contsolns

D. P. Story

July 9, 2013

Installation. Open the file `contsolns.dtx` in your favorite tex editor (WinEdt) and compile using the ***tex*** *compiler*, not the **latex** compiler. Doing so, generates the file `contsolns.def`. The files `contsolns.dtx` and `contsolns.def` are distributed with AeB. The exerquiz package has the `contsolns` option to use the code in this file.

Demo file. The demo file for this feature of exerquiz is `contsolns_ex.tex`. This feature is available to users of any supported driver.

1 ⟨∗package⟩

1 Introduction

This collections of definitions is designed for use with **AeB** (more specifically, the web and exerquiz package). My occasional friend Jürgen wanted to have a "continued on next page" string to appear at the bottom of a solution (when it appears at the end of the file) to indicate the solution to this current problem continues onto the next page. Originally, this was done for the `shortquiz` environment, later I extended the feature to the `quiz` and `exercise` environments.

Assumptions. The code assumes that the web package page style, `webheadings`, is in effect; otherwise, this code fails.

2 Documentation

`\rfootContStr` `\Qlabel` `\Elabel` `\promoteNPHskip`

The use of the `contsolns` option should be pretty seamless. The message that appears in the right footer is `\rfootContStr`; the command takes two argument, the second of which is a short string that identifies the question. It can be redefined. The labels `\Qlabel` and `\Elabel` may also be redefined. When compiled for paper (`forpaper` option), we emit the exerquiz command `\promoteNewPageHere` with an argument of `\promoteNPHskip` in a vain attempt to get the numbers right. `\promoteNPHskip` is set to `.1\textheight` and may be redefined if this value is not working as it should.

This code file uses the running headers, left and right. If you want to use these footers and this feature, you need to preserve the code. This is what I do in this code file:

```
\expandafter\lfooter\expandafter{\expandafter\newLFooterCmd\web@lfoot}
\expandafter\rfooter\expandafter{\web@rfoot\newRFooterCmd}
```

This demonstrates how to put the new code on the left of the old, or on the right.

Problems. Problems arise if the numbers in the footer do not match the exercise or quiz numbers, this may occur when using the `forpaper` option, the workaround is to adjust `\promoteNPHskip` to a larger value.

When using the `navibar` option of `web`, there may be insufficient room to the right of the navigation bar; in this case, shorten the text created by `\rfootContStr`, or reduce the number of navigation buttons on the solution pages.

3 The Code

`\setQNum` `\setENum` `\Qlabel` `\Elabel` sets the question number in to the solutions at end of the file. While `\setENum` does the same thing for exercises. `\Qlabel` provides a label for the quiz question as a result, `\setQNum` expands to strings like 'Q1', 'Q1(a)', or 'Q1(a)(i)'. `\Elabel` does the same for exercises. `\setEnum` expands to 'E1' and 'E1(a)', for example. Both `\Qlabel` and `\Elabel` are fragile and may be redefined. does the same thing for exercises.

```
\def\setQNum{\Qlabel\ifcase\@eqquestiondepth\or\arabic{eqquestionnoi}%
        \or\arabic{eqquestionnoi}(\alph{eqquestionnoii})%
        \or\arabic{eqquestionnoi}(\alph{eqquestionnoii})%
            (\roman{eqquestionnoiii})\fi}
\def\setENum{\Elabel\if\exerstar*\theeqexno(\alph{partno})\else
    \theeqexno\fi}
```

The default definitions of `\Qlabel` and `\Elabel`.

```
\def\Qlabel{Q}\def\Elabel{E}
```

`\solContMarks` `\solContMarks` really does nothing other than expand to arguments enclosed in braces. The arguments act as a signal to the running footer, telling it to insert some text or not.

```
\def\SolContMrk#1#2{\gdef\solContMarks{{#1}{#2}}}
```

The default definition of `\solContMarks`

```
\def\solContMarks{{}{}}
```

We use `\sqPostHeaderHook` (`\qPostHeaderHook`) to insert some special code into the top of each solution. It inserts the command `\SolContMrk` with two arguments, the quiz number (`Q\@shortquizCnt`, resp., `Q\@quizCnt`) and the question number (`\setQNum`).

```
\def\sqPostHeaderHook{%
    \string\SolContMrk{Q\@shortquizCnt}{\setQNum}\relax}
\def\qPostHeaderHook{%
    \string\SolContMrk{Q\@quizCnt}{\setQNum}\relax}
```

```
\def\exer@solnheadhook{%
    \string\SolContMrk{Ex}{\setENum}}
```

\promoteNPHskip used if compiled for paper, this is the amount that we use to promote a new page. We have to do this for otherwise, if TEXs page breaking algorithm is used, then the page number may be wrong.

```
\newcommand{\promoteNPHskip}{.1\textheight}
\ifeqforpaper
```

At the end of each solution we add some code as well.

If the document is compiled for paper, we promote a new page using the command `\promoteNewPageHere{\promoteNPHskip}`, and modify the command `\fpAfterSolutionsSkip` which appears in the `forpaper` option. It's the last command that appears at the end of each solution. Here we clear the arguments of `\SolContMrk`.

```
    \let\eqSQtsave\eqSQt
    \def\eqSQt#1{\promoteNewPageHere{\promoteNPHskip}\eqSQtsave{#1}}
    \let\eqQtsave\eqQt
    \def\eqQt#1{\promoteNewPageHere{\promoteNPHskip}\eqQtsave{#1}}
    \let\eqEXtsave\eqEXt
    \def\eqEXt#1#2{\promoteNewPageHere{\promoteNPHskip}%
        \eqEXtsave{#1}{#2}}
    {\toks0=\expandafter{\fpAfterSolutionsSkip\SolContMrk{}{}}%
    \xdef\fpAfterSolutionsSkip{\the\toks0}}
\else
```

Now if the document is compiled for the screen, each solution starts on a new digital piece of paper, so no need to modify `\eqSQt` as above. The command `\fpAfterSolutionsSkip` is not used in the screen version, so the last command is `\endeqSqt`, which we modify.

```
    \def\endeqSQt{\par\SolContMrk{}{}}
    \let\endeqQt\endeqSQt
    \let\endeqEXt\endeqSQt
\fi
```

\getSolContMarks is the command that decides whether to put something in the running footer. If the first argument is empty (we have already cleared the arguments of `\SolContMrk` we do nothing; if the first argument is nonempty, we emit a message. \rfootContStr is the string that contains the string that appears in the running footer.

\rfootContStr

```
\newcommand{\getSolContMarks}[2]{\ifcontSoln
    \rfootContStr{#1}{#2}\fi}
\newcommand{\rfootContStr}[2]{Solution to {#2} continues next page}
```

Set the running footer; this should only appear in the solutions section at the end of the file. We establish a new switch `\ifcontSoln` that will be use to signal the need for a continuation of the solution.

\ifcontSol

```
\newif\ifcontSoln \contSolnfalse
```

We put this in the left footer. TEX processes from left to right, so we need this on the left, so the switch will be set for the `\cfooter` and `\rfooter`. The \bSolContMarks command makes the decision of setting the switch `\ifcontSoln`. It reads the expansion of `\solContMarks`, which holds two arguments.

\bSolContMarks

```
\newcommand{\bSolContMarks}[2]{%
    \def\eq@argi{#1}\def\eq@argii{#2}%
    \ifx\eq@argi\@empty\global\contSolnfalse\else
        \global\contSolntrue\fi}
```

`\lFootbCont` is placed in `\lfooter`, and it will set the switch by first expanding `\solContMarks`, then `\bSolContMarks`.

```
\def\lFootbCont{\expandafter\bSolContMarks\solContMarks}
```

`\rFootCont` goes in `\rfooter`, and holds the continuation string.

```
\def\rFootCont{\makebox[0pt][r]{%
    \expandafter\getSolContMarks\solContMarks}}
```

Set the left and right footers. `\web@footerhook`

```
\def\addtolfooter{\expandafter
    \lfooter\expandafter{\expandafter\lFootbCont\web@lfoot}}
\def\addtorfooter{\expandafter
    \rfooter\expandafter{\web@rfoot\rFootCont}}
```

Install these footers at the beginning of the document. First check to see if the `webheadings` pagestyle is being used.

```
\def\cs@testWH#1#2{\ifx\webfootwrapper#1
    \def\cs@next{\AtBeginDocument{\addtolfooter\addtorfooter}}\else
    \def\cs@next{\PackageError{contsoln.def}{%
        webheadings of the web package are NOT\MessageBreak
        in effect. The contsoln.def file requires\MessageBreak
        webheadings}{Use the default webheadings pagestyle
        from the web package.}}\fi
    \cs@next
}
```

The default definition of `\@oddfoot` is

```
\renewcommand{\@oddfoot}{\webfootwrapper{..}}
```

So, the first token is `\webfootwrapper`, if that token is there, we install the footers, and they should work as expected; otherwise we declare an error.

```
\AtBeginDocument{\expandafter\cs@testWH\@oddfoot}
```

⟨/package⟩