

AcroTeX.Net

**The eq-pin2corr package
Apply PIN security to quizzes**

D. P. Story

Table of Contents

1	Introduction	3
2	The preamble	4
3	Package options	4
4	Setting the pin-hash values	4
5	Turning the PIN security feature on and off	6

1. Introduction

This package is an add-on to the `quiz` environment of the `exerquiz` package.

Purpose. This package adds PIN security to a quiz created by the `quiz` environment. To correct a quiz, the document consumer must press the Correct button of a quiz and successfully enter the correct PIN number. For example,

Solve each, passing is 100%.

1. The sum of 1 and 1 is ...

0 1 2 3 4

2. $\cos(\pi) =$

3. $\frac{d}{dx} \sin(x) =$

Take the quiz, when finished, press the Correct button and enter the PIN number to correct the quiz.

(PIN: 5243)

Answers:

This package is designed for the educational sector, for instructors who use the quizzes from the `exerquiz` package to assess the understanding their students of the course material.

PDF Viewers. Discussion of PDF viewers for the document author and the document consumer.

Instructor: Any PDF viewer may be used as a PDF *previewer*, `sumatraPDF`, for instance. To verify the newly created quiz is functioning correctly, Adobe Reader DC (AR) or Acrobat DC (AA) *must be used*.

AR
AA

Document consumers (students): The `exerquiz` and `eq-pin2tocorr` extensively use JavaScript to perform many background tasks. For the student to have any success in this workflow, he/she must use Adobe Reader DC(AR).

AR

Workflow. The package is designed for the following workflow:

- create quiz* 1. The instructor creates the quiz using the `exerquiz` and `eq-pin2corr` packages.
- deliver quiz* 2. The instructor delivers the “PDF quiz” to each student. (System drive or email)
- take quiz* 3. The student takes the quiz. The student can press the Correct but, unless s/he knows the PIN, the quiz is not marked up.
- save quiz* 4. The student saves the PDF quiz in Adobe Reader DC.
- return quiz* 5. The student returns the PDF to the instructor. (System drive or email)
- correct quiz* 6. The instructor opens the returned PDF quiz in AA/AR and presses the Correct button to mark up the quiz. The instructor *saves the quiz*. The instructor records the grade of the student.
- returns quiz* 7. The instructor returns the PDF, at some point, to the student.
- happy people* 8. Both instructor and student happily go on with their lives.

The implementation details are left to the instructor.

Document creation. Any of the PDF creators current to the L^AT_EX world can be used: (1) pdf_lat_ex, (2) lua_lat_ex, (3) x_el_at_ex, and my old friend, (4) dvips/Distiller (or dvips/ps2pdf). In the latter case (4), Acrobat is required to import Document JavaScripts.

2. The preamble

The minimal preamble for documents that use the eq-pin2corr package.

```
\documentclass{article}      % or some other class
\usepackage[<options>]{web}  % recommended but not required
\usepackage[<options>]{exerquiz}[2021/02/17]
\usepackage[<options>]{eq-pin2corr}
...
\declPINId{<pin-num>}{<hash-string>}
\classPINVar{<class-pin-var>}
...
\begin{document}
```

A recent version of exerquiz (2021/02/17 or later) is required; eq-pin2corr brings in the eq-save package (2021/02/17 or later).

3. Package options

There are two options for this package: `showscore` and `!showscore`. If you took the test on page 3, you will have noticed that when End Quiz control is pressed the phrase "Success! Now save and send to the instructor" appears in the `\ScoreField` or the `\PointsField`, this is the default behavior. Passing `showscore` in the optional argument list of eq-key2corr causes the actual score to appear in the same box (eg, `Score: 2 out of 3`), which is historically what always appears in these boxes. The other option `!showscore` is a convenience option to aid in switching from one option to the other without too much cut and paste. The default definition of the text string that appears is,

```
\f1JSStr[noquotes]{\SaveAndSendMsg}{Success! %
Now save and send to the instructor}
```

Refer to the `eformsman.pdf` for a discussion of `\f1JSStr`.

`\showScoreOff` **Local controls.** The two options can be turned off and on locally with `\showScoreOff` and `\showScoreOn` commands.

4. Setting the pin-hash values

In the preamble, as indicated above, are two commands, the first is required, the second is optional.

```
\declPINId{<pin-num>}{<hash-string>}
\classPINVar{<class-pin-var>} (optional)
```

It is through the `\declPINId` command that the PIN security is set up.

`\dec1PINid{<pin-num>}{<hash-string>}` The *<pin-num>* is a number, perhaps four digits, that is used to pass through the security of the Correct button.¹ Once you decide on the PIN number, you need to generate the corresponding *hash-string*. The hash-string is obtained from the demo file `get-hash-string.pdf`, the contents of that files is reproduced below.

Press the button labeled Push and enter your PIN number into the dialog box (enter, for example, 5243), then press OK, the corresponding hash-string appears in the text field, 02JRVZdRgYgCA-Rtje8Vkd. Copy the PIN number (5243) into the first argument of `\dec1PINid` and copy the hash-string (02JRVZdRgYgCA-Rtje8Vkd) from the text field into the second argument, like so,

```
\dec1PINid{5243}{02JRVZdRgYgCA-Rtje8Vkd}
```

The above is placed in the preamble of the document.

When the user enters a PIN number (right or wrong), it is converted to a hash-string and compared with the hash string of the correct PIN number; if they match, then the quiz is corrected. It is important to note that the PIN number *does not appear anywhere* in the document, so it cannot be discovered. Knowledge of the correct hash-string does not help. Of course both PIN number and hash string appear in the source file of the document.

`\classPINvar{<class-pin-var>}` From the instructor's viewpoint, it takes a lot of effort to enter the PIN number for every quiz that is submitted (it may be in the hundreds). There is a way of avoiding the requirement of entering the PIN number, and that is through the use of the `\classPINvar` command. This method will work if you have Adobe Acrobat or if you only have Adobe Reader DC. The steps for doing this are as follows:

1. Decide on a class PIN variable name, it can be a generic one, or one specialize to your class, for example, `\classPINvar{Calc2Sprg21}`, then declare it in the preamble:

```
\classPINvar{Calc2Sprg21}
```

The value of the argument must be a valid JavaScript name.

2. Go to the user config.js file of AA/AR and edit that file by including the following line in it,

```
var Calc2Sprg21 = "02JRVZdRgYgCA-Rtje8Vkd";
```

where the hash string used here is the same one corresponding to the PIN number for the document (5243 in our example).

¹The *<pin-num>* does not have to be a number, it can be any password (passcode) that is easy to remember. I prefer a four digit number.

AA/AR only reads `config.js` one time when it opens, so `Calc2Spr21` will not be known until the next time AA/AR is opened. The location of the `config.js` file is described in general in the file [install_jsfiles.pdf](#), found in the `doc` folder of this distribution.

Once the above line is placed in the `config.js` file, the instructor does not enter the PIN number, pressing `Correct` immediately corrects the quiz.

5. Turning the PIN security feature on and off

The security PIN feature can be turned on and off through the following two commands.

<code>\usePINCorrBtn</code>	(Turn on PIN security)
<code>\restoreCorrBtn</code>	(Turn off PIN security)

These can be placed anywhere outside a quiz environment. The commands take effect beginning at the next quiz in the document.

Now, I really must get back to retirement. ~~DS~~