

AcroTeX.Net

The eq-save Package

D. P. Story

Copyright © 2018 dpstory@acrotex.net
Prepared: April 23, 2018

www.acrotex.net
Version 1.1, 2018/04/07

Table of Contents

1	Introduction	3
2	Requirements and options of the package	3
3	Basic commands	4
3.1	Bookkeeping commands	5
3.2	Gatekeeper commands	6
4	Demonstration files	6
5	My retirement	6

1. Introduction

In this documentation, the one reading an AeB¹ document is referred to as a student, as that is the intended target audience for education materials developed by the various packages of AeB.

The `exerquiz` package defines the `oQuestion`, `shortquiz`, and `quiz` environments that are used to pose interactive questions to students. The original concept of `exerquiz` was as an integral component in the development of education materials such as digital tutorials or worksheets. A student may work through such a document, learning concepts and answering questions to reinforce his understanding.

In the past, Adobe Reader (now Adobe Acrobat Reader)—henceforth referred to as AR—did not save form data; consequently, work done by the student is lost when the document is closed. In the more recent versions, beginning perhaps with version 11, AR can now save form data. The `eq-save` package was written at a user's request to save all the quiz data so that the student does not lose her results after saving and closing the document.

To be clear, when a PDF (by AR) document is saved, the form data is also saved: the value of the field and properties of the field, such as border color, are saved. However, `exerquiz` keeps running tallies on the student's progress through the document, these are in the form of JavaScript variables, arrays and objects; the current values of these are *not saved* with the document. Therefore, if the student saves the PDF, it is necessary to save a minimal amount of information that can later be used to reconstruct the state of the document at the time student saved and closed the document. This 'state data' is saved to a hidden text field. When the document is opened again, this hidden field is read, and the state of the document is minimally restored.

2. Requirements and options of the package

The `eq-save` package, being a support package, requires `exerquiz`, dated 2017/07/30 or later. Use the package in the usual way,

```
\documentclass{article}
\usepackage[options]{exerquiz}
\usepackage[options]{eq-save}
```

Options: There are only two options for `eq-save`, these are `devmode` and `!devmode`. As we shall see below, there is an optional 'gatekeeper' command `\nameField`. The student is not allowed to peek at the document until a name is entered (preferably the student's own name) into the `\nameField`. When the `\nameField` and its companion `\BeginNoPeeking` are present it is rather inconvenient to develop the document, create and test quizzes because the author must first pass the 'gatekeeper'. When the `devmode` is in force, entering a name in `\nameField` is no longer required to view the rest of the document. The other option, `!devmode` is the opposite of `devmode`, that is, when `\nameField` and `\BeginNoPeeking` are present in the document, the student (or author) must enter a name into the name field before viewing the rest of the document. The `!devmode` option is the default, passing no option is the same as passing `!devmode`.

¹AcroTeX eEducation Bundle

3. Basic commands

Actually, without any of the supporting commands, yet to be described, the document may be saved, closed, opened again. The responses to questions within the `oQuestion` and `shortquiz` environments are as when they were saved, the same is true for a full quiz of a `quiz` environment. So what is the need for this document? This package provides *bookkeeping services*. Before continuing, let's illustrate by example. Answer one of the `shortquiz` questions and answer both questions from the `quiz` environment. Save the document, open and return to the page. When you return answer (correctly) the other question and/or change the answer for the other questions.

Quiz Answer each of these to test your understanding.

Hint: N&L

1. Select the two mathematicians recognized as the originators of Calculus.

Banach

Newton

Hilbert

Liebniz

2. $1 + 16 =$

3. Which of the individuals below is the originator of \TeX ?

Goossens

Rahtz

Mittelback

Lamport

Knuth

Now complete this quiz consisting of the first two questions above, for simplicity.

Respond to each without error. Passing is 100%.

1. Select the two mathematicians recognized as the originators of Calculus.

Banach

Newton

Hilbert

Liebniz

2. $1 + 16 =$

Answers:

①

The field labeled with '①' is one of the new fields, as is the one to its right. As you work through the above questions, the field keeps track of your success rate in all quizzes (from the `oQuestion`, `shortquiz`, and `quiz` environments). When you save, close, and open again, the tally continues where it left off. Ideal for working through a long document. The field 'Clear All' clears all quiz results as well as the field labeled '①', then all is forgotten.

Before continuing on to the discussion of the various commands of this package, several observations are appropriate.

- As soon as you respond to an *immediate-feedback question* (a question from either the `oQuestion` or `shortquiz` environment), the total number of questions of this type is known, its 4 for this set of questions. In the **Quiz**, problem 1 is a multiple selection (MS) question with two correct answers; problem 2 has one correct answer; problem 3 is multiple choice (MC), only one correct question. There is a total of 4 (two + one + one = 4).

- As you answer more immediate-feedback questions, or change your answers from a correct ones to an incorrect ones, the tally changes appropriately, but always says 'out of 4' (in this set of quizzes).
- The quiz (as constructed from the `quiz` environment) has only two questions, but the tally box does not know this until 'Begin' and 'End' are pressed. A quiz consists of *delayed-feedback* questions.
- Results (for the quiz) are not known until the 'End' button is pressed, then the tally field '①' is updated. You'll note that the multiple selection (MS) question only contributes one (1) to the 'out of' number, that's because MS is scored differently than it is for immediate-feedback.

3.1. Bookkeeping commands

The 'bookkeeping' commands are create text field to hold ongoing tally data.

<code>\sField[⟨opts⟩]{⟨width⟩}{⟨height⟩}</code>	(score field)
<code>\ooField[⟨opts⟩]{⟨width⟩}{⟨height⟩}</code>	('out of' field)
<code>\sooField[⟨opts⟩]{⟨width⟩}{⟨height⟩}</code>	(combined score field)

All three field generate a text field with dimensions `⟨width⟩` and `⟨height⟩`, the `opt` allows you to pass eforms field options to change the appearance of the field. `\sField` holds only the number of question answered correctly; `\ooField` holds the 'out of' value, the number of questions detected; `\sooField` is a combination of the first two values, typically is displays '3 out of 6'. The 'out of' phrase is the expansion of `\eqOutOf`, a command defined and documented in `exerquiz`. A convenience way of redefining the 'out of' phrase is through `\declareScorePhrase`:

<code>\declareScorePhrase{⟨JS-str⟩}</code>	
<code>\declareScorePhrase{#1+" \space\eqOutOf\space"+#2}</code>	(default definition)

The argument `⟨JS-str⟩` is a JavaScript string. It is phrased using `#1` and `#2`, which at compile time, are replaced with JavaScript variables that will hold the score value and the 'out of' value.

With `exequiz` you can also assign points for questions in `quiz` environments; as a consequence, there is a similar set of commands to those above.

<code>\psField[⟨opts⟩]{⟨width⟩}{⟨height⟩}</code>
<code>\pooField[⟨opts⟩]{⟨width⟩}{⟨height⟩}</code>
<code>\psooField[⟨opts⟩]{⟨width⟩}{⟨height⟩}</code>

These fields report the point totals. For `oQuestion` and `shortquiz` environments, each question is only 1 point, but for `quiz` environment you can assign points using the `\PTs` command, as illustrated the documentation of `AeB`. We do not illustrate this set of commands in the documentation.

To clear all the fields just described, as well as all fields in the `oQuestion`, `shortquiz` and `quiz` environment, insert the `\clearAllField` into your document.

```
\clearAllField[⟨opts⟩]{⟨width⟩}{⟨height⟩}
```

The interpretation of the parameters are as above.

3.2. Gatekeeper commands

The application that inspired this package was the Professor of the course did not want the student to look at the quizzes until he entered his name. As a result, should you want this sort of ‘security’, include the `\nameField` and `\BeginNoPeeking` commands. The latter should begin on the first page you don’t want your students to look upon until they enter their names.

```
\nameField[⟨opts⟩]{⟨name⟩}{⟨width⟩}{⟨height⟩}
\BeginNoPeeking
\flJSStr[noquotes,noparens]{\EnterNameFirstMsg}{⟨msg⟩}
```

The *name* field allows you to enter a personalized field name, such as ‘ThrName’, or something. The other parameters are the same as above and are described briefly in the description of `\sField`.

When the student enters her name, the document is cleared: all fields that support questions from the `oQuestion`, `shortquiz`, and `quiz` environments; all fields whose commands begin with ‘s’, ‘oo’, ‘soo’, ‘ps’, ‘poo’, and ‘psoo’ fields, as described above. This is a weak security measure to make it more difficult for a student to just take the worksheet of an ‘A’ student and simply remove the name and enter his own name. Of course, unless students supervised in a lab, these measures are only inconveniences.

There are two JavaScript strings that can be rephrased or localized. The default declarations are,

```
\flJSStr[noquotes,noparens]{\EnterNameFirstMsg}
{You must enter your name first}
\dJSStr{\eqerrUnfinishQuizAtSave}{One of your quizzes
is not finished, you will lose those responses.}
```

4. Demonstration files

There are two sample files, `eqsave-name.tex` and `eqsave-noname.tex`, found in the `examples` folder. The first has uses the gatekeeper commands (these are `\sField` and `\BeginNoPeeking`), while the second one does not; otherwise, the files are the same.

5. My retirement

Now, I simply must get back to it. ~~DS~~