, 2969, 2970
<code>\eq@@emitMessageNearBottom</code>	2507, 2508	<code>\eq@insertContAnnot</code>	3000, 3002, 3004, 3310
<code>\eq@@writeexheaderlist</code>	2840	<code>\eq@insertverticalspace</code>	69, 145, 146, 151, 1769, 1807, 2029, 2422, 3707, 3717, 3720, 3784, 3823, 4991
<code>\eq@argi</code>	2052, 2058, 2060	<code>\eq@item</code>	2835, 2837
<code>\eq@b@ddCodeSpecial</code>	3560, 3732	<code>\eq@item@latex</code>	2990
<code>\eq@cfoot</code>	1311, 1315	<code>\eq@l@check@driver</code>	982, 1046
<code>\eq@chead</code>	1294, 1303	<code>\eq@l@l</code>	969, 1033
<code>\eq@cheadSol</code>	1318, 1323	<code>\eq@leadin@item</code>	2835, 2839
<code>\eq@ckglobalhide</code>	2839	<code>\eq@lfoot</code>	1309, 1315
<code>\eq@ColorPackage</code>	158, 198, 200, 201, 209, 211, 213, 216, 219, 221, 224, 226, 265	<code>\eq@lhead</code>	1292, 1303
<code>\eq@currProbStartPage</code>	2513, 2983, 2984, 2996	<code>\eq@lheadSol</code>	1316, 1323
<code>\eq@custom</code>	4756	<code>\eq@linesXPgs</code>	3334, 3339
<code>\eq@default@Instructions</code>	2043, 2049	<code>\eq@listType</code>	966, 994, 1029, 1056
<code>\eq@e@ddCodeSpecial</code>	3733, 3769, 3774	<code>\eq@lw@f</code>	997, 1059
<code>\eq@Email</code>	900, 1420	<code>\eq@lw@l</code>	968, 1032
<code>\eq@emitMessageNearBottom</code>	2504–2506	<code>\eq@max@selected</code>	130, 131
<code>\eq@eqFillin@align</code> 4711, 4839, 4853, 4910, 4923, 4932		<code>\eq@NO</code>	3717, 3784
<code>\eq@eqFillin@align@default</code>	4710, 4721	<code>\eq@nolinkfalse</code>	238, 343, 2836
<code>\eq@eqFillin@boxsize</code>	4694, 4698, 4905, 4947	<code>\eq@nolinkifSave</code>	347
<code>\eq@eqFillin@fontsize</code>	4703, 4707, 4858, 4906, 4921, 4945, 4948, 4950, 4951, 4966	<code>\eq@nolinktrue</code>	170, 514, 2841, 2842
<code>\eq@ExamLastPage</code>	43, 1255, 1319	<code>\eq@nosolutionsfalse</code>	236, 343
<code>\eq@ExamName</code>	879, 1297, 1335, 1418	<code>\eq@nosolutionsifSave</code>	348
<code>\eq@extralabelsep</code>	2444, 2454, 5019	<code>\eq@nosolutionstrue</code>	170, 514, 519
<code>\eq@fillin@addtoautolift</code>	4764, 4929, 4976	<code>\eq@nosummarytotals</code>	55, 148, 496, 506, 507, 2050, 2051, 2058, 2060
<code>\eq@fillin@boxcmd</code>	4738, 4741, 4744, 4903, 4904, 4944, 4947	<code>\eq@nototals</code>	147
<code>\eq@fillin@color</code>	4708, 4911, 4924, 4933	<code>\eq@nSelectedVersion</code> ..	718, 721, 767, 769, 771, 784
<code>\eq@fillin@defaultbox</code>	4737, 4740, 4744	<code>eq@numparts</code> (environment)	19
<code>\eq@fillin@defaulttul</code>	4746, 4749, 4755	<code>\eq@nVersions</code> ..	593, 596, 623, 633, 714, 721, 726, 748
<code>\eq@fillin@fboxsep</code>	4699, 4700, 4829	<code>\eq@online</code>	119, 138, 139, 351, 396, 907, 978, 1000, 1043, 1062, 1111, 1172, 1189, 2419, 4988
<code>\eq@fillin@format</code>	4732, 4733, 4859, 4891, 4902, 4904, 4911, 4925, 4928, 4934, 4943, 4944, 4946, 4973, 4980	<code>\eq@parttotals</code>	46, 149, 508, 509, 2149, 2155
<code>\eq@fillin@lift</code>	4761, 4831, 4922, 4930, 4931, 4934, 4967, 4977–4979	<code>\eq@pointbox</code>	483, 1640, 1642, 1644, 1646, 1647, 3629, 3630, 3632, 3642, 3644, 3655, 3670, 3686, 3690
<code>\eq@fillin@ulcmd</code> 4747, 4750, 4752, 4757, 4758, 4881		<code>\eq@pointboxtotalheight</code> 484, 1642–1645, 2150– 2153, 2160, 2526, 2528, 2529, 2531, 2543–2546	
<code>\eq@fillinColor</code>	552, 4708	<code>\eq@proofingfalse</code>	189, 237, 342, 518
<code>\eq@fillinhiddenbox</code>	4688, 4690, 4834, 4835	<code>\eq@proofingifSave</code>	346
<code>\eq@fillinparbox</code>	4685, 4686, 4837, 4840, 4856, 4883, 4892, 4912, 5001	<code>\eq@proofingtrue</code>	85, 187, 188, 281, 3904
<code>\eq@fillintext</code>	4683, 4684, 4850, 4859, 4893, 4913, 4949, 4951	<code>\eq@protect</code>	982, 1016, 1046, 1078
<code>\eq@fititin</code>	512, 513, 1152, 4598, 4608	<code>\eq@Radio@driver</code>	993, 1017, 1079
<code>\eq@globalshowsolutionsfalse</code>	235	<code>\eq@RadioCheck@driver</code>	993
<code>\eq@globalshowsolutionstrue</code>	173, 175	<code>\eq@ren@selected</code>	130, 133
<code>\eq@hidesolutionfalse</code>	234, 2836	<code>\eq@renditionOptions</code>	129, 132, 134, 752
		<code>\eq@replaceToken</code>	673

<code>\eq@rfoot</code>	1313, 1315	<code>\eqdashruleVfill</code>	3087, 3089
<code>\eq@rgi</code>	576	<code>\eqdotrulefill</code>	3072, 3076
<code>\eq@rhead</code>	1296, 1303	<code>\eqdotruleVfill</code>	3084, 3086
<code>\eq@rheadSol</code>	1320, 1323	<code>\eqDriverName</code>	164,
<code>\eq@selectedItem</code>	2964, 2968	165, 208, 210, 212, 215, 218, 220, 223, 228, 393	
<code>\eq@selectedVersion</code>	598,	<code>\eqDvipsone</code>	391, 393
607, 671, 672, 710, 711, 717, 724, 2630, 2785		<code>\eqe@holdTemp</code>	690, 691, 762–764, 777
<code>\eq@selectedVersion@save</code>	598, 599, 607, 608, 648, 650, 652, 654	<code>\eqe@leadinitem</code>	2796, 2808
<code>\eq@setWidgetProps</code>	982, 1017, 1046, 1079	<code>\eqe@numLinesL</code>	3106, 3243
<code>\eq@shortTitlesFix</code>	624, 655	<code>\eqe@numLinesR</code>	3107, 3248
<code>\eq@shortVersionAtext</code>	582, 585	<code>\eqe@offVersion</code>	688, 689, 707, 768
<code>\eq@shortVersionBtext</code>	582, 586	<code>\eqe@onVersion</code>	686, 687, 706, 761, 763
<code>\eq@SID</code>	890, 1419	<code>\eqe@SumryVert</code>	1396, 1398, 1471, 1500
<code>\eq@solutionsafterfalse</code>	229, 232, 342, 518	<code>\eqe@absorbTokensLong</code>	638, 647
<code>\eq@solutionsafterifSave</code>	347	<code>\eqe@absorbTokensShort</code>	640, 651
<code>\eq@solutionsaftertrue</code>	85, 179, 231, 282	<code>\eqe@adjForSepRule</code>	2149, 2171, 2183, 2186
<code>\eq@solutionshook</code>	1324, 4524, 4525	<code>\eqe@afterexamsepcode</code>	2154, 2216
<code>\eq@sqlabel</code>	2404	<code>\eqe@align@hfill</code>	4713–4715, 4719, 4838, 4839, 4850, 4852, 4854
<code>\eq@sqllabel</code>	1222, 1223, 2403	<code>\eqe@align@hfill@default</code>	4723–4725, 4854
<code>\eq@sqlrtnlabel</code>	2387, 2391, 2393	<code>\eqe@allow</code>	1678, 1680, 1682, 1690
<code>\eq@sqlsecurunhead</code>	2402	<code>\eqe@cargi</code>	1730, 1742, 1752, 1929, 2069, 2070, 2093,
<code>\eq@sqlsectitle</code>	2401	2098, 2099, 2309, 2310, 2364, 2365, 2664, 4781	
<code>\eq@temp@exp</code>	4823, 4825, 4827, 4828	<code>\eqe@cargii</code>	621, 622, 2664, 2665, 4832, 4833, 4857, 4869
<code>\eq@tmp</code>	2303, 2304	<code>\eqe@cargiii</code>	4837, 4849, 4860, 4893,
<code>\eq@tmpbox</code>	969, 998, 1033, 1060, 1136	4913, 4925, 4928, 4935, 4957, 4969, 4974, 4981	
<code>\eq@tmpdima</code>	969, 981, 988, 989,	<code>\eqe@auto@chk@drivers</code>	115, 118, 120, 121, 123
998, 1033, 1045, 1052, 1053, 1060, 1137, 1213,		<code>\eqe@Bbox</code>	39, 1218
1214, 1902, 1903, 1946, 1948–1950, 2814–2817		<code>\eqe@BboxCirc</code>	1169, 1173, 1183, 1190
<code>\eq@tmplength</code>	968, 971, 997, 999,	<code>\eqe@BboxRect</code>	1146, 1148, 1177, 1182, 1193, 1197
1032, 1035, 1059, 1061, 2818–2821, 2985, 2986		<code>\eqe@BGColor</code>	3171, 3172, 3487, 3488
<code>\eq@tmplengthA</code>	481, 880, 882, 890, 891,	<code>\eqe@bParts</code>	2852–2854
900, 901, 1388, 1390, 2452–2454, 3351, 3357,		<code>\eqe@bw</code>	4849, 4880, 4898, 4908,
3363, 3372, 3376, 3380, 3386, 3389, 3407, 3417		4910, 4923, 4932, 4953, 4955, 4956, 4968, 4979	
<code>\eq@tmplengthB</code>	482, 3625, 3634, 3646	<code>\eqe@calc@percent</code>	2340, <u>2344</u>
<code>\eq@TW</code>	1611, 1612, 1614	<code>\eqe@cb</code>	4786, 4799
<code>\eq@usexkeys</code>	127, 128, 4677	<code>\eqe@chkZeroTotals</code>	1678, 1681, 1689
<code>\eq@VersionAtext</code>	581, 583	<code>\eqe@cirDiam</code>	1162, 1167
<code>\eq@VersionBtext</code>	581, 584	<code>\eqe@cirRadius</code>	1164, 1166
<code>\eq@WriteLineColor</code>	3061, 3067, 3076, 3080, 3116, 3498, 3508	<code>\eqe@color@opt</code>	198, 200
<code>\eq@YES</code>	3349, 3370, 3587, 3707, 3720, 3771, 3814	<code>\eqe@commentChkMsg</code>	367, 371
<code>\eqbothmargins</code>	491, 1650	<code>\eqe@contTitleText</code>	603, 612, 615
<code>\eqcenterWidget</code>	1139, 1142–1144, 1171, 1218	<code>\eqe@coverPageNaming</code>	1400–1402, 1466
<code>eqComments</code> (environment)	<u>2088</u>	<code>\eqe@cprulelength</code>	1478, 1486, 1491, 1494, 1507, 1527, 1536, 1539
<code>\eqCommentsColor</code>	<u>530</u>	<code>\eqe@csarg</code>	242, 243
<code>\eqCommentsColorBody</code>	536, 537, 563	<code>\eqe@decPointMrg</code>	4037, 4082, 4085, 4089, 4092, 4539, 4549
<code>\eqcustomdesignfalse</code>	6, 277	<code>\eqe@decPointPrb</code>	429,
<code>\eqcustomdesigntrue</code>	5	430, 436, 2375, 2379, 2382, 3295, 4069, 4072, 4113	
<code>\eqdashrulefill</code>	3077, 3080		

`\eqe@defNumRefii` 43, 1269, 1286, 1287
`\eqe@DoNum` 2460, 2462, 2706
`\eqe@DoWarning` 2458, 2460, 2461, 2472, 2473
`\eqe@dpsepPrb` 430
`\eqe@drivernum`
 208, 210, 212, 215, 217, 220, 223, 227, 374
`\eqe@emnb` 2503, 2505, 2512
`\eqe@eqOpts` 166, 167, 384
`\eqe@fbox@corr` 4734, 4844, 4908, 4953
`\eqe@fillLinesNumFmt` 3173, 3245, 3247, 3249
`\eqe@fillwidth` 3277, 3292, 3314, 3435,
 3463, 3478, 3484, 3489, 3491, 3511, 3516, 3613
`\eqe@flextendedfalse` 106, 111, 2031, 3716
`\eqe@flextendedtrue` 102, 3705
`\eqe@flnexti` 3734, 3755, 3789, 3808, 3810, 3817
`\eqe@flnextii` 3811, 3813, 3819
`\eqe@flnosolnsfalse` 3549, 3551, 3695
`\eqe@flnosolnstrue` 3550, 3694
`\eqe@Four` 42, 2729
`\eqe@fpmrk` 2076, 2636, 2949, 2950
`\eqe@getiiiOpts` 4765, 4840
`\eqe@gobbletoend` 2558, 2603, 2714
`\eqe@gobnxtpar` 2683, 2684, 2740
`\eqe@grabarg` 2559, 2602, 2713
`\eqe@h` 43, 2574
`\eqe@HOfVrule` 3449, 3458, 3619, 3660, 3664,
 3665, 3668, 3669, 3675, 3679, 3680, 3683, 3684
`\eqe@HOfVruleFrstRow` .. 3453, 3455, 3489, 3491, 3514
`\eqe@holdTemp` 691, 804
`\eqe@hspannerMrg`
 4000, 4013, 4029, 4056, 4084, 4087, 4092
`\eqe@hspannerPrb` 433, 434, 436,
 1636, 2233, 2256, 2375, 4071, 4075, 4114, 4494
`\eqe@hspannerSoln` 438, 439, 2255, 2379, 2382, 2383,
 4078, 4081, 4100, 4123, 4126, 4133, 4493, 4510
`\eqe@idinfohl` 875, 883, 892, 1391
`\eqe@initializeMultiVersions` 677, 685, 705
`\eqe@innermarkpts` 1545, 1549
`\eqe@insert@more@content` 3308,
 3421, 3530, 3704, 3719, 3745, 3752, 3826, 3832
`\eqe@insertContAnnot` 2982, 3002
`\eqe@insertLogo` 1359, 1361, 1365
`\eqe@insertSolns` 3704, 3729, 3745, 3752
`\eqe@isle@dinnext` 2679, 2680
`\eqe@isPts` 2493,
 2496, 2560, 2592, 2597, 2598, 2701, 2704, 2705
`\eqe@IW` 1908, 2001, 2008, 2034
`\eqe@IW0` 278, 296, 298, 326, 1234,
 1242, 1247, 1665, 1702, 1704, 1717, 1731, 1734,
 1739, 1750, 1754, 1758, 1876, 1882, 1891, 1896,
 1912, 2113, 2129, 2141, 2288, 2291, 2359, 2361,
 2366, 2581, 2618, 2749, 2754, 2763, 2778, 2924
`\eqe@l` 1868, 1870
`\eqe@lcir` 1159, 1161, 1168
`\eqe@leadinitem` 2733, 2794
`\eqe@localTextTitle` 600–602, 609–611
`\eqe@lPanel` 1960, 1966, 1975, 1982, 2014, 2020
`\eqe@makeVgrid` 109,
 3060, 3269, 3273, 3432, 3535, 3702, 3748, 3829
`\eqe@margininsert` 4205, 4206
`\eqe@marktxt` 1546, 1549, 1550
`\eqe@MarParBox` 4199, 4343, 4345
`\eqe@mkRoomPgBrk` 2484, 2490
`\eqe@movePTs` 2653, 2656
`\eqe@DoNum` 2461, 2463, 2706
`\eqe@next` .. 92–94, 623, 624, 634, 636, 645, 1910,
 1912, 1915, 1919, 1925, 2245, 2443, 2446, 2449,
 2726, 2731, 2732, 2796, 2799, 2804, 3260, 3261,
 3263, 3906, 3907, 4800–4802, 5018, 5021, 5023
`\eqe@nextitem` 2954, 2955, 2959, 2960, 2973
`\eqe@NO` 40, 46, 69, 81, 96, 138, 140–142, 145,
 147, 148, 169, 177, 182, 494, 502, 506, 508,
 511, 757, 1000, 1062, 1679, 1680, 2029, 2269,
 2271, 2566, 2593, 2659, 2660, 2677, 2702, 2807,
 2808, 2832, 2858, 2862, 2965, 2970, 3119, 3179,
 3185, 3202, 3210, 3222, 3230, 3865, 4515, 4786
`\eqe@nocustomdesign` 7, 8, 277
`\eqe@nopointsfalse` 493
`\eqe@nopointstrue` 495
`\eqe@nOutOfNum` 2473, 2475, 2707
`\eqe@nskeyflsplit` 3070, 3321, 3545, 3546, 3609,
 3629, 3642, 3655, 3656, 3658, 3671, 3687, 3840
`\eqe@nskeyfltop` 3071, 3630, 3631, 3634,
 3644–3646, 3656, 3657, 3666, 3681, 3685, 3841
`\eqe@numbersep` 3114, 3115, 3245, 3247, 3249
`\eqe@numLinesL` 3105–3107, 3111, 3313
`\eqe@numLinesR` 3105–3107, 3111, 3314
`\eqe@numRefii` 44, 1278, 1281, 1282
`\eqe@offVersion` 649, 653, 689, 743
`\eqe@One` 41,
 2156, 2158, 2165, 2177, 2601, 2602, 2607,
 2648, 2666, 2713, 2715, 2724, 2896, 2908, 2942
`\eqe@onVersion` 687, 740
`\eqe@optiont@kenMsg` 78, 82, 97, 170, 178, 183
`\eqe@opts@argi` 4778, 4846
`\eqe@opts@argii` . 4779, 4842–4844, 4847, 4848, 5006
`\eqe@opts@argiii` 4780, 4848
`\eqe@opts@cbf` 1025, 1027, 1076
`\eqe@opts@schl` 1026, 1027, 1044
`\eqe@opts@fillIn` 2407, 2408, 2433, 5003, 5008

<code>\eqe@optsmultf</code>	1087, 1088, 1101	<code>\eqe@tmpexp</code>	264, 265
<code>\eqe@optsRadiof</code>	962, 964, 1014	<code>\eqe@tmphold</code>	4159, 4160, 4163, 4164
<code>\eqe@optsRadiol</code>	963, 964, 980	<code>\eqe@turnOffComment</code>	698, 742, 745, 803, 806
<code>\eqe@OutOfNum</code>	2472, 2474, 2707	<code>\eqe@turnOnComment</code>	693, 737, 739, 801
<code>\eqe@p@gobnxtpr</code>	2570–2572, 2625	<code>\eqe@Two</code>	42, 2865
<code>\eqe@p@gobnxtprDef</code>	2568, 2570	<code>\eqe@usedeffboxrule</code>	3119, 3132, 3143, 3179, 3265
<code>\eqe@p@gobtop@rnext</code>	2568, 2569	<code>\eqe@webOpts</code>	164, 165, 378
<code>\eqe@paneli</code>	1862, 1863, 1865, 1867	<code>\eqe@wordNumbsError</code>	2467, 2471, 2478
<code>\eqe@panelir</code>	1862, 1864	<code>\eqe@wordNums</code>	80, 2466, 2470, 2476, 2481
<code>\eqe@parboxOptArgs</code>	4846, 4849	<code>\eqe@workfill</code>	1841, 1845
<code>\eqe@partsIndent</code>	2986, 2989, 2991	<code>\eqe@workwidth</code>	1839, 1843
<code>\eqe@pForPart</code>	2352, 2354–2356	<code>\eqe@writetoAux</code>	1245, 2311, 2314, 2812, 2846
<code>\eqe@pointsEach</code>	2762, 2858, 2862, 2866	<code>\eqe@writetoSolns</code>	1238, 2311, 2314
<code>\eqe@pointsPartsId</code>	2648, 2668, 2724, 2729, 2863, 2865	<code>\eqe@wrtLineKernal</code>	3278, 3342
<code>\eqe@prehold</code>	4016, 4026, 4043, 4046	<code>\eqe@x</code>	3291, 3298, 3310, 3342, 3352, 3353, 3358, 3361, 3374
<code>\eqe@prevProbZero</code>	2492, 2592, 2593, 2701, 2702	<code>\eqe@y</code>	3342, 3353, 3481
<code>\eqe@priorw@content</code>	3791, 3826, 3832	<code>\eqe@YES</code>	40, 55, 73, 80, 95, 119, 120, 122, 124, 127, 128, 139, 146, 149–151, 168, 176, 181, 373, 374, 381, 383, 385, 496, 503, 507, 509, 510, 565, 755, 766, 907, 931, 978, 1043, 1111, 1172, 1189, 1420, 1601, 1678, 1682, 1690, 1769, 1807, 2050, 2051, 2058, 2060, 2149, 2155, 2170, 2182, 2190, 2267, 2268, 2294, 2306, 2419, 2422, 2492, 2589, 2592, 2658, 2694, 2695, 2701, 2762, 2811, 2834, 2866, 2874, 2969, 2972, 3132, 3143, 3186, 3187, 3194, 3214, 3265, 3823, 3862, 3876, 4677, 4786, 4799, 4988, 4991
<code>\eqe@prtsepMrg</code>	4055, 4083, 4086	<code>\eqe@Zero</code>	41, 2157, 2603, 2647, 2665, 2714, 2719, 2721, 2863
<code>\eqe@prtsepPrb</code>	431, 432, 2814, 2844, 2985, 3355, 4070, 4073, 4074	<code>\eqe@zeroTotalsAllowed</code>	502, 503, 1679
<code>\eqe@prtsepSoln</code>	4077, 4080, 4127, 4132, 4505	<code>\eqeachLabel</code>	1565, 1579, 2640
<code>\eqe@ps@gobnxtpar</code>	2682	<code>\eqeAEFormatting</code>	292, 293, 297, 298
<code>\eqe@ps@gobtop@r</code>	2679, 2681	<code>\EQEcalculateAllTotals</code>	1341, 2332
<code>\eqe@ptsFmt</code>	1487, 1492, 1528, 1537, 1570, 1571, 1576, 1578, 1584, 1673, 1710, 1711, 1713, 2064, 2637, 2639	<code>\eqeCurrProb</code>	2975
<code>\eqe@r</code>	1868, 1870	<code>\eqedbfalse</code>	3424
<code>\eqe@reqPack</code>	421, 423	<code>\eqedecPointSoln</code>	4076, 4079, 4100, 4117, 4538
<code>\eqe@rgi</code>	1775, 1776	<code>\eqedepth</code>	1843, 1952, 1960, 1967, 1977, 1984, 1998, 2000, 2006, 2014, 2021
<code>\eqe@rPanel</code>	1963, 1971, 1976, 1983, 2034, 2035	<code>\eqedsplyOnlyFrst</code>	4012, 4119, 4536
<code>\eqe@setfillinDefaults</code>	4782–4784, 4822, 4824	<code>\eqefillLines@bgonlystar</code>	3169, 3170, 3525, 3526
<code>\eqe@setStartSolns</code>	3759	<code>\eqefillLines@outlineonlystar</code>	3163, 3164, 3521, 3522
<code>\eqe@setVRule</code>	3454, 3457, 3498, 3508	<code>\eqeGenProbNumfalse</code>	4545, 4547
<code>\eqe@showArg</code>	692, 695	<code>\eqeGenProbNumtrue</code>	4548
<code>\eqe@striphbox</code>	1151, 1152	<code>\eqeGrandTotal</code>	2338, 2339
<code>\eqe@Strut</code>	2093, 2098	<code>\eqeifnext</code>	4359, 4362
<code>\eqe@subleftgutter</code>	1965, 1981, 2015	<code>\eqeList (environment)</code>	4052
<code>\eqe@SumryHoriz</code>	1395, 1397, 1449	<code>\eqeLW</code>	1089, 1091, 1092, 1102
<code>\eqe@SumryVert</code>	1394, 1397, 1449	<code>\eqEmail</code>	33, 898, 905
<code>\eqe@tableadin</code>	2734, 2846	<code>\eqEmail@argi</code>	903
<code>\eqe@tb@ship@out</code>	4303, 4307		
<code>\eqe@tb@ship@out</code>	4296, 4476		
<code>\eqe@temp</code>	3015, 3017, 3018		
<code>\eqe@tempbox</code>	3069, 3073, 3074, 3279, 3280		
<code>\eqe@tempcnta</code>	475, 3460, 3495, 3497, 3499, 3500, 3502		
<code>\eqe@Three</code>	42		
<code>\eqe@tmp</code>	661, 665, 669, 679–682, 2630, 2631, 2785, 2786, 3437, 3438, 3445, 3446		
<code>\eqe@tmp@exp</code>	27, 31, 246		
<code>\eqe@tmp@i</code>	26, 28, 30, 243–245		

<code>\eqEmail@argii</code>	899, 901, 902	<code>\eqExam@Ans@sq@f</code>	994, 1106
<code>\eqemaketitle</code>	1339, 1355	<code>\eqExam@Ans@sq@l</code>	965, 1105
<code>\eqemargin</code>	304, 311, 420, 428, 436, 1582, 1614, 1637, 1965, 2005, 2009, 2231, 2249, 2816, 2829, 2988, 2990, 2992, 3345, 3361–3363, 4107, 4108, 4487	<code>\EqExam@SubmitURL</code>	915, 937
<code>\eqEndExamTotalColor</code>	530	<code>\eqexamargi</code>	2299, 2300, 2302
<code>\eqeomarginbox</code>	48, 50, 51, 2167, 2179	<code>\eqexamargii</code>	2296–2298, 2300, 2303, 2311, 2312, 2321
<code>\eqeomarginboxleft</code>	50, 1632	<code>\eqexamCFG</code>	19, 25
<code>\eqeomarginboxright</code>	48, 51, 1622	<code>\eqexamdefReq</code>	1119, 1122
<code>\eqeonlinefalse</code>	117	<code>\eqexammargin</code>	428
<code>\eqeonlinetrue</code>	119	<code>\eqExamName</code>	32, 877, 886
<code>\eqePANELCUT</code>	1899, 1906	<code>\eqExamName@argi</code>	878, 884
<code>\eqepanelheight</code>	1852, 1856, 1857, 1860, 1948, 1949	<code>\eqExamName@argii</code>	878, 881, 883, 1414
<code>\eqepanelwidth</code>	1851, 1855, 1859	<code>\eqExamNumPagesSolns</code>	1260, 1265, 1266
<code>eqepartsquestions</code> (environment)	4497	<code>\eqExamPageLayout</code>	450
<code>\eqepques@itemsep</code>	4499, 4507	<code>\eqExamPriorVspace</code>	1093, 1113
<code>\eqepques@parsep</code>	4498, 4506	<code>\eqExamRunHead</code>	469, 1303, 1326
<code>\eqepques@topsep</code>	4497, 4508	<code>\eqexamsubject</code>	1372, 1439
<code>\eqepquesitemsep</code>	4499, 4502	<code>\eqexcoverpage</code>	56, 1358
<code>\eqepquesparsep</code>	4498, 4501	<code>\eqexcoverpagedesign</code>	1366, 1370
<code>\eqepquestopsep</code>	4497, 4500	<code>\eqExerSolnHeader</code>	2840
<code>\eqeques@itemsep</code>	2237, 2251, 4489, 4500, 4502	<code>\eqexheader</code>	1110, 1123, 4600, 4610
<code>\eqeques@listparindent</code>	2238, 2254, 4492	<code>\eqexheader@wrapper</code>	1109, 1123, 4600, 4610
<code>\eqeques@parsep</code>	2236, 2250, 2324, 4488, 4501	<code>\eqexlisttabheader</code>	2843
<code>\eqeques@topsep</code>	2235, 2252, 4490	<code>\eqEXt</code>	3906, 3910, 3911
<code>\eqequesitemsep</code>	2237, 2241, 4482	<code>\eqExtArg</code>	146, 4514
<code>\eqequeslistparindent</code>	2238, 2242, 4483	<code>\eqfititin</code>	512, 3100, 4598, 4608
<code>\eqequesparsep</code>	2236, 2240, 4481	<code>\eqforinstrfalse</code>	12, 13
<code>\eqequestopsep</code>	2235, 2239, 4480	<code>\eqforinstrtrue</code>	11
<code>\eqeSetExamPageParams</code>	440, 462	<code>\eqforpaperfalse</code>	241
<code>\eqeSolnItemMngt</code>	2244, 2248, 4486	<code>\eqforpapertrue</code>	157, 159, 161
<code>\eqeSumryHoriz</code>	48, 1397, 1399	<code>\eqfortextbookfalse</code>	10
<code>\eqeSumryVert</code>	1394	<code>\eqfortextbooktrue</code>	9
<code>\eqetmplengtha</code>	485, 1090, 1091, 1477, 1478, 1506, 1507, 1792, 1796, 1799, 3113, 3114, 3275, 3277, 3345–3347, 3350, 3351, 3355–3358, 3362, 3367–3369, 3371, 3372, 3375, 3376, 3379, 3380, 3385, 3386, 3388, 3389	<code>\eqglobalversionfalse</code>	674
<code>\eqetmplengthb</code>	486, 3295, 3347, 3369, 4805–4807, 4809, 4810, 4815, 4858, 4862, 4863, 4866, 4870, 4876, 4880, 4891, 4895, 4896, 4898, 4955, 4956	<code>\eqglobalversiontrue</code>	708
<code>\eqevtranstotbox</code>	58, 1676, 1677, 1684, 1693	<code>\eqgrii</code>	326
<code>\eqeWrtExamTitleToSolns</code>	2267–2269, 2306	<code>\eqgriii</code>	313
<code>\eqex@coverpage</code>	56, 152, 1544, 3023	<code>\eqleftmargin</code>	487, 1581, 1650
<code>\eqex@coverpageSubjectFmt</code>	1378, 1437	<code>\eqleftmarginbox</code>	1563
<code>\eqex@coverpageTitleFmt</code>	1381, 1443	<code>\eqlocalversionfalse</code>	675, 2633, 2788
<code>\eqex@coverpageUniversityFmt</code>	1375, 1431	<code>\eqlocalversiontrue</code>	781
<code>\eqExam</code>	1653, 1654, 1656	<code>\eqobeylocalversionfalse</code>	125, 812
<code>\eqExam@Ans@ck@sq@f</code>	1056, 1108	<code>\eqobeylocalversiontrue</code>	126, 776
<code>\eqExam@Ans@ck@sq@l</code>	1028, 1107	<code>\eqp@rtc@lcm@rk</code>	2835, 2916
		<code>\eqpanelbox</code>	64, 1850, 1854–1857
		<code>\eqpointLabel</code>	1569, 2066
		<code>\eqpointsLabel</code>	1567, 2065, 2067
		<code>eqpointsofar</code> (environment)	19
		<code>eqpointsthispage</code> (environment)	19
		<code>\eqPriorVspace</code>	1113
		<code>\eqptLabel</code>	1564, 1577, 1579, 1585, 1674, 2638, 2640

<code>\eqptsLabel</code>	1563, 1577, 1580, 1585, 1675, 2638, 2640
<code>\eqrightmarginbox</code>	489, <u>1586</u> , 1651
<code>\eqSavedComment</code>	413, 3808
<code>\eqSID</code>	<u>32</u> , 889, 895
<code>\eqSID@argi</code>	889, 893
<code>\eqSID@argii</code>	889, 891, 892, 1415
<code>\eqSolnExCmds</code>	3592, 3615
<code>\eqsolutionshook</code>	<u>1316</u> , 4524, 4526
<code>\eqtemptokena</code>	591, 617, 628, 630, 632
<code>\eqtemptokenb</code>	592, 627, 629, 631
<code>\eqTopOfSolnPage</code>	280
<code>\eqTWSave</code>	1612, 2281, 3868
<code>\equalCellSizesOff</code>	.. <u>109</u> , 3157, 3193, 3269, 3273, 3283, 3396, 3398
<code>\equalCellSizesOn</code>	<u>109</u> , 3156, 3394
<code>\equsecolorfalse</code>	156, 157
<code>\equsecolortrue</code>	159, 161
<code>\eqWLSpacing</code>	<u>3063</u> , 3189
<code>\eqWriteLine</code>	3048, 3049, 3051, 3053, 3055, 3057, 3060, 3233, 3234, 3285, 3292, 3314, 3511, 3516
<code>\eqWriteLineBlankFill</code>	3055, 3081, 3082
<code>\eqWriteLineColor</code>	<u>3061</u>
<code>\eqWriteLineDashFill</code>	3053, 3079, 3127
<code>\eqWriteLineDashVFill</code>	3089, 3128
<code>\eqWriteLineDots</code>	3051, 3075, 3130
<code>\eqWriteLineFill</code> 3049, 3057, 3060, 3066, 3124, 3136, 3237
<code>\eqWriteLineVDots</code>	3086, 3131
<code>\eqWriteLineVFill</code>	3083, 3125, 3137, 3238
<code>\eqwritetomarginsfalse</code>	14
<code>\eqwritetomarginstrue</code>	15
<code>\eqx@separationrule</code>	150, 510, 511, 2170, 2182, 2190
<code>\evensidemargin</code>	.. 445, 4250, 4254, 4256, 4365, 4434, 4451, 4458
<code>\every@qRadioButton</code>	983, 1047
<code>\every@RadioButton</code>	983, 1018, 1047, 1080
<code>\every@sqRadioButton</code>	1019, 1081
<code>\everypar</code>	2833, 2834, 3800
<code>\eWebCustomize</code>	249
<code>\ex@solns</code>	295, 309, 325, 1224, 1232, 1992, 4387–4389, 4522
<code>\Exam</code>	<u>581</u> , 603, 648
<code>\exam</code>	4587
<code>exam</code> (environment)	<u>2259</u>
<code>\examAnsKeyLabel</code>	870, 872
<code>\exambegdef</code>	<u>2103</u> , 2320
<code>\examEmailLabel</code>	896, 897
<code>\examenddef</code>	<u>2147</u> , 2323
<code>\examenvfalse</code>	2219, 4149
<code>\examenvtrue</code>	2144
<code>\examNameLabel</code>	<u>32</u> , 873, 874
<code>\examNum</code>	<u>579</u>
<code>example</code> (environment)	<u>4593</u>
<code>example*</code> (environment)	<u>4603</u>
<code>\examplenoname</code>	<u>150</u> , 4592
<code>\examSIDLabel</code>	887, 888
<code>\examSolnHeadFmt</code>	<u>1224</u> , <u>4563</u>
<code>\excludacomment</code> 527, 681, 700, 4166, 4167, 4394, 4424, 4427
<code>\execjs</code>	385
<code>\ExecuteOptionsX</code>	100, 115, 116, 119, 124, 206
<code>\exer@solnheadhook</code>	758
<code>\exercisesAtEndOfChapter</code>	<u>4394</u>
<code>exerquizOpts</code> (option)	<u>9</u>
<code>\exerSolnHeader</code>	4557
<code>\exerSolnInput</code>	4520
<code>\exerSolnsHeadnToc</code>	184, 4517, 4530
<code>\exerwparts@cols</code>	2445, 5020
<code>\xlabel</code>	2369, 4596, 4597, 4606, 4607
<code>\xlabelformat</code>	.. 2373, 2374, 2376, 4112, 4597, 4607
<code>\xlabelformatwp</code>	2376
<code>\exp@temp</code>	681, 696, 700
<code>\expGT</code>	2349, 2350
<code>\exPrtsep</code>	4074
<code>\exrtnlabelformat</code>	2385, 2389, 4599, 4609
<code>\exrtnlabelformatwp</code>	2386, 2390
<code>\exsecrunhead</code>	2399, 2400
<code>\exsectitle</code>	2395, 2396
<code>\exsectitletext</code>	2394, 2395, 2397
<code>\exslllabelformat</code>	2377, 4116
<code>\exslllabelformatwp</code>	2380, 4118
<code>\exsolafter</code>	2371, 3571, 3574
F	
<code>\F</code>	941, 943, 945, 947, 949, 1590, 1596
<code>\fbInsSolnsStyle</code>	4517–4519
<code>\fbox</code>	1461, 1472, 1495, 1501, 1540, 4737, 4738
<code>\fboxrule</code>	1473, 1502, 1559, 3330, 4689, 4734, 4863, 4896, 5006
<code>\fboxsep</code>	1473, 1502, 4308, 4689, 4734, 4829, 4863, 4896, 4955, 5006
<code>\Ff</code>	886, 895, 959, 1102, 5009
<code>\FfMultiline</code>	1102, 5009
<code>\FfNoToggleToOff</code>	959
<code>\FfRequired</code>	886, 895
<code>\FHidden</code>	941, 943, 945, 947, 949, 1590, 1596
<code>\fieldName</code>	972, 980, 1007, 1015, 1036, 1044, 1069, 1076, 1094, 1102, 1588, 1591, 1595, 1599, 2424, 2434, 4993, 5004, 5009
<code>\filename@base</code>	4618, 4619, 4647, 4648

<code>\filename@ext</code>	4617, 4619, 4646, 4648	3335, 3337, 3358, 3374, 3402, 3405, 3416, 3434,
<code>\filename@parse</code>	4616, 4645	3452, 3505, 3554, 3620, 3664, 3668, 3679, 3683
<code>\fillerBgIm@ge</code>	3425, 3426, 3486, 3491	<code>\flfrstsplittfalse</code>
<code>\fillerCustomBg</code>	110, 3425	3662, 3677
<code>\fillerLinesAlignDef</code>	101, 2027, 3095, 3150	<code>\flfrstsplitttrue</code>
<code>\fillerLinesOnLeftMargin</code>	101, 3094, 3151	3332, 3548
<code>\fillin</code>	158, 2407, 2443, 2447, 4781, 5018, 5022	<code>\fliPartNo</code>
<code>\fillinColor</code>	530	2963, 2971
<code>\fillineol</code>	157, 4792, 4795, 4799	<code>\flnum</code>
<code>\fillineol@i</code>	4801–4803	3173, 3174
<code>\fillineol@ii</code>	4803, 4804	<code>\flPageBreakMsg</code>
<code>\fillineolNoCBMsg</code>	4791, 4801	3098
<code>\fillineolTooLongMsg</code>	4787, 4806, 4810	<code>\fls@vebaselinelineskip</code>
<code>\fillInFormatDefault</code> ..	155, 2457, 4516, 4730, 4731	3566, 3617
<code>\fillinTotalHeight</code> ..	4917, 4938, 4963, 4984, 5004, 5010	<code>\flSeparateCutNames</code>
<code>\fillinWidth</code>	2450	113, 3563
<code>\fillLinesLineWidth</code>	1795, 1796, 3250	<code>fltype</code> (key)
<code>\fillLinesNumFmt</code>	103, 3173–3175	102
<code>\fillTypeBlankLine</code>	3055, 3144	<code>\flwriteexsol@fter</code>
<code>\fillTypeDashLine</code>	3053, 3141	3570, 3593, 3600, 3603
<code>\fillTypeDefault</code>	3044, 3057	<code>\font</code>
<code>\fillTypeDots</code>	3051, 3142	1159
<code>\fillTypeGrid</code>	3059, 3145, 3706	<code>\fontnotesize</code>
<code>\fillTypeHRule</code>	3049, 3140	3423
<code>\firstemittrue</code>	4150	<code>\footskip</code>
<code>\firstitemfalse</code>	2244, 4144	448, 459, 4201, 4202
<code>\firstitemtrue</code>	2258, 4143, 4496	<code>\forceEqu@lCells</code>
<code>\firstPageOfExam</code>	1267	3286, 3399
<code>\firstPartLtr</code>	4536, 4541	<code>\forceEqualCellsfalse</code> ..
<code>\fitwidth</code>	4679	3192, 3268, 3272, 3282, 3397
<code>\fl@Cnt</code>	3563, 3568, 3579, 3581	<code>\forceEqualCellstrue</code>
<code>\fl@CutName</code>	3563, 3569, 3731, 3768, 3795, 3810	3395
<code>\fl@firstpassfalse</code>	3299, 3618	<code>\forceNoColor</code>
<code>\fl@firstpasstrue</code>	3257, 3756	530
<code>\fl@getContent</code>	3578, 3768, 3810	<code>\ForceNoColorfalse</code>
<code>\fl@iSmash</code>	3473, 3475, 3478	160
<code>\fl@nRows</code>	3297, 3319, 3433	<code>\forceNoColorSet</code>
<code>\fl@oSmash</code>	3473, 3475, 3477, 3533	556, 569, 572
<code>\fl@set@nnotContStrSkip</code>	3253, 3258	<code>\ForceNoColortrue</code>
<code>\fl@vsplitandplace</code>	3607, 3731, 3795	162, 571
<code>\fl@wlspaceing</code>	3176, 3177, 3189	<code>forcolorpaper</code> (option)
<code>\flbaselineskip</code>	3567, 3638, 3643, 3648	9
<code>\fleqnOff</code>	362, 363	<code>forcolorpaper*</code> (option)
<code>\fleqnOn</code>	362, 363	9
<code>\flEqu@lLineWidth</code>	3287, 3288, 3394, 3396, 3418	<code>forinstr</code> (option)
<code>flextended</code> (key)	7	4
<code>\flextendedInput</code>	101, 102, 257	<code>\foritem</code>
<code>\flfboxrule</code>	3068, 3078,	2949
	3083, 3088, 3264, 3265, 3271, 3311, 3312, 3328,	<code>\foritem@cont</code>
		2956, 2961, 2962
		<code>\forleadinitem</code>
		2949
		<code>\format</code>
		4679
		<code>forpaper</code> (option)
		9
		<code>\forproblem</code>
		2949
		<code>forstudent</code> (option)
		4
		<code>fortextbook</code> (option)
		4
		<code>\forVersion</code> .. 134, 602, 611, 649, 653, 674, 2630, 2785
		<code>\FPdiv</code>
		2352
		<code>\FPmul</code>
		2354
		<code>\FPround</code>
		2355
		<code>\fracPrt</code>
		3422
		<code>\frstProbNumShownfalse</code>
		2738, 4537
		<code>\frstProbNumShowntrue</code>
		4541, 4544
		<code>\ftb@argi</code>
		3995
		<code>\ftb@CloseEqeList</code>
		129, 4015
		<code>\ftb@defineInsSpan</code>
		128, 3995, 4025
		<code>\ftb@endprobstarCks</code>
		2741, 2783
		<code>\ftb@EndSpanPrts</code>
		4000, 4004
		<code>\ftb@EqeListPrtsFmt</code>
		128, 4006, 4013, 4029
		<code>\ftb@InsSpan</code>
		3996, 3997, 4032
		<code>\ftb@isANSListOpenfalse</code>
		3955, 3970, 4047
		<code>\ftb@isANSListOpenttrue</code>
		4040
		<code>\ftb@OpenEqeListPrts</code>
		128, 4010, 4032
		<code>\ftb@spanPrts</code>
		128, 3996, 4002

<code>\ftb@tmp@exp</code>	4588, 4589	<code>\hrulefill</code>	3082
<code>\ftbFmtChapter</code>	4569, <u>4575</u>	<code>\hsc</code>	2093, 2100
<code>\ftbInputBookAux</code>	<u>4615</u>	<code>\hsi</code>	2053, 2058–2060
<code>\ftbInputEqTextb</code>	4673, 4675	<code>\htPtBox</code>	1647
<code>\ftbInputSolnFiles</code>	<u>4644</u>	<code>\hyper@anchor</code>	2074, 3016
<code>\ftblabel</code>	4636, 4642	<code>\hypersetup</code>	821, 826, 832, 835
<code>ftbsolns</code> (option)	7	<code>\hypertarget</code>	1110, 4560
<code>fullwidthtext</code> (environment)	<u>4441</u>		
<code>\fvsizeskip</code>	2263		
		I	
		<code>\i@eqe@getiiiOpts</code>	4767, 4769
		<code>\iacvspace</code>	2976, 2977, 2979, 2982, 3099
G		<code>\ic@vspace</code>	2976
<code>\g@addto@macro</code>	132, 134, 687, 689, 691	<code>\ic@vspacedefault</code>	97, 2978, 2979
<code>\getDimSSPanel</code>	64, 1853, 1900	<code>\idinfoHighlight</code>	875, 876
<code>\getSpaceLeftOnPage</code>	82, 2514	<code>\if@bypasseqexamheading</code>	204, 1546, 1698, 3857
<code>\gobblelabel</code>	<u>4642</u>	<code>\if@equalignfilllinestoleft</code>	3093, 3298, 3344, 3384, 3406, 3472, 3800
<code>\gobbletoEndEXT</code>	146, 3907, 4513	<code>\if@eqeuseclassmaketitle</code>	202, 1352
<code>\gobtodot</code>	3422, 3438, 3446	<code>\if@eqlinedfiller</code>	3259, 3560, 3735
<code>\goodbreak</code>	1225	<code>\if@equsedim</code>	1944, 3090
<code>\gridHLineFill</code>	3124, 3127, 3130, 3136, 3233, 3237, 3285	<code>\if@temptlocalversion</code>	676, 812
<code>\gridIndent@dj</code>	3429–3431, 3479, 3480	<code>\ifAB</code>	581, 582, 671
<code>\gridIndentAdj</code>	110, 3428	<code>\ifaebshowgrayletters</code>	190
<code>\gridpgbrkfalse</code>	3251, 3517, 3732, 3758	<code>\ifallowcircmc</code>	113, 1158
<code>\gridpgbrktrue</code>	3300	<code>\ifanswerkey</code>	70, 91, 256, 425, 871, 1004, 1066, 2028, 2384, 3005, 3006, 3590, 3600, 3603, 3615, 3698, 3714, 3740, 3860, 3875
<code>gridtype</code> (key)	102	<code>\ifc@sewrap</code>	2465, 2466, 2469, 2470
<code>\gridtypeselectd</code>	3120, 3122, 3180, 3236	<code>\ifchapterexercises</code>	4377, 4422
<code>\gridVLineFill</code>	3125, 3128, 3131, 3137, 3234, 3238, 3455, 3458, 3497	<code>\ifcont@nnot</code>	2997, 3742, 3820
<code>\grpANSDelimitter</code>	129, 4018, 4043	<code>\ifdim</code>	319, 1948, 2161, 2229, 2444, 2489, 2512, 2524, 2526, 2541, 2545, 3635, 3637, 3647, 3925, 4107, 4241, 4806, 4810, 4833, 4922, 4967, 5019
		<code>\ifdispl@yPoints</code>	522, 1552, 2609, 2634, 2790, 2898, 2908, 2912, 2941
H		<code>\ifdisplayworkarea</code>	193, 1768, 1806, 3809, 3812, 3822
<code>\halfHtPtBox</code>	1643, 2150, 2493, 2528, 2543	<code>\ifdisplayworkareaOff</code>	10
<code>\hangafter</code>	971, 999, 1035, 1061	<code>\ifdisplayworkareaOn</code>	10
<code>\hangindent</code>	971, 999, 1035, 1061	<code>\ifDoNotRecordThisExam</code>	2272, 2282, 2328
<code>\hangSolWPrtsFmt</code>	<u>4135</u>	<code>\ifeq@globalshowsolutions</code>	235, 354, 400, 1881, 1918
<code>\hb@xt@</code>	3073, 3074, 3078, 3279	<code>\ifeq@hidesolution</code>	234, 353, 399, 2839
<code>\hbadness</code>	4189	<code>\ifeq@nolink</code>	238, 357, 403
<code>\hCommSpace</code>	2090	<code>\ifeq@nosolutions</code>	236, 355, 360, 401, 426, 908, 931, 987, 1051, 1913, 1954, 2007, 2420, 2841, 3807, 4419, 4523, 4947, 4989
<code>\HEADERcolor</code>	<u>4208</u>	<code>\ifeq@proofing</code>	237, 356, 402, 1251, 2413, 4889
<code>\headheight</code>	441, 456, 4261	<code>\ifeq@solutionsafter</code>	229, 352, 397, 398, 427, 908, 931, 1805, 1836, 1842, 1847, 1880, 1904, 1909, 1955, 1956, 1959, 1974, 1993, 2018, 2035, 2421, 2842, 4990
<code>\headsep</code>	443, 457, 4261		
<code>\height</code>	1616, 1626, 1633, 1684, 1693		
<code>\hglsue</code>	3245, 3247, 3249, 3352, 3358, 3374, 3480, 3495, 3507		
<code>\hidden@ttr</code>	2575, 2576, 2590, 2591, 2624		
<code>\hiddenbox</code>	<u>4679</u>		
<code>\hideproofing</code>	10, 189		
<code>\Hidesymbol</code>	175		
<code>\hInstrSpace</code>	2047, 2053		
<code>\hline</code>	1557–1559		
<code>\hrule</code>	1146, 1147, 3068		

<code>\ifeqcustomdesign</code>	6, 461	<code>\ifKV@eqFillin@underline</code>	4881, 4884
<code>\ifeq@flextended</code>	111, 3741	<code>\ifl@stDispl@yPoints</code>	525, 2495
<code>\ifeq@flnosolns</code>	3549, 3809, 3812, 3821	<code>\ifmarginsonleft</code>	4246, 4248, 4266, 4369
<code>\ifeq@nopoints</code>	493, 1698	<code>\ifmode</code>	4707, 4855, 4901, 4920
<code>\ifeqedb</code>	3424, 3481, 3482, 3499, 3509, 3659–3661, 3663, 3667, 3674–3676, 3678, 3682	<code>\ifnosolutions</code>	360
<code>\ifeqegenProbNum</code>	4547, 4549, 4550	<code>\ifObeyPTsStar</code>	2642, 2651
<code>\ifeqeonline</code>	117, 255	<code>\ifOKToWriteExamData</code>	308, 324, 1228, 1231, 1240
<code>\ifeqexamCFG</code>	19, 24	<code>\ifpdf</code>	115
<code>\ifeqforinstr</code>	13	<code>\ifpreview</code>	239, 358, 404
<code>\ifeqforpaper</code>	241, 359, 405, 4558	<code>\ifshowlsols</code>	3920
<code>\ifeqfortextbook</code>	10, 421, 1128, 1349, 2143, 2219, 2224, 2736, 2783, 2797, 3856, 4473, 4673	<code>\ifsolutionsafter</code>	230, 398
<code>\ifeqglobalversion</code>	598, 607, 674, 767, 2629, 2784	<code>\ifsolutionsAtEnd</code>	72, 1511
<code>\ifeqlocalversion</code>	675, 2629, 2784	<code>\ifsolutionsonly</code>	233, 294, 307, 323, 527, 1231, 1239, 4522, 4525
<code>\ifeqobeylocalversion</code>	125, 756, 774, 780	<code>\ifsp</code>	4818, 4879
<code>\ifequsecolor</code>	156	<code>\iftb@shipoutPermitted</code>	4296, 4299
<code>\ifeqwritetomargins</code>	15, 4475	<code>\iftherearesolutions</code>	4524
<code>\ifexamenv</code>	4149	<code>\ifuseNumForParts</code>	2962
<code>\iffalse</code>	2274	<code>\ifuserectforms</code>	1203
<code>\IfFileExists</code>	197, 4676	<code>\ifvoid</code>	3321, 3632, 3658, 4321, 4346
<code>\iffirstemit</code>	4150	<code>\ifvspacewithsolns</code>	86, 93, 341, 424, 1127, 1257, 1953, 3721, 4390, 4950
<code>\iffirstitem</code>	2243, 2244	<code>\ifWithinANSGrp</code>	2742, 3954, 3962, 3981, 4040, 4041
<code>\iffl@firstpass</code>	3255, 3608	<code>\ifwithinparts</code>	2987, 3348, 3366, 3586, 3771, 3814
<code>\ifflfrstsplit</code>	3548, 3622, 3662, 3677	<code>\ifwithinsoldoc</code>	966, 995, 1030, 1057, 2234, 2248, 2255, 2258, 4486, 4493, 4496, 4538
<code>\ifforceEqualCells</code>	1794, 3393, 3400	<code>\ifwriteVertic@lFLines</code>	3252, 3483, 3700
<code>\ifForceNoColor</code>	160, 568, 571	<code>\ifxetex</code>	116
<code>\iffrstProbNumShown</code>	4537, 4543	<code>\ignorePTsStar</code>	85, 86, 2644
<code>\ifftb@isANSListOpen</code>	3955, 4028	<code>\ii@eqe@getiiiOpts</code>	4771, 4773
<code>\ifgridpgrk</code>	3251, 3514	<code>\iii@eqe@getiiiOpts</code>	4768, 4772, 4775–4777
<code>\ifiscarryover</code>	4276, 4291, 4359	<code>\importdljs</code>	385
<code>\ifisinlineans</code>	4021	<code>\include@quizzesolutions</code>	2405
<code>\ifisinstred</code>	2147, 3904, 3922, 3944, 4020, 4050, 4221	<code>\include@solutions</code>	4380, 4383
<code>\ifisleadin</code>	2806, 4139	<code>\include@solutions@chapter</code>	4380, 4382
<code>\ifismarginans</code>	2147, 3944, 4024, 4049, 4221	<code>\includecomment</code>	528, 696, 4166, 4167, 4395, 4426
<code>\ifisOnline</code>	139	<code>\includeexersolutions</code>	372, 4381, 4386
<code>\ifisstudented</code>	3899, 3923	<code>\initChapAfterSolns</code>	4463
<code>\ifkeepdeclaredvspacing</code>	3760, 3861	<code>\inlineans</code>	3897
<code>\ifkeyalt</code>	7, 87, 90	<code>\input</code>	1117, 1118, 1845, 1901, 1962, 1969, 1979, 1988, 3617, 4407, 4412
<code>\ifkeyOrkeyalt</code>	91	<code>\InputIfFileExists</code>	20, 103, 136, 245, 250–253, 4531, 4625, 4649, 4674
<code>\ifKV@eqefillLines@bgonly</code>	3187, 3195, 3205, 3270, 3289, 3311, 3485, 3506, 3561, 3736	<code>\inputlineno</code>	3781, 3848
<code>\ifKV@eqefillLines@outlineonly</code>	3186, 3215, 3225, 3266, 3312, 3353, 3496, 3510, 3738	<code>\inputRandomizeChoices</code>	135, 136
<code>\ifKV@eqefillLines@topline</code>	3243, 3290, 3450, 3513	<code>\insE@rlyAtQues</code>	3884, 3885
<code>\ifKV@eqFillin@autolift</code>	4831, 4927, 4972	<code>\insert@circlesymbol</code>	1019
<code>\ifKV@eqFillin@boxed</code>	4830, 4841, 4850, 4859, 4861, 4892, 4894, 4900, 4907, 4942	<code>\insertContAnnot</code>	2504, 3004
<code>\ifKV@eqFillin@enclosesoln</code>	4954	<code>\insertContent</code>	2168, 2172, 2180, 2184, 2192, 2195
<code>\ifKV@eqFillin@fitwidth</code>	4890	<code>\insertGrayLetters</code>	1001, 1013, 1063, 1075

<code>\leadinIndentPrtSep</code>	1802, 1838, 1995, 2815, 2820, 2824, 2827, 3385	<code>\manualcalcparts</code>	93, 2731, 2859
<code>\leadinitem</code>	91, 2672, 2680, 2733, <u>2793</u> , 2959, 2960	<code>\marginans</code>	3897
<code>\leadinitemWarningStar</code>	2669, 2719	<code>\marginboxdesign</code>	53, 1555, 1617, 1627, 1634, 1641, 1685, 1694
<code>\leftmargin</code>	2257, 2825, 4057, 4447, 4495, 4511	<code>\marginparafterhook</code>	2610, 2611, 2856, 2899, 2909, 2943
<code>\leftmarginii</code>	2824, 2825	<code>\marginparpriorhook</code>	2612, 2902, 2913, 2944
<code>\leftmarginPtsEaTxt</code>	54, 1574, 1578	<code>\marginparsep</code>	446, 1582, 1614, 1624, 1637, 1686, 1692, 4249, 4254, 4267, 4365
<code>\leftmarginPtsTxt</code>	54, 1573, 1576	<code>\marginparwidth</code>	447, 4236
<code>\leftmark</code>	4368, 4438	<code>\marginpoints</code>	154, 487, 489, 491, 496, 1552, 2600, 2601, 2887, 2890, 2894, 2907, 2914, 2929, 2939
<code>\leftskip</code>	1802	<code>\marginpointsboxtext</code>	<u>1584</u> , 1618, 1628, 1635, 1641
<code>\lfooteqe</code>	<u>1309</u>	<code>\marginpointtext</code>	54, <u>1570</u> , 1581
<code>\thead</code>	<u>1298</u>	<code>\marginsonleft</code>	3897
<code>\headeqe</code>	<u>1292</u>	<code>\marginsonleftfalse</code>	4246
<code>\theadSol</code>	<u>1316</u>	<code>\mark</code>	1549, 1550, 1699, 2148, 4231
<code>\lift</code>	<u>156</u>	<code>\markEndFor</code>	1764
<code>\lineskip</code>	3562, 3621, 3654, 3664, 3668, 3679, 3683, 3737, 3739	<code>\markEndtFor</code>	<u>1763</u>
<code>\link@@Content</code>	970, 988, 1034, 1052	<code>\markerTotalFmt</code>	<u>1727</u>
<code>\linkcolor</code>	530	<code>\markNumQsFor</code>	<u>1763</u>
<code>\linkContentFormat</code>	954, 956	<code>\markStartFor</code>	<u>1763</u>
<code>\linkContentWrapper</code>	956, 970, 1034, 1213	<code>\MarParBoxFmt</code>	<u>4187</u> , 4313, 4327
links (option)	7	<code>\marparboxwidth</code>	4234, 4239, 4243, 4244
<code>\listparindent</code>	2254, 4446, 4492	max (option)	8
<code>\llap</code>	1167, 1252, 2403, 3244, 3246, 3324, 3659–3661, 3663, 3667, 3674–3676, 3678, 3682	<code>\maxdepth</code>	3623, 3627, 3637
<code>\load@exerquiz</code>	122, 142, 381, 383, 4515	<code>\MCcolor</code>	1211, 1212
<code>\load@web</code>	120, 122, 141, 373, 374, 565	<code>\measurePtBoxHt</code>	56, 1640, 1649
<code>\longTitleText</code>	597	<code>\medbreak</code>	4563, 4574
<code>\loop</code>	715, 731, 795, 2966	<code>\medskipamount</code>	4594, 4604
<code>\lower</code>	1171, 1218, 4201	<code>\MidMargcolor</code>	<u>4195</u>
<code>\lowercase</code>	734, 798, 1891, 1896, 2001, 2008, 2573, 2964	<code>\midMargFmt</code>	4196, 4198
<code>lsol (environment)</code>	<u>4166</u>	<code>\minipage</code>	1997, 2000, 2014, 2021, 3589
<code>\lsols</code>	<u>3897</u>	<code>\month</code>	860, 862, 864
M		<code>\mp@Width</code>	1904, 1956, 2000
<code>\makeAnsEnvForSolsAtEnd</code>	280, <u>285</u>	<code>\mrgDecPt</code>	<u>4082</u>
<code>\makeatletter</code>	4624	<code>\mrgDigitFmt</code>	<u>4062</u>
<code>\makeatother</code>	4635	<code>\mrgNumPrtsep</code>	<u>4082</u>
<code>\makeDoNum</code>	2464, 2878, 2882	<code>\mrgPartFmt</code>	<u>4064</u>
<code>\makeOutOfNum</code>	2468, 2878, 2879	<code>\mrgPrtsep</code>	<u>4082</u>
<code>\makeRefsNums</code>	1283, 1291	<code>\multicolsep</code>	418
<code>\makeRoomForProb</code>	2284, 2485, 2519, 2598, 2705, 4595, 4605	<code>\multiply</code>	771, 790, 2350, 2772, 2777, 3444, 3448, 3464, 3469
<code>\maketitle</code>	<u>1330</u>	myconfig (option)	5
<code>\maketitledesign</code>	<u>1330</u>	myconfigi...myconfigvi (option)	5
<code>\makeVgrid</code>	1793, 3050, 3052, 3054, 3056, 3058, 3060, 3102, 3190, 3269, 3273, 3281, 3296, 3299, 3308, 3319, 3320, 3699, 3702, 3743, 3748, 3824, 3829	N	
<code>\manswers</code>	289	<code>\nbaselineskip</code>	83, 2555, 4614
<code>\manswers@sq</code>	289	<code>\nbaselineskipReset</code>	2556, 2557
		<code>\nDoNum</code>	2461, 2463, 2464, 2706
		<code>\needsModArith</code>	755, 757, 766

ftbsolns	7	
links	7	
max	8	
myconfig	5	
myconfigi...myconfigvi	5	
noHiddensolutions	9	
nohiddensolutions	9	
nomarginwrite	4	
noparttotals	5	
nopoints	5	
noseparationrule	5	
nosolutions	9	
nospacetowork	6	
nosummarytotals	5	
nototals	5	
nozerototals	5	
obeylocalversions	8	
online	7	
parttotalsonleft	5	
parttotalsonright	5	
pdf	7	
pdftex	11	
pointsonboth	5	
pointsonleft	5	
pointsonright	5	
preview	9	
proofing	10	
rendition	8	
showgrayletters	10	
solutionsafter	9	
solutionsonly	9	
textures	11	
totalsonleft	5	
totalsonright	5	
usecustomdesign	4	
useforms	7	
usexkv	8	
vspacewithsolns	7	
webOpts	9	
xetex	11	
\optsCkBxf	1025	
\optsCkBxl	1026	
\optsFillIn	2407	
\optsMlTxFld	38, 1087	
\optsRadioBtnf	962	
\optsRadioBtnl	963	
outlineonly* (key)	103	
\OutOfNum	2470, 2472, 2474, 2707	
		P
\p@ssToFLs	2029, 3254,	
	3259, 3262, 3263, 3744, 3751, 3804, 3825, 3831	
\p@ssToSolns	3728, 3757, 3805, 3850	
\PackageError	268, 722, 1871, 2071, 2478, 2743,	
	2799, 3958, 3965, 3977, 3986, 4241, 4629, 4653	
\PackageInfo	78, 104,	
	375, 712, 1285, 2685, 2688, 4360, 4626, 4650, 4886	
\PackageWarning	18, 21, 65, 106, 497,	
	594, 656, 2458, 2669, 2673, 3109, 3134, 3146,	
	3152, 3198, 3206, 3218, 3226, 3321, 3536, 4531,	
	4695, 4704, 4717, 4727, 4759, 4787, 4791, 4871	
\PackageWarningNoLine		
	74, 258, 368, 1120, 3540, 3777, 3844	
\pagegoal	2161, 2227, 2488, 2511, 2515	
\pagenumbering	1364, 1368	
\pageref	1267, 1268	
\pagestyle	3858	
\pagetotal		
	2159, 2227, 2488, 2511, 2515, 2523, 2540, 2856	
\panel	65, 1862	
\panel@cnt	64	
\panel@write		
	1827, 1874, 1876, 1877, 1882, 1883, 1889, 1897	
\panelgap	64, 1849, 1905, 1957, 2002, 2016, 2021	
\panelGetDimen	1844, 1891, 1900, 1961, 1968, 1978, 1987	
\panelheight	64, 65, 1817, 1860, 1933	
\panelwidth	64, 65, 1817, 1821, 1859, 1934, 2003, 2016	
\paperheight	454, 4261	
\paperwidth	451, 4251, 4256, 4435, 4436	
\paragraph	2041	
\parbox	4679	
\parindent	1417, 1425, 1624, 1770, 3274, 4188, 4446	
\parsep	2250, 4059, 4445, 4488, 4506	
\parsetotals	57, 1653, 1656	
\parshape	2829, 2831, 3801	
\parskip	1483, 1510, 3274, 4058, 4190	
\partialspillovertotals	2104, 2110, 2112	
\partialtotaloe	2111, 2114, 2206, 2214	
\partialtotalpg	2109, 2114, 2207, 2215	
\partNames	480, 1705, 2303, 2304	
\partopsep	4058	
\parts@indent	2814, 2818, 2985	
\partsitemsep	2562	
\partsleadinindent	2819, 2843	
\partsparsep	2562	
\partstabcolsep	2563	
\partstabrowsep	2564	
\partstabtopsep	2564	
\partstopsep	2562	

parttotalsonleft (option)	5	\previewtrue	163, 240, 1000, 1004, 1062, 1066
parttotalsonright (option)	5	\prior@parts@hook	4138
\passedss@Argii	2025, 2026, 2032	\priorexlabelheader	3890
\PassOptionsToPackage	158,	\priorexsectitle	4530
199, 209, 211, 213, 216, 219, 221, 224, 226, 4664		\priorexslinput	4530
\PBS	968, 997, 1032, 1059	\priorP@geBre@kMsg	3096, 3097, 3299, 3309
pdf (option)	7	\priorPageBreakMsg	3096, 3099
pdftex (option)	11	\priorPNPAction	2482, 2484, 3297
\penalty	2327, 4208	\priorSubmitJS	927, 936
\percentForPart	<u>1711</u>	\priorw@Msg	3540, 3543, 3544
\phantom	4551, 4954, 4969, 4980	priorworkarea (environment)	<u>3791</u>
\placeAtxy	<u>1766</u>	\priorWorkAreaCmds	1785–1787, 3596, 3616
\placeCoverPageLogo	<u>1358</u>	\priorworkareaCmds	1784
\placeMarkerHere	<u>1716</u>	\prob@Arg	2857, 2860
\pnpDflt	2517, 2518	\probInMinip@ge	1610, 1620, 3868, 3869
\pointLabel	1568, 1569	\probInMinipage	3859
\pointsAsText	<u>1570</u>	\probInsertSoln	<u>122</u> , <u>3888</u>
\pointsLabel	<u>1563</u>	\problem	2573
\pointsmarginparpush	82, 2152, 2520, 2532, 2547	problem (environment)	<u>2558</u>
\PointsOnBoth	492	problem* (environment)	<u>2668</u>
pointsonboth (option)	5	\problem@cont	2577, 2589
\PointsOnBothSides	35, 490, 492	\probpointseach	2712, 2725, 2752, 2753, 2865
\PointsOnLeft	33, 487	\probSet	<u>4582</u> , 4586
pointsonleft (option)	5	probset (environment)	<u>4585</u>
\PointsOnRight	34, 488	\probstar	953, 955,
pointsonright (option)	5	973, 1008, 1037, 1070, 1095, 2078, 2425, 2445,	
\popEnvir	3802	2451, 2599, 2709, 3957, 3976, 4027, 4994, 5020	
\popProblem	<u>2849</u>	\probV@luesInsert	2579, 2584
\post@parts@hook	4141	\probV@luesInsert	...
\postChapSolnHead	4570, <u>4572</u>	83, 2578, 2584, 2585, 2606, 2892, 2895, 2940	
\postdisplaypenalty	3612	\probvalue	<u>1545</u> , 2613, 2903
\postExamSolnHead	<u>1224</u> , <u>4563</u>	\ProcessComment	4405, 4414
\postPNPAction	2483, 2484, 3300	\ProcessOptions	4665
\postSubmitJS	928, 939	\ProcessOptionsX	254
\pr@b@secondarg	2710, 2728, 2759, 2760	\promoteNewPage	<u>2517</u> , 2982
\pr@bl@m@star	2697, 2698	proofing (option)	10
\pr@blem@star	2699, 2700	\proofingsymbol	40, 286, 1153, 1187, 1207, 1210, 2708
\prbDecPt	429, <u>4069</u>	\proofingsymbolColor	<u>530</u>
\prbNumFmt	<u>4112</u>	\protected@edef	4588, 4823, 4827
\prbNumPrtsep	433, <u>4069</u>	\protected@xdef	648, 652, 4219, 4220
\prbPrtsep	431, <u>4069</u>	\providecommand	195,
\preChapSolnHead	4568, <u>4572</u>	196, 429, 431, 433, 439, 530, 532, 534, 536,	
\predisplaypenalty	3611	538, 540, 542, 544, 546, 548, 550, 553, 573,	
\preExamSolnHead	<u>1224</u> , <u>4563</u>	576, 850, 1563, 1564, 2236, 2400, 4072–4075, 4081	
\presets	980, 1014, 1044, 1076, 1101, 2433, 5003, 5008	\ProvidesFile	1134, 3895
\prevExamName	2277, 2287, 2289, 2292	\ProvidesPackage	4662
preview (option)	9	\prtsIndntSep	3350, 3352
\previewfalse	239, 240	\ps@eqExamheadings	465
\previewOff	240	\ptLabel	1564
\previewOn	240	\PTs	2610, 2611, <u>2637</u>

\ptsLabel 1563
 \pushButton 932
 \pushEnvir 3799
 \pushProblem 2849
 \put 4342, 4344

Q

\Q 1590, 1596
 \qeSumryVert 48
 \qNewPage 3005
 \quiz@solns 295, 296,
 298, 309, 325, 326, 1224, 1232, 1234, 1242, 4388

R

\r 934–936, 938
 \radio@Button 980, 1014
 \RadioFieldSize 981, 1015, 1016, 1022,
 1045, 1077, 1084, 1138, 1170, 1217, 1219, 1220
 \raggedbottom 474
 \raggedcolumns 418
 \raggedright 968, 997, 1032, 1059, 4190
 \raise 3515, 3660, 3675
 \raisebox 909, 1166, 1210, 1345, 1360, 1616,
 1626, 1633, 1684, 1693, 1771, 3500, 3664, 3668,
 3679, 3683, 4931, 4934, 4978, 4979, 5002, 5007
 \RecordThisExamOff 75, 2273
 \Rect 40, 1150, 1213
 \refstepcounter ... 967, 996, 1031, 1058, 2809, 3011
 \removelastparskip 1790, 1808
 \removelastskip 2092, 2164
 rendition (option) 8
 \repeat 720, 748, 809, 2970
 \RequirePackage 2, 3, 265,
 361, 364, 378, 384, 406–409, 416, 421, 4666, 4677
 \reset@color 3325
 \resetacvspace 2974
 \resetFillerCustomBg 110, 3426, 3427
 \resetFldWdth 1092, 1102
 \resetMargHeadColor 4208
 \RestoreCommentCutFile 4397, 4407, 4412
 \restoreFromChapAfterSolns 4467
 \restorelabel 4642
 \restoreLastBotMargin 4185
 \restoreLastTopMargin 4183
 \restorele@dinfixDef 2682
 \restorele@dinpfixDef 2681, 2683
 \restorePageLayout 4449
 \RESTOREPAR 3892
 \ReturnTo 40, 1152
 \reversemarginpar 3917

\reverseVSWs 345
 \rfooteqe 1309
 \rhead 1298
 \rheadeqe 1292
 \rheadSol 1316
 \rightmargin 4447
 \rightmark 4371, 4373, 4439
 \rlap 1168, 1210,
 1252, 3248, 3481, 3482, 3488, 3491, 3495, 3499,
 3507, 3509, 3664, 3668, 3679, 3683, 4948, 4951
 \rowsepDefault 2563
 \rule 1412, 2232, 3078, 3088, 3324, 3489
 \runExamFooter 45, 471, 1315
 \runExamHeader 1303
 \runExamHeaderSol 1316

S

\S 959
 \sameVspace 1947, 3329, 3331, 3879
 \save@message
 ... 678, 683, 730, 749, 760, 794, 810, 814, 3583
 \save@setckpt 4622, 4639
 \save@wlog 3582
 \save@writefile 4620, 4638
 \saveBasicLayoutParams 4432, 4456
 \savedeq@online 351, 396
 \savedifeq@globalshowsolutions 354, 400
 \savedifeq@hidesolution 353, 399
 \savedifeq@nolink 357, 403
 \savedifeq@nosolutions 355, 401
 \savedifeq@proofing 356, 402
 \savedifeq@solutionsafter 352, 397
 \savedifeqforpaper 359, 405
 \savedifpreview 358, 404
 \saveIFEQE 333, 336–339
 \sbox 969, 1033, 1854
 \scriptsize 3499
 \sectionColor 553, 555, 566
 \selectedMC 263, 1149, 1157, 1174, 1184, 1199
 \selectVersion 753
 \selVersion 27, 711
 \separationrule 54, 2171, 2183, 2191, 2231
 \separationruleOff 53, 511
 \separationruleOn 510
 \ServerRetnMsg 34, 923, 924
 \set@display@protect 1233, 1241, 1246
 \set@typeset@protect 1235, 1244, 1248, 4316
 \setBotMargin 135, 4179, 4185
 \setbox 1161,
 1640, 3279, 3545, 3546, 3609, 3629–3631, 3642,

3644, 3645, 3655–3657, 3671, 3687, 3690, 3840, 3841, 3872, 3881, 4309, 4320, 4324, 4334, 4335, 4337, 4803, 4903, 4921, 4928, 4945, 4966, 4973	<code>\solutionparshape</code> 1801, 1837, 1994, 1996, 2793, 2795, 2830, 2833, 2989, 3343, 3798
<code>\setBtnExamSkip</code> 318, 321	<code>\SolutionsAfter</code> 4601, 4611
<code>\setcounter</code> 57, 616, 709, 719, 729, 782, 1661, 1663, 1722, 1725, 1729, 1737, 1743–1745, 2136–2138, 2212, 2615, 2710, 2758, 2809, 2854, 2867, 2875, 2931, 2947, 2951, 2953, 2958, 2963, 4536	<code>solutionsafter</code> (option) 9
<code>\setDefaultfvssizeskip</code> 2259, 4613	<code>\solutionsafterfalse</code> 232
<code>\setDefaultnbaselineskip</code> 2553, 2554	<code>\solutionsafterSkip</code> 1836, 1847, 2022
<code>\setfillinDefaults</code> 157, 4781	<code>\solutionsaftertrue</code> 231
<code>\setFillLinesFmt</code> 103, 3178	<code>\solutionsAtEndfalse</code> 84, 425–427, 517
<code>\setFullWidthHeader</code> 4363, 4468	<code>\solutionsAtEndtrue</code> 72, 424
<code>\setFullWidthLayout</code> 4431	<code>solutiononly</code> (option) 9
<code>\setMarIndents</code> 4088	<code>\solutiononlyfalse</code> 233
<code>\setmulticolprob</code> 419	<code>\solutiononlytrue</code> 185
<code>\setSolnIndent</code> 4098	<code>\solWoPrtsFmt</code> 4112
<code>\setSolnMargins</code> 304, 311, 4106	<code>\solWPrtsFmt</code> 4112, 4145, 4146
<code>\setTopMargin</code> 135, 4171, 4183	<code>\solWPrtsFmt@chang</code> 4130, 4145
<code>\settotalsbox</code> 471, 1655	<code>\spaceskip</code> 4190
<code>\SetUpCutFile</code> 4402	<code>\SpaceToWork</code> 146
<code>\sExam</code> 581, 612, 630, 652	<code>\splitbotmark</code> 4347, 4349
<code>\shortquiz</code> 2321	<code>\splitmaxdepth</code> 3627, 3628
<code>\shortstack</code> 1333–1335	<code>\splitsolution</code> 1815
<code>\shortTitleText</code> 597	<code>splitsolution</code> (environment) 63, 1815
<code>\shortVersionAtext</code> 583	<code>\splitsolutioni</code> 1816, 1819
<code>\shortVersionBtext</code> 583	<code>\splitsolutionii</code> 1820, 1823
<code>\shortwebsubject</code> 830, 831, 1293, 1317	<code>\splitsolutioniii</code> 1817, 1821, 1824, 1826
<code>\shortwebtitle</code> 822, 823, 1293, 1317	<code>\splittopskip</code> 3626, 3653
<code>\showAllAnsAtEnd</code> 279	<code>\sq@priorhook</code> 4552
<code>showgrayletters</code> (option) 10	<code>\sqForms</code> 144
<code>\showproofing</code> 10, 188	<code>\sqlabel</code> 2370
<code>\sloppy</code> 3807	<code>\sqlinks</code> 143
<code>\slshape</code> 4368, 4371, 4373, 4438, 4439	<code>\sqsllabel</code> 1223
<code>\small</code> 1576, 1578, 1584, 1672	<code>\sqslrtnlabel</code> 2393
<code>\smallskip</code> 1993, 4565	<code>\sqsolafter</code> 2372
<code>\smash</code> 1166, 1360, 1811, 3244, 3246, 3248, 3324, 3473, 3475	<code>\sqstar</code> 112, 143, 144, 2321
<code>\solAtEndFormatting</code> 292	<code>\square</code> 2389–2391, 4599, 4609
<code>\solDecPt</code> 4076	<code>\ss@Argi</code> . . . 1839, 1841, 1868–1870, 1958, 1998, 2027
<code>\soln@keys@nLines</code> 1932, 1937, 1941, 2024, 2026, 3336, 3340	<code>\ss@Argii</code> 1828, 1878, 1884, 1915, 1919, 1927, 1930, 1931, 1938, 1942, 1944, 1946, 1950, 1953–1955, 2007, 2025, 2026
<code>\solnGutter</code> 4098, 4101, 4107–4110	<code>\ss@Argiii</code> 1828, 1839, 1902, 1903, 1905, 1957, 1960, 1967, 1977, 1985, 1998, 2003, 2021
<code>\solnhspace</code> 2406, 4560	<code>ssol</code> (environment) 4167
<code>\solnItemMngt</code> 2248, 4486	<code>\ssols</code> 3897
<code>\solnsAtEndcomment</code> 4399, 4409, 4417	<code>\stepcounter</code> 626, 659, 716, 732, 796, 966, 967, 994, 995, 1029, 1030, 1056, 1057, 1873, 2295, 2423, 2967, 4992
<code>\solNumPrtsep</code> 439, 4076	<code>\stretch</code> 1396, 1446, 1448
<code>\solPrtsep</code> 4076	<code>\strip@pt</code> 3437, 3445, 4262, 4265, 4268
<code>\solution</code> 1829, 1831	<code>\stripeqExam</code> 57, 1654, 1660

<code>\strut</code>	1333–1335, 1811, 2010, 2011, 2019, 2092, 2093, 2100, 2415, 2417, 2975, 3100, 4694, 4698, 4910, 4923, 4932, 4958, 4970, 4981	<code>\tb@showlsols</code>	3920, <u>4166</u>
<code>\styleComm</code>	2091, 2093, 2096	<code>\tb@showssols</code>	<u>4167</u> , 4168, 4169
<code>\styleInstr</code>	2048, 2056	<code>\tb@so@next</code>	4301, 4303, 4304
<code>\subject</code>	<u>816</u>	<code>\tb@soln@choice</code>	3920, 4168, 4474
<code>\subjectColor</code>	<u>530</u>	<code>\tb@sols@choice</code>	4169
<code>\SubmitButton</code>	<i>34</i> , 929, 1345	<code>\tb@tws</code>	4452, 4459
<code>\SubmitButtonLabel</code>	<i>34</i> , 925, 926	<code>\tb@woparts@len</code>	4052, 4090
<code>\SubmitInfo</code>	<i>33</i> , 914	<code>\tb@wparts@len</code>	4011, 4054, 4093
<code>\summaryPointTotal</code>	<i>70</i> , 2051, 2063, 2064, 2066	<code>\tballowAllNums</code>	3909
<code>\SummaryTotalsOff</code>	507	<code>\tbBaseName</code>	4618, 4625, 4644, 4647, 4649, 4651
<code>\SummaryTotalsOn</code>	506	<code>\tbBotMargin</code>	4157, 4178, 4179, 4181, 4323
<code>\summaryTotalsTxt</code>	<i>70</i> , 2059, 2064	<code>\tbcontinued</code>	4219, 4226, <u>4233</u>
<code>\sumryAnnots</code>	61–63, 67, 1450	<code>\tbfilterOutEvenNums</code>	3900, 3905
<code>\symbolchoice</code>	382, 389, 1186, 1206, 1209	<code>\tblastpageshipped</code>	4300, 4302, 4306
		<code>\tbMakeFinalCalcs</code>	3926, 4237, <u>4270</u>
		<code>\tbMarginHeaderFmt</code>	<u>4208</u> , 4230, 4349
		<code>\tbmarpboxwidth</code>	3925, 4191, <u>4193</u> , 4200, 4203, 4235, 4236, 4241, 4249, 4254, 4311, 4326, 4365
		<code>\tbminskipbtnlayers</code>	<u>4274</u> , 4315, 4333, 4338
		<code>\tbmrgpartwdth</code>	3999, 4007, 4095
		<code>\tbplaceMargins</code>	<u>4234</u> , <u>4259</u> , 4272
		<code>\tbPostMarginHeader</code>	<u>4208</u> , 4231, 4350
		<code>\tbprbNumFmt</code>	4113, 4114
		<code>\tbPreMarginHeader</code>	<u>4208</u> , 4229, 4348
		<code>\tbSaveBotMargin</code>	4180, 4182, 4186
		<code>\tbSaveTopMargin</code>	4174, 4177, 4184
		<code>\tbSetupForMargins</code>	<u>4247</u> , 4271
		<code>\tbsolnpartwdth</code>	4103, 4127, 4133, 4505
		<code>\tbsolWoPrtsFmt</code>	4116, 4122
		<code>\tbsolWPrtsFmt</code>	4118, 4125, 4131
		<code>\tbSourceFile</code>	4619, 4627, 4630, 4631, 4648, 4654
		<code>\tbTopMargin</code>	4156, 4170, 4172, 4176
		<code>\temp@Exp</code>	2321
		<code>\temp@exp</code>	738, 740, 741, 743, 802, 804, 1935, 1936, 4753, 4756
		<code>\texorpdfstring</code>	817
		<code>\text</code>	1454, 1570, 1576, 1578, 1579, 1584, 1673, 2064, 2637, 2639, 4694, 4698, 4855, 4901
		<code>\textbf</code>	1222, 1226, 1453, 2373, 2374, 2378, 2381, 2403, 2460, 2461, 2472, 2473, 2975, 4115, 4124, 4129, 4214, 4564, 4573, 4597, 4607
		<code>\textbookOpts</code>	<u>3897</u>
		<code>\textColor</code>	960, 1590, 1596
		<code>\textcolor</code>	988, 1052, 1150, 1154, 1421, 2057, 2097, 2166, 2178, 3067, 3076, 3080, 3488, 3498, 3508, 3998, 4018, 4066, 4068, 4214
		<code>\textField</code>	910,
		<code>\textheight</code>	941, 943, 945, 947, 949, 1101, 2432, 5003, 5008
			2284, 2517, 4199, 4202, 4261, 4312, 4320, 4331

<code>\textit</code>	2371, 2372	<code>\thisOpt@OK</code>	73,
<code>\textorpdfstring</code>	395		80, 81, 95, 96, 168, 169, 176, 177, 181, 182
<code>\textsf</code>	3423	<code>\thispagestyle</code>	1426
<code>\textSize</code>	960	<code>\thisPart</code>	4540, 4541
<code>textures</code> (option)	11	<code>\thisterm</code>	858
<code>\textwidth</code>	451–453, 1305, 1306, 1326, 1327, 1332, 1343, 1407, 1427, 1460, 1611, 1624, 1692, 2281, 4251, 4255, 4267, 4435, 4452, 4459	<code>\thisUFexamlabel</code>	2301
<code>\TF</code>	2441, 5017	<code>\tiny</code>	3244, 3246, 3248
<code>\the@cntfillin</code>	2427, 2430, 4996, 4999	<code>\title</code>	816
<code>\the@exno</code>	1110, 2950	<code>\titleColor</code>	530
<code>\thebackofpage</code>	3011, 3013, 3015, 3038	<code>\toggleInstrAns</code>	3933
<code>\thechapter</code>	4569	<code>\toks</code>	2760, 2761
<code>\theduedate</code>	854, 856	<code>\tolerance</code>	4189
<code>\theeq@count</code>	2756, 2779, 2869, 2872, 2877, 2881, 2884, 2933, 2936	<code>topline</code> (key)	102
<code>\theeq@numparts</code>	1547	<code>topmargin</code>	442, 458, 4261
<code>\theeqpointsthispage</code>	1667, 1673, 1674, 1680	<code>\topofprobhook</code>	85, 2567, 2621
<code>\theeqpointvalue</code>	1547, 1718, 2360	<code>\topofprobstarhook</code>	88, 2678, 2735
<code>\theeqquestionnoi</code>	973, 975, 1008, 1011, 1037, 1040, 1070, 1073, 1096, 1098, 1589, 1732, 2076, 2079, 2082, 2086, 2362, 2373, 2375, 2379, 2382, 2403, 2426, 2430, 2716, 2750, 2755, 2765, 2779, 2813, 2847, 4012, 4022, 4037, 4113, 4117, 4119, 4514, 4995, 4999	<code>\topsep</code>	2252, 4059, 4445, 4490, 4508
<code>\theexampleno</code>	4591, 4597, 4607	<code>\tot@lForPart</code>	1712, 2348
<code>\theGrandTotal</code>	1709	<code>\totalForPart</code>	1711
<code>\theHeqquestionnoi</code>	2075	<code>\totalsbox</code>	155, 495, 500, 501, 505, 1668
<code>\theHpartno</code>	2085	<code>\totalsboxleft</code>	58, 500, 1681
<code>\theHquizno</code>	2077	<code>\totalsboxright</code>	58, 501, 1689
<code>\themarkerCnt</code>	1728	<code>\totalsboxtext</code>	57, 1672, 1685, 1694
<code>\thepage</code>	1588, 1595, 1598, 1666, 2142, 2619, 2925, 4282, 4283, 4361, 4368, 4371, 4374, 4438, 4439, 4583	<code>\TotalsOnLeft</code>	37, 500
<code>\thepanel@cnt</code>	1845, 1874, 1901, 1962, 1969, 1979, 1988	<code>totalsonleft</code> (option)	5
<code>\thePartNames</code>	1288, 1488	<code>\TotalsOnRight</code>	38, 501
<code>\thepartno</code>	973, 1009, 1038, 1071, 1096, 2079, 2086, 2383, 2426, 4004, 4008, 4013, 4119, 4120, 4536, 4995	<code>totalsonright</code> (option)	5
<code>\thequizno</code>	953	<code>\totHtPtBox</code>	1645
<code>\therearesolutionstrue</code>	183	<code>\trackProblemsOff</code>	84, 2585, 2586
<code>\thereissolutionfalse</code>	2594, 2696	<code>\trackProblemsOn</code>	84, 2584
<code>\thesection</code>	4586	<code>\turnContAnnotOff</code>	2997
<code>\thinspace</code>	3509	<code>\turnContAnnotOn</code>	2997, 3780, 3847
<code>\thisexamlabel</code>	2050, 2063, 2065, 2068, 2070, 2107, 2118, 2120, 2122, 2123, 2125, 2128, 2130, 2140, 2142, 2223, 2270, 2298, 2360, 2362, 2582, 2587, 2619, 2718, 2720, 2750, 2755, 2764, 2779, 2813, 2847, 2869, 2872, 2877, 2880, 2883, 2925, 2933, 2936	<code>\turnfl@nskeyMsg</code>	257, 3536, 3539
		<code>\turnfl@nskeyOnOff</code>	3693, 3697, 3713, 3740, 3744, 3751, 3755
		<code>\turnflanskeyOff</code>	116, 3547, 3712, 3727
		<code>\turnflanskeyOn</code>	116, 3539, 3547, 3550–3553, 3696
		<code>\turnflnoslnOff</code>	3553
		<code>\turnflnosolnsOff</code>	3551, 3695
		<code>\turnflnosolnsOn</code>	3550, 3552, 3694
		<code>\turnOffFTBShipout</code>	139, 4298
		<code>\turnOffMarAnsOnAnsInline</code>	3933
		<code>\turnOnFTBShipout</code>	139, 4297
		<code>\turnOnMarAnsOffAnsInline</code>	3933
		<code>\twe@kBre@kPoint</code>	3556, 3558, 3559, 3624, 3628, 3691, 3764, 3835
		<code>\tweakBreakPoint</code>	3555
		<code>\txtbkb@xb@t</code>	4153, 4309, 4320, 4321, 4335, 4336, 4346, 4354
		<code>\txtbkb@xh@ld</code>	4155, 4332

\txbkb@xt@p 4154, 4320, 4334, 4336, 4337, 4343, 4345
 \txbkt@ks
 . 4151, 4159, 4160, 4163, 4164, 4317, 4346, 4353
 \txbkt@ksi 4152, 4158, 4159

U

\u,b 4679
 \ulcmd 4743
 \ulcustom 4752
 \underbar 876, 902, 1485,
 1490, 1493, 1525, 1535, 1538, 2410, 4750, 4885
 \underline 4679, 4747
 \unhbox 4807, 4811, 4816, 4918, 4939, 4964, 4985
 \unhcopy 3074
 \unitlength 4308
 \university 816
 \universityColor 530
 \unvbox 3631, 3645, 3657, 3666, 3670, 3681,
 3685, 3686, 4336, 4337, 4339, 4343, 4345, 4354
 \uppercase 600, 609, 710
 \use@email 124, 140, 931, 1420, 1601
 \usebox 1811
 \useCheckForProof 1206, 1208
 \useCircForMC ... 1173, 1174, 1183, 1184, 1190, 1198
 \useCircForMS 41, 1205
 \useCircForProof 1186, 1201
 \useclassmaketitle 10
 \useCrossForProof 1209
 usecustomdesign (option) 4
 \useCustomPartNames 48, 1402
 \useFillerDefault 3046, 3782, 3783, 3849, 3850
 \useFillerLines 3042
 useforms (option) 7
 \useMCCircles 1173, 1190
 \useMCRects 1176, 1177, 1192, 1193
 \usenLineDimen 101, 3092
 \usepackage 4, 272
 \useRectForMC 1148, 1149, 1157,
 1176–1178, 1181, 1182, 1192, 1193, 1196–1199
 \useRectForMS 41, 1204
 \userectformfalse 1205
 \userectformstrue 1204
 \useUIPartNames 48, 1401
 \useVspaceDimen 101, 3091
 usexkv (option) 8

V

\V 942, 944, 946, 948, 950
 vadjForSolnInBx (environment) 3870
 \vbadness 3730, 3767, 3794, 4189

\verbatim@out 1889,
 1891, 1896, 1912, 1992, 3572, 3574, 3584, 3585,
 3588, 3589, 3591, 3595, 3772, 3773, 3815, 3816
 \verbatimwrite 1892, 3605
 \version 838
 \VersionAtext 583
 \VersionBtext 583
 \versionLabel 844, 845
 \vfGo@l 3295, 3302, 3303, 3306, 3327
 \vfil 3666, 3670, 3681, 3685, 3686, 4318, 4339
 \voidb@x ... 3546, 3671, 3687, 3690, 3840, 3841, 3881
 \vphantom 3082
 \vpwsSimulateNoSolns 340, 350
 \vrule 1147, 1558, 3083
 \vsize 2228, 3665, 3669, 3680, 3684, 4312
 \vspaceFiller 3042, 3044, 3047, 3059
 \vspaceFillerDefault 3047
 \vspaceFillerLines
 3043, 3044, 3059, 3252, 3338, 3341, 3401
 \vspaceFillerLines@i 3260, 3264
 \vspacewithkeyOff 520, 1131, 3766, 3806, 3863
 \vspacewithkeyOn 520, 3877
 vspacewithsolns (option) 7
 \vspacewithsolnsfalse 86, 89, 345, 3722
 \vspacewithsolnstrue 88, 98, 282, 3708
 \vsplit 3630, 3644, 3656, 4320, 4334
 \vss 3084, 3087

W

\W 959
 \w@csarg 332
 \w@rningBadOpts 74, 83, 99, 172, 180, 186
 \w@sKeyOn 3862, 3865, 3876
 \widthMTxtFld 38, 1090
 \web@continued 847
 \web@save@maketitle 1350, 1353
 \web@save@title 818, 823
 \web@saved@author 824, 827
 \web@sectionsColorOld 553
 \web@toc 846
 \web@version@value 838, 839, 841, 843
 \web@versionlabel 840, 844
 \webArg 822, 829, 830
 \webauthor 825, 1335, 1421
 \webauthor@color 542, 1421
 \webcopyrightyears 837
 \webemail 833
 \webkeywords 834
 \webnewpage 4521, 4527, 4528, 4558
 webOpts (option) 9

<code>\websubject</code>	831, 918, 1333, 1372, 2398, 2400	<code>\write</code>	278,
<code>\websubject@color</code>	544, 1438		3572, 3574, 3588, 3589, 3591, 3595, 3772, 3815
<code>\webtitle</code>	820, 920, 1334, 1445, 2394	<code>\writeAllAnsAtEnd</code>	294, 1127, 1128, 4390
<code>\webtitle@color</code>	540, 1444	<code>\writeallsolutions</code>	4393
<code>\webuniversity</code>	836, 1433	<code>\writeBeginEqeQuestions</code>	307, 1125, 2139
<code>\webuniversity@color</code>	538, 1432	<code>\writeEndEqeQuestions</code>	322, 2220
<code>\webversion</code>	839	<code>\writelastpage</code>	1706, 2223, 2364
<code>\widowpenalty</code>	3730, 3793	<code>\writeT@SolnFile</code>	302, 311–313, 328,
<code>\widthOfParts</code>	3356		1237, 2274, 2275, 2737, 2797, 4140, 4142, 4567
<code>\widthtpboxes</code>	53, 1554, 1556, 1559, 1591, 1599	<code>\writeToSolnFile</code>	1224, 3888
<code>\WithinANSGrpfalse</code>	3954, 3963, 3982	<code>\writetotalstoaux</code>	2221, 2359
<code>\WithinANSGrptrue</code>	3969	<code>\writeVertic@lFLinesfalse</code>	3701, 3746, 3827
<code>\withinsoldocfalse</code>	2234	<code>\writeVertic@lFLinestrue</code>	3252, 3715, 3753, 3833
<code>\withinsoldoctrue</code>	303	<code>\writeWithSolDocTrue</code>	302, 1129
<code>\wlog</code>	3582	<code>\wrtChapSolnHead</code>	4566
<code>\wLVspace</code>	1940,	<code>\wrtExamTitleInSolns</code>	2268
	2506, 3063, 3064, 3264, 3275, 3302, 3303, 3310,		
	3318, 3334, 3339, 3401, 3434, 3436, 3451, 3461,		
	3468, 3471, 3493, 3622, 3626, 3653, 3668, 3683		
<code>\work@reaCmds</code>	1781–1783, 1802		
<code>workarea</code> (environment)	1774		
<code>\workareaCmds</code>	1780		
<code>\workareadepth</code>	1791, 1799		
<code>\workareasb</code>	62, 1774, 1792, 1811		
<code>\workareaV@dj</code>	1777, 1800		
<code>\workareaVadj</code>	62, 1775, 1779		

X

<code>\x</code>	2198, 2201
<code>xetex</code> (option)	11
<code>\XKV@rm</code>	1938, 1943, 1944
<code>\xspaceskip</code>	4190

Y

<code>\y</code>	2200, 2201
<code>\y@st@Do</code>	2658, 2662
<code>\yest@D@</code>	2658–2660, 2762, 2874

21 Change History

1.6a (2006/01/22)	<code>\selectVersion</code> : added <code>\selectVersion</code> command, also the command was fixed <code>\forVersion</code> so that it can be changed within the document, added switches to control new selection.	28	initial value of <code>\selectVersion</code> to <code>\selectVersion{26}</code> and set the initial value of <code>\forVersion</code> to <code>\forVersion{A}</code>	28	
1.6b (2006/04/03)	<code>\forVersion</code> : Changed the alphabet environments A , B , etc. due to a conflict with <code>\S</code> , the control sequence for the <code>amsmath</code> package for a section: <code>\S</code> expands to §. Names changed to <code>verA</code> , <code>verB</code> , etc.	26	2.0a (2010/05/06)	<code>\rheadeqe</code> : Changed the definitions of <code>\lhead</code> , <code>\chead</code> , and <code>\rhead</code> so they don't clash with the <code>fancyhdr</code> package. If <code>fancyhdr</code> is not loaded at the time <code>eqexam</code> is loaded, we <code>\let</code> the old names to the new names. Therefore, when <code>fancyhdr</code> is loaded first use the new definitions.	44
1.6b (2006/3/12)	<code>\popProblem</code> : Added <code>\pushProblem</code> and <code>\popProblem</code> to grant the ability to interrupt a parts environment for, for example, a multicolumn environment.	92	4.4a (2015/06/01)	problem: Added the hook <code>\topofprobstarhook</code> .	85
1.6c (2006/05/02)	<code>\forVersion</code> : At the end of the package, set the		(2015/06/01)	problem*: Added the hook <code>\topofprobstarhook</code>	88
			(2015/06/02)	problem*: Added <code>\@markttotalvalue</code> which seems to solve a problem with <code>multicols</code> .	

<p>Sometimes the totals are not correct, they do not ‘migrate’ out as they should. 90 (2015/07/21)</p> <p>General: add <code>\let\CT@arc@relax</code> 13</p>	<p><code>\markerTotalFmt</code> to enable the calculation of totals of segments of the exam. 59</p>
<p>v1.6e (2006/05/07)</p> <p>General: Added a custom option feature. Just before the options are processed, the tex compiler looks for the file <code>eqecus.opt</code>. This file should contain one or more custom options. 12</p> <p><code>\eqEndExamTotalColor</code>: Added easy user access to various colors, <code>\proofingsymbolColor</code>, <code>\instructionsColor</code>, <code>\eqCommentsColor</code>, <code>\authorColor</code>, <code>\titleColor</code>, <code>\universityColor</code> and <code>\subjectColor</code> . . . 21</p>	<p>v1.9a (2009/28/09)</p> <p><code>\coverpageTitleFmt</code>: Added these various Fmt commands for coverpage. 48</p>
<p>v1.6f (2006/10/24)</p> <p><code>\forVersion</code>: Fixed a bug in the <code>\eqe@initializeMultiVersions</code> command, made sure that any already defined comment environments are set to relax. 26</p>	<p>v1.9b (2009/09/29)</p> <p><code>\placeCoverPageLogo</code>: Added <code>\placeCoverPageLogo</code> to insert a logo on the cover page. 47</p>
<p>v1.6g (2006/11/29)</p> <p><code>\itemPTsFormatted</code>: Added <code>\itemPTsTxt</code> and <code>\itemPTsFormatted</code> to work with <code>\PTs</code>. Also added a <code>*</code> option, to <code>\PTs</code>, in this case the points are not typeset. 85</p>	<p>v1.9f (2009/10/06)</p> <p><code>\endshortTitleText</code>: Modified <code>\longTitleText</code>, <code>\shortTitleText</code> 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 <code>\forVersion</code> is used. 23</p>
<p>v1.6h (2007/01/24)</p> <p><code>\forleadinitem</code>: Added <code>\forproblem</code>, <code>\foritem</code>, <code>\aNewPage</code> 96</p> <p><code>\qNewPage</code>: Added <code>\aNewPage</code> and <code>\qNewPage</code> . . 98</p>	<p>v2.0 (2010/03/05)</p> <p>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</p> <p>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</p>
<p>v1.6i (2007/09/18)</p> <p><code>\altTitle</code>: Added the command <code>\altTitle</code> as an alternate title for the exam document. This alternate title appear centered under the title of the document. 46</p>	<p>v2.0c (2011/01/11)</p> <p><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. 86</p>
<p>v1.7a (2007/12/10)</p> <p><code>\separationrule</code>: Added <code>\separationrule</code> so user’s can redesign the separation rule that is created between two parts of an exam. . . . 74</p>	<p>v2.0d (2011/03/04)</p> <p>General: Added the <code>vspacewithsolns</code> option . . . 7</p>
<p>v1.7b (2007/21/07)</p> <p>General: Added a <code>solutiononly</code> option 9</p> <p><code>\encloseProblemsWith</code>: Added <code>\encloseProblemsWith</code> to support the <code>solutiononly</code> option 21</p>	<p>v2.0e (2011/03/07)</p> <p><code>\useFillerLines</code>: 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. 99</p>
<p>v1.7c (2008/08/21)</p> <p>General: Added the <code>showgrayletters</code> option to <code>eqexam</code> (ported from <code>exerquiz</code>) 10</p>	<p>v2.0h (2011/04/14)</p> <p><code>\fillin</code>: Modified the calculation of the width of <code>\fillin</code>, the width of enclosing box now equals the requested width 161</p>
<p>v1.8 (2008/11/02)</p> <p><code>\placeMarkerHere</code>: Added a set of commands <code>\placeMarkerHere</code>, <code>\calcFromMarkers</code>, and</p>	<p>v2.0i (2011/04/17)</p> <p>General: Added the switch <code>\ifdisplayworkarea</code> to better control when the work area is to be displayed. 10</p>

v2.0j (2011/04/19)	Added the wrapper package for textbook. . . .	152
enclosesoln: added the <code>enclosesoln</code> key to		
eqFillin family.		153
v2.0k (2011/04/29)		
<code>\vspacewithkeyOff</code> : Added user interface to the		
switch <code>\ifkeepdeclaredvspacing</code> , which is		
defined in <code>eqexam.def/exerquiz</code>		20
v2.0l (2011/05/05)		
<code>\firstPageOfExam</code> : Returns the page number of		
the beginning of the exam with a given name.		43
<code>\lastPageOfExam</code> : Returns the page number of		
the end of the exam with a given name.		43
v2.0n (2011/05/13)		
<code>\promoteNewPage</code> : A simple variation on		
<code>\makeRoomForProb</code> designed for user use.		82
problem: Added <code>*<num></code> to signal in-line display		
of points.		84
v3.0f (2011/08/13)		
<code>\ftbInputBookAux</code> : Added <code>\ftbInputBookAux</code> to		
support solution manual		150
<code>\ftbInputSolnFiles</code> : Added		
<code>\ftbInputSolnFiles</code> to support solution		
manual		152
v3.0g (2011/08/15)		
<code>\hangSolWPrtsFmt</code> : Use this to use “hanging		
indentation” for the parts for problems with		
parts in the solutions file.		133
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.		154
v3.0l (2011/08/22)		
<code>\annotContStr</code> : Defined <code>\eqe@insertContAnnot</code>		
and related commands		97
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.		149
v3.0o (2011/09/20)		
<code>\cngMargHeadColorTo</code> : Added		
<code>\cngMargHeadColorTo</code> and		
<code>\resetMargHeadColor</code> to make it easier to		
change the color of the header globally, or		
just once.		136
v3.0p (2011/09/22)		
General: Added <code>\NoSolutions</code> to be executed in		
preamble, needed with the <code>fortextbook</code>		
package.		20
v3.0q (2011/14/22)		
<code>\eGrpANS</code> : Added <code>\bGrpANS</code> and <code>\eGrpANS</code>		126
v3.0s (2012/01/01)		
<code>\eqexammargin</code> : Moved a copy of		
<code>\eqe@spannerSoln</code> out of the <code>ftbsty</code> to the		
package section, its needed here as well.		18
v3.0t (2012/25/01)		
General: Added four more CFG files are the		
request of a user.		5
v3.0u (2012/09/03)		
General: Added the <code>cfg</code> option for inputting a		
custom config file.		5
v3.0v (2012/03/14)		
<code>\textbookOpts</code> : Include <code>\tbMakeFinalCalcs</code> at		
end of <code>\textbookOpts</code>		125
v3.0w (2012/03/27)		
<code>parbox</code> : Added <code>parbox</code>		154
<code>hiddenbox</code> : Added <code>hiddenbox</code> options		154
v3.0x (2012/04/03)		
<code>hiddenbox</code> : Added <code>\eqe@align@hfill</code> to align		
property. used to set position of content		
when <code>parbox</code> is used.		154
v3.0y (2012/04/20)		
<code>\altTitle</code> : Moved <code>\EQEcalculateAllTotals</code>		
from the bottom to the top of		
<code>\eqemaketitle</code> . In case the author wants the		
grand total of the exam in the title, we need		
to make all calculations before		
<code>\maketitledesign</code>		46
v3.1 (2012/05/16)		
<code>\calcQsBtwnMarkers</code> : Added		
<code>\calcQsBtwnMarkers</code>		60
v3.1a (2012/05/21)		
<code>problem*</code> : Using a more robust method of		
detecting the presence of <code>\auto</code>		88
v3.1b (2012/06/18)		
General: Added <code>\eqe@auto@chk@drivers</code>		7
New requirement <code>ifpdf</code>		4
v3.1c (2012/07/30)		
<code>ulcmd</code> : Added <code>ulcmd</code> to <code>\fillin</code>		155
<code>boxcmd</code> : Added <code>boxcmd</code> to <code>\fillin</code>		155
v3.1d (2012/09/29)		
General: Added <code>\ifeqexamCFG</code> and		
<code>cseqexamCFG</code> ; these are used by ATB to		
attached the configuration file.		5
v3.1e (2012/11/10)		
General: Changed default for <code>\proofingsymbol</code>		41
<code>\makeAnsEnvForSolnsAtEnd</code> : Removed		
hard-wired <code>\proofingsymbol</code>		13

problem* : Removed hard-wired	v4.6a (2016/06/04)
\proofingsymbol 88	General: Added <code>\ifequsecolor</code> to easily distinguish between <code>forpaper</code> and <code>forcolorpaper</code> options. 9
v3.1f (2012/11/27)	v4.6b (2016/10/02)
\leadinitem : Created <code>\leadinitem</code> command for a lead in question for the <code>parts*</code> environment. 90	\showAllAnsAtEnd : Include <code>\eqTopOfSolnPage</code> from <code>exerquiz</code> to support copying question to solution page. 13
v3.2 (2012/12/10)	v4.6c (2017/01/28)
splitsolution : Rewrote <code>splitsolution</code> and related code to change the syntax. 63	ulcmd : added custom underline option 156
v3.2c (2013/02/23)	v4.7 (2017/01/04)
General: Added <code>\selectedMC</code> at end of package . 13	\forleadinitem : Modified <code>\foritem@cont</code> to allow for numbering parts 96
v3.3 (2013/04/07)	v4.9 (2017/03/16)
lsols : <code>exerquiz</code> changed <code>\eqEXT</code> to two variables, so we make the same change here. 125	General: Updated <code>eqexam</code> to reflect changes in <code>eqexam.def</code> 38
v3.3a (2013/05/03)	v4.9a (2017/04/05)
\leadinitem : Changed name of <code>\eq@writeexheader</code> to <code>\eqExerSolnHeader</code> 92	\leadinitem : Added a test for solutions after within lead-in item 92
v3.5a (2014/12/19)	v4.9b (2017/10/10)
General: Made <code>usexkeys</code> as default 8	\eqEndExamTotalColor : Provides color for check boxes when a PDF-related option is not taken 22
v3.6a (2015/01/31)	v4.9c (2017/11/03)
\forleadinitem : Added <code>\eqe@fpmrk</code> to add to the uniqueness of <code>\theHeqquestionnoi</code> 96	\fillTypeGrid : Added <code>\fillTypeGrid</code> 100
v3.7 (2015/02/11)	v5.0 (2017/11/08)
\selectVersion : Worked on the logic 28	General: Added <code>\setFillLinesFmt</code> 103
v3.7a (2015/02/26)	Added hooks <code>\priorPNPAction</code> and <code>\postPNPAction</code> to <code>\makeRoomForProb</code> 81
\eqe@tb@shipout : Added a new Boolean switch <code>\iftb@shipoutPermitted</code> to turn off the shipout. Two convenience commands in support are also added, <code>\turnOffFTBShipout</code> and <code>\turnOffFTBShipout</code> 139	Added support for <code>\fillTypeGrid</code> , added keys <code>numbers</code> , <code>numbersep</code> , and <code>color</code> as well 101
v3.8 (2015/03/17)	Added switch <code>\ifgridpgbrk</code> 105
\leadinitem : Spacing is not correct with standard <code>itemize</code> and <code>enumerate</code> environments, this is a fix. 91	Added vertical rules 100
v4.1 (2015/04/23)	workarea : Added <code>\workareaVadj</code> 62
General: Moved this segment of code to <code>\manualcalcparts</code> , it is taken from the main <code>problem*</code> env. 94	Changes in <code>workarea</code> to support <code>\fillTypeGrid</code> 62
v4.4 (2015/05/24)	Within <code>workarea</code> , adjust <code>\leftskip</code> when the problem is a lead-in type 62
exam : Finish with a <code>\vskip</code> to clear the last item or problem. 76	v5.0a (2017/11/14)
v4.5 (2015/11/10)	General: Placed <code>\eqe@makeVgrid</code> in a group, <code>\count0</code> was leaking out. 110
\fillin : Added <code>\setfillinDefaults</code> 157	v5.0c (2017/11/18)
v4.5a (2015/11/10)	General: Added the <code>outlineonly</code> , <code>bgonly</code> , and <code>bgcolor</code> keys to the <code>eqfileLines</code> family. 103
hiddenbox : Added some logic to the underline key 154	v5.0d (2017/11/19)
v4.6 (2016/01/18)	General: Added a custom background feature . 110
\eqe@tb@shipout : Removed <code>\vfil</code> between unboxing, this allows correct breaking of margin content across pages. 140	v5.0g (2017/12/05)
	General: Added filler lines with <code>nosolutions</code> option 118
	Added option <code>flexextended</code> 7

priorworkarea: Added the priorworkarea environment	118	<code>\eq@RadioCheck@driver</code>	36
v5.0o (2017/12/26)		Insert <code>\bWebCustomize</code> and <code>\eWebCustomize</code> to conform with web dated 2019/10/23.	12
General: Added <code>\chngToNoSolns</code>	20	Replace <code>\eq@RadioCheck@driver</code> by <code>\eq@Radio@driver</code>	36
v5.1.1 (2018/12/05)		v5.1.6 (2019/10/29)	
General: Added <code>\eqprior</code>	14	<code>\fillin</code> : Allow the third argument of <code>\fillineol*</code> to have verbatim text as well.	158
v5.1.10 (2020/03/14)		v5.1.7 (2019/12/17)	
General: <code>\optsMsTxtFld</code> : Pass options to the multi-line text field. in for <code>online</code> option.	38	General: Added <code>\autoInsSolns</code>	146
Added commands for inserting figures in a problem	120	v5.1.8 (2020/01/06)	
Change to reflect <code>eqquestions</code>	69	General: Define <code>\aebc@end</code> as a work around for <code>aeb-comment</code>	16
Introduced control over width of text field in for <code>online</code> option.	38	v5.1.9 (2020/13/06)	
<code>vadjForSolnInBx</code> : Add <code>vadjForSolnInBx</code> to correct for vertical space for a problem in a <code>minipage</code>	121	<code>\eqrightmarginbox</code> : Allow adjustment of <code>\textwidth</code> with a problem is within a <code>minipage</code> .	55
<code>\bProbInsert</code> : Define <code>\bProbInsert</code> and <code>\bItemInsert</code>	122	v5.1a (2018/01/21)	
<code>\makeAnsEnvForSolnsAtEnd</code> : Added <code>\cqqsfalse</code> to definition of <code>\writeWithSolDocTrue</code>	14	problem: added <code>\eqe@p@gobnxtpr</code>	83
<code>\probInMinipage</code> : Define <code>\probInMinipage</code>	121	problem*: added <code>\eqe@gobnxtpar</code>	87
v5.1.11 (2020/11/16)		v5.1b (2018/01/24)	
General: Adj <code>\linewidth</code> in soln file	68	<code>\vspaceFillerLines</code> : Added	107
v5.1.12 (2021/01/07)		v5.1c (2018/01/31)	
problem: Added <code>\probV@luesInsert</code> to support skills package	83	General: Added additional lines to get a better break point	114
Added <code>\trackProblemsOn</code> and <code>\trackProblemsOff</code> to support skills package	84	v5.1d (2018/02/02)	
v5.1.2 (2018/12/05)		<code>\fillin</code> : removed <code>\space\ignorespaces</code> from <code>\fillin</code>	164
General: Use version 3.2 of comment style under the name of <code>aeb-comment</code>	16	exam: Added <code>\RecordThisExamOff</code>	75
v5.1.3 (2018/12/13)		v5.1e (2018/02/09)	
General: <code>\ifdisplayworkarea</code> conditionally defined; defined <code>\displayworkareaOn</code> and <code>\displayworkareaOff</code> . Consistent with <code>exerquiz</code> .	10	problem: Support for <code>h</code> and <code>H</code> argument	83, 85
<code>workarea</code> : added solutions-after condition to displaying <code>workarea</code>	63	v5.1f (2018/02/09)	
<code>\postExamSolnHead</code> : Write to solution file if not solutions-only	42	problem: modify problem env to analyze two optional args	83
v5.1.4 (2019/01/31)		v5.1g (2018/02/14)	
General: <code>\let \comment</code> and <code>\endcomment</code> to <code>\relax</code> we use the definition of the comment environment given in <code>verbatim</code> .	16	<code>hiddenbox</code> : Added <code>fontsize</code> and <code>fboxsep</code> keys	154
v5.1.5 (2019/10/28)		v5.1h (2018/02/18)	
General: Define <code>\eq@Radio@driver</code> in case <code>eqexam.def/exerquiz</code> still use		<code>\fillin</code> : Added <code>\mtoeol</code>	157
		Added test, <code>Opt</code> same as empty width	158
		v5.1i (2018/02/19)	
		<code>\fillin</code> : Changed name to <code>\fillineol</code> and changed order of arguments	157
		v5.1j (2018/04/15)	
		problem: Change comparison to <code>\ifx</code> , <code>\if</code> give incorrect result when points are of the form 11, 22, 33, etc.	83