AcroTEX.Net

The mi-solns Package

D. P. Story

Prepared: January 6, 2019

www.acrotex.net
Version 0.6, 2018/12/29

Table of Contents

1. Introduction

The name of the package is mi-solns, where 'm' is for 'mark' and 'i' is for 'insert'. So the mi-solns package can mark and insert solutions (created by the exerquiz and eqexam packages).

The exerquiz (ctan.org/pkg/acrotex) and eqexam (ctan.org/pkg/eqexam) packages are capable of creating questions and solutions to exercises and quizzes. The purpose of this package is to present commands for marking, and for inserting any marked solution—for whatever reason—into the body of the document.[1] To accomplish this goal, recent versions of exerquiz and eqexam are required (dated 2018/12/13 or later); also required is the shellesc package (ctan.org/pkg/shellesc).

While the document is being compiled, the solution files (SOL and QSL) are being written to, so we cannot input them or read them. What we do is to make a copy of the solution files from within the operating system, and input the copy back into the body of the document when required. Consequently, it is necessary to activate the feature of executing an OS script from within the compiling operation. To activate the feature, the document needs to be compiled with the `--shell-escape` switch (for latex, pdflatex, lualatex, or xelatex, depending on your workflow).

Description of workflow. The procedure is simple and short: (1) mark a solution for insertion (`\mrkForIns{`⟨*name*⟩`}`), just above the `solution` environment; and (2) insert the solution with `\insExSoln{`⟨*name*⟩`}`, `\insSqSoln{`⟨*name*⟩`}`, or `\insQzSoln{`⟨*name*⟩`}`, depending on whether the solution came from an `exercise`, `shortquiz`, or `quiz` environment.

Demo files. There are four demonstration files.

- `mi-solns-eq.tex` uses exerquiz to test the features of mi-solns.
- `mi-solns-eqe.tex` uses eqexam to test the features of mi-solns.
- `create-db.tex` part of a novel idea, it is an application for changing the output file names. Compiling this file creates `poems.lst`, which is a very small database of poetic quotations.
- `use-db.tex` is the companion file to `create-db.tex`. Once `poems.lst` is created, compile `use-db.tex`, which will input `poems.lst` and display the quotations.

2. Copying the SOL and QSL files

Assuming you have determined how to compile using the `--shell-escape` switch, when you compile a mi-solns document, the files `\jobname.sol` and `\jobname.qsl` are copied at the end of the document. The destination filenames are `\jobname-cpy.sol` and `\jobname-cpy.qsl` by default, but these names can be changed through the commands `\declSOLOut` and `\declQSLOut`.

`\declSOLOut{`⟨*sol-out*⟩`}`	(`\declSOLOut{\jobname-cpy.sol}`)
`\declQSLOut{`⟨*qsl-out*⟩`}`	(`\declQSLOut{\jobname-cpy.qsl}`)

[1] A solution may be inserted into anohter solution, but may not be inserted into itself.

At the end of the document, the following OS operations occur:

- Copy `\jobname.sol` to ⟨*sol-out*⟩ (`\jobname.sol` to `\jobname-cpy.sol`)
- Copy `\jobname.qsl` to ⟨*qsl-out*⟩ (`\jobname.qsl` to `\jobname-cpy.qsl`)

Freely change the names of the output files according to your own whims.

The copy commands. The following two commands expand the run-time scripts for copying one file to another.

```
\newcommand{\copyfileCmdEx}{copy \jobname.sol \misolout}
\newcommand{\copyfileCmdQz}{copy \jobname.qsl \miqslout}
```

where `\misolout` and `\miqslout` expand to the names assigned by the declarations `\declSOLOut` and `\declQSLOut`, respectively. These are the definitions for Windows OS, it might be necessary to redefine these commands corresponding to the OS in which you reside.

Turn copying on and off. The SOL and QSL files are automatically copied, by default; however, this operation can be turned off.

```
\copySolnsOn     \copySolnsOff   (preamble only)
\readSolnsOn     \readSolnsOff   (anywhere)
```

`\copySolnsOn` is the default, it causes the solutions to be copied at the end of the document; `\copySolnsOff` turns off the copying. Why would you do this? Well, if you have completed composing your solutions and you are concentrating on the other content of the document, there is no need to repeatedly copy the solution files. Perhaps turn it on for the final *two compiles*. Yes, you read correctly, two compiles. Since the solution files are copied at the end of the document, the refreshed ⟨*sol-out*⟩ and ⟨*qsl-out*⟩ files do not get read until the next compile.

`\readSolnsOn` is initially expanded by the package, so it is the default. If you expand `\readSolnsOff`, the commands `\insExSoln`, `\insSqSoln`, and `\insQzSoln` do not input their target file; instead `\miReadOffMsg` is typeset. The default definition of this command is,

```
\newcommand\miReadOffMsg{(\textbf{?? read is off ??})}
```

This command may be redefined as desired.

3. Marking a solution

The first step is to mark a solution you wish to insert elsewhere in the document.

```
\mrkForIns{⟨name⟩}
```

The placement is just above the targeted `solution` environment.

```
\mrkForIns{⟨name⟩}
\begin{solution}[⟨options⟩]
  ⟨solution⟩
\end{solution}
```

Here, ⟨*name*⟩ is a word consisting of only ASCII characters. As the ⟨*sol-out*⟩ file is read in, a simple string comparison is performed. If mi-solns is not displaying the solution you want to insert, this could be the problem; simplify the word(s) used for the ⟨*name*⟩ argument.

Below is a simple example we'll work with throughout the rest of this documentation.

EXERCISE 1. An intelligent question.

The verbatim listing is,

```
\newif\ifmiswitch \miswitchfalse % used in Example 4
...
\begin{exercise}
\begin{cq}An intelligent question.\end{cq}
\mrkForIns{myIQ}
\begin{solution}
A brilliant answer to the intelligent question.
\ifmiswitch(D. P. Story)\fi
\end{solution}
\end{exercise}
```

Continue to the next section for information on how to insert the solution into the document.

4. Inserting a solution

The second step is to insert the solution into the document. There are three commands for inserting marked content back in the document.

`\insExSoln[⟨inserts⟩]{⟨name⟩}`	(for `exercise` env.)
`\insSqSoln[⟨inserts⟩]{⟨name⟩}`	(for `shortquiz` env.)
`\insQzSoln[⟨inserts⟩]{⟨name⟩}`	(for `quiz` env.)

where, ⟨*name*⟩ must match up with the ⟨*name*⟩ argument of a `\mrkForIns` command somewhere in the document. For solution content to appear correctly, the choice of the insertion command must match the environment that has been marked. If you marked a solution to an exercise, then use `\insExSoln` to insert that solution. This is possible error point, if the ⟨*name*⟩ does not match up with the correct insertion command.

⟨*inserts*⟩ *placed in a group*

The ⟨*inserts*⟩ optional parameter can be most anything that does not disrupt the expansion of the `\ins??Soln` commands. *It is placed within a group.* This may be an interesting, useful, and significant feature, depending on your imagination to use it. One command designed for the ⟨*inserts*⟩ option for an exercise is `\ignoreques`. This is discussed in **Example 2** below.

Example 1. Insert the solution to EXERCISE 1: *Question*: An intelligent question.

Solution: A brilliant answer to the intelligent question.

```
Insert the solution to ⟨link removed⟩: \insExSoln{myIQ}
```

The above is the verbatim listing. It's just that simple! However, the question to the problem also appears! Continue this problem with **Example 2**. □

Example 2. EXERCISE 1 has its question enclosed in the `cq` environment. This environment passes the question along with the solution. If we don't want to include the question, pass a special command `\ignoreques` in the optional ⟨*inserts*⟩ argument. The expansion of ``\insExSoln[\ignoreques]{myIQ}'' is "A brilliant answer to the intelligent question." □

Example 3. Again referring to EXERCISE 1, the `cq` environment (defined in `exerquiz` and `eqexam`) has certain formatting controls. We can pass some formatting commands through the ⟨*inserts*⟩ argument without affecting other `cq` environments in the document. For example,

Question: An intelligent question.

Solution: A brilliant answer to the intelligent question.

```
\newcommand\myFmt{\declCQPre{\emph{\textcolor{blue}{\cqQStr}:}\space}%
  \declCQPost{\par\medskip\noindent
    \emph{\textcolor{blue}{\cqSStr}:}\space\ignorespaces}}
```

Verbatim listing:

```
For example,\par\medskip\noindent\insExSoln[\myFmt]{myIQ}
```

Be sure not to introduce any spurious spaces. □

Example 4. (**Conditional content**) The ⟨*solution*⟩ can have conditional content, which may be controlled locally by passing the state of a Boolean switch through the ⟨*inserts*⟩ argument.

```
\newif\ifmiswitch \miswitchfalse
...
This code results in ``\insExSoln[\ignoreques\miswitchtrue]{myIQ}''
```

This code results in "A brilliant answer to the intelligent question. (D. P. Story)" □

Example 5. (**Changing ⟨*sol-out*⟩**) There is a file `poems.lst` produced by the demo file `create-db.tex`; here, we insert one of the verses by changing the ⟨*sol-out*⟩ file.

The Lama by **Ogden Nash**

The one-l lama,
He's a priest,
The two-l llama,
He's a beast.
And I will bet
A silk pajama
There isn't any
Three-l lllama[a]

[a]The author here is referring to a three-alarm fire, called a "three-alarmer".

Verbatim listing of the poem:

```
\insExSoln[\declSOLOut{poems.lst}]{The Lama}
```

Refer to demo files `create-db.tex` and `use-db.tex`. □

5. Building the solution sets

We can extract the solution to a problem only if the solution file is fully built. Generally, the solution sets are fully built when the solutions appear at the end of the file, but that is not the only situation. The following list summarizes the options for which the solution files are created in their entirety.

- For the `exerquiz` package, any option other than `solutionsafter`. This includes the following options: `nosolutions`, `noquizsolutions`, `vspacewithsolns`, and no solution-related option at all (`\usepackage{exerquiz}`). Hidden solutions are not displayed.
- For the `eqexam` package, any option other than `solutionsafter` and `answerkey`. This includes the following options: `nosolutions`, `vspacewithsolns`, and no solution-related option at all (`\usepackage{eqexam}`). Hidden solutions are not displayed.

You can insert solutions even when the `answerkey` or `solutionsafter` option is in effect. First compile with `vspaceiwthsolns`,[2] for example, then place `\copySolnsOff` in the preamble and compile again with the `answerkey` or `solutionsafter` option. The insertions will then appear on the next compile.

6. My retirement

Now, I simply must get back to it. ☺

[2]This creates fully built ⟨*sol-out*⟩ and ⟨*qsl-out*⟩ files.