

**THE UNIVERSITY OF AKRON**  
**Theoretical and Applied Mathematics**

**eCards: Electronic Flash  
Cards Manual of Usage**

**D. P. Story**

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Documentation</b>	<b>3</b>
2.1	Preamble: Required Packages and Options . . . . .	3
	• Required Packages . . . . .	3
	• Drivers . . . . .	3
	• Options . . . . .	3
	• Various Commands . . . . .	5
2.2	<code>\begin{document}</code> : Start creating Cards . . . . .	6
	• The <code>card</code> Environment . . . . .	6
	• The <code>response</code> Environment . . . . .	7
	• The <code>hint</code> Environment . . . . .	7
	• <code>answer</code> Environment . . . . .	7
2.3	<code>\end{document}</code> . . . . .	7

## 1. Introduction

The initial version of this package was developed at the request of my colleague, Dr. Thomas Price, for use in the senior honors project of Ms. Katie Jones on Trig Flash Cards. Upon completion of the honors project, I generalized and extended the original package developed specifically for them.

## 2. Documentation

In this section, the major elements of this package are highlighted. For those who want to know more, you can peruse the  $\LaTeX$  code, there are comments contained there as well. The demo document `eCardstst.tex` illustrates most of what you need to know for creating your own electronic flash cards.

### 2.1. Preamble: Required Packages and Options

- **Required Packages**

This package depends heavily on the **Acro $\TeX$  eEducation Bundle**: (1) the `web` package provides page setup, backgrounds, and navigational elements; (2) the `exerquiz` package allows you to author the questions, both non-responsive and responsive (fill-in and multiple choice); and (3) the `insdljs` packages is the mechanism for introducing document-level JavaScripts into the final document.

Each of the packages listed above have their own required packages: `hyperref`, `color`, `verbatim`, `everyshi` and `eso-pic`. These are all available as part of the standard distribution of  $\LaTeX$  or through any CTAN site.

- **Drivers**

The supported drivers are the same as that of the `web` package: `dvipson`, `dvips`, `pdftex` and `dvipdfm`. There is also the `textures` option, but I have never tried it.

A typical set of packages used for on screen presentation:

```
\usepackage[<driver_option>,tight,nodirectory,rightpanel]{web}
\usepackage{exerquiz}
\usepackage{eCards}
```

where `<driver_option>` is any of the drivers listed above.

- **Options**

**eCards options.** The `eCards` package really has only 4 options:

1. `nohints`: If you do not want to provide hints in your eCards, use this option. See also the comments in ‘The `hint` Environment’ on page 7.

2. **listing**: This option gives you a printable version of your eCards. In this way, you can proofread, check your questions, hints, and answers. Suggested packages and options are given below:

```
\usepackage[<driver_option>,forpaper,tight,nodirectory]{web}
\usepackage[solutionsafter,proofing,preview]{exerquiz}
\usepackage[listing]{eCards}
```

3. **memLogo**: The logo, if any, is read and re-read for each page on which it appears. Using this option, the logo is read once and saved in a box for use.
4. **custom**: If this option is chosen, the package will look for and input a file `eCard.cus`. This file can be used to customize the environments. This file should be kept in the source directory, not in the L<sup>A</sup>T<sub>E</sub>X search path.

**Other options.** Selecting the various options of the `web` and `exerquiz` packages can give you different looks. It is important to be aware of all the options associated with these two package; in the paragraphs below, various options are discussed that may be useful in `eCards`.

**Useful Web Package Options.** There are three background/panel options: `usetemplates`, `rightpanel` and `leftpanel`.

Using the `usetemplates` option does not give you the vertical navigational strip, but it does give you the background colors; the `rightpanel` and `leftpanel` given you a vertical panel on the right and left, respectively. Use one of these three options only, if any at all. Using none of these three will just get you the default white background. The panels themselves can be redesigned completely, see the manual of usage for the AcroT<sub>E</sub>X eDucation Bundle at

<http://www.math.uakron.edu/~dpstory/webeq.html>

for details.

There are certain “standard” page designs, or you can create your own using the `\margins` and `\screensize`; the demo document has

```
\margins{.25in}{.25in}{24pt}{.25in} % left,right,top, bottom
\screensize{3.72in}{366.24bp} % height, width
```

See the `Web` package documentation for details on these and other options.

**Useful Exerquiz Package Options.** If you are not using multiple choice or fill-in questions, you should use the `exercisely` option. This removes much of the document level JavaScript from the PDF document.

For authors that use the full Acrobat 5.0, or the newer Acrobat 6.0 Standard or Acrobat 6.0 Professional, you can use the `execJS` option. If this option is taken, then when the document is first loaded into Acrobat (following distillation, or creation using `dvipdfm` or `pdftex`), the document will be automatically saved; this saves any imported document level JavaScript in the document. The document always needs to be saved after creation so save the scripts with the document, this does it automatically so you can't forget to do it—as one of my colleagues once did.

The `nosolutions` option removes the `response` environment leaving only the questions. The `proofing` and `preview` options can be useful for proofreading, as described in the `listing` option described above.

#### • Various Commands

The `eCards` defines several (text) commands, there are listed here:

- `\cardsFinishedMsg`: When the user has reviewed all the electronic flash cards, an alert dialog appears with a message. The contents of the message are defined by this command. The default is

```
\cardsFinishedMsg{You've seen all the cards!}
```

- `\cardColor`, `\hintColor`, and `\solnColor`: The background colors of various pages. The defaults are

```
\cardColor{vlightblue}
\hintColor{cornsilk}
\solnColor{webyellow}
```

Additionally, `\textBgColor` is used to define the default background color. This will be the color of the first page, the default is `\textBgColor{cornsilk}`. The panel background is controlled by `\panelBgColor`, for example, `\panelBgColor{logoblue}`.

- `\ecLogo`: The logo emblem that would appear in the upper portion of the vertical navigational strip. This assumes you are using either the `rightpanel` or `leftpanel` option for web. The default is

```
\ecLogo{\includegraphics[scale=.4]{graphics/uakron}}
```

As you can see, the argument of this command is a graphic command, usually an `\includegraphics` command from the `graphicx` package.

- `\ecLogoLink`: This is the URL of the link destination that will be placed around the logo. The default is an empty address, in this case, no link is placed around the logo emblem. An example of usage:

```
\ecLogoLink{http://www.uakron.edu/}
```

- `\ecHomePage`: This is the URL to a page. This will be used as a link destination of the ‘Home’ button seen on the first page of the eCards document.

```
\ecHomePage{http://www.math.uakron.edu/\string~dpstory/%acrotex.html}
```

- `\noHintJSAction`: When there are no hints (the document author has taken the `nohints` option, the ‘hints’ button is grayed out, but still functional, with no JavaScript action. You can add some action to this button through this command. An example of usage is

```
\renewcommand\noHintJSAction{app.alert("No hints provided!")}
```

## 2.2. `\begin{document}`: Start creating Cards

The process of creating these electronic cards is quite simple, the sections below describe the `card`, `response`, `hint` and `answers` environments.

- **The card Environment**

The main environment is the `card` environment, which is used for posing questions, offering a hint, and an answer. The syntax is

```
\begin{card}
< a question >
\begin{response}
\begin{hint}
< a hint >
\end{hint}
\begin{answer}
< an answer >
\end{answer}
\end{response}
\end{card}
```

► Important. You can pose a question which requires a verbal response, or one for which there is a choice of alternatives, or a fill-in the blank (math or text). See the demo file `eCardstst.tex` for examples.

- **The response Environment**

Immediately following and nested within the `card` environment is the `response` environment. This sets things up for the responses to the question: the hint and the answer.

- **The hint Environment**

The first environment to appear within the `response` is the `hint` environment. Here you can provide additional information to help the student answer the question successfully. If you've posed a multiple choice or fill-in question, you can simply copy the multiple choice or fill-in into the hint, just as I have done in the demo file `eCardstst.tex`.

In this release, hints can be provided for **all** of the questions or for **none** of the questions. You can enter hints using the `hint` environment, illustrated above, or not include a `hint` environment. When you do not want to include hints—whether you've entered the environments or not—use the `nohints` package option. This will convert the `hint` environment into a `comments`, and redefined some of the navigational buttons.

- **answer Environment**

After the hint environment comes the answer environment where the answer to the original question can be presented. At the end of this environment, you need to back out of your nest: `\end{answer}`, `\end{response}` and `\end{card}`.

### 2.3. `\end{document}`

That's the end! Once you have completed your `eCards` file, you are ready to create a PDF document! The `eCards` package supports PDF creation using the following methods:

1. `.tex`  $\mapsto$  `.dvi`  $\mapsto$  `.ps`  $\mapsto$  `.pdf`. This route uses the `dvipsone` or `dvips` option (for the `web` package), followed by the use of the Acrobat Distiller (Version 5.0 or later suggested).
2. `.tex`  $\mapsto$  `.dvi`  $\mapsto$  `.pdf`. Here, you latex the document, then hit the result using `dvipdfm`. Naturally, you would use the `dvipdfm` option with the `web` package.
3. `.tex`  $\mapsto$  `.pdf`. Here, you use the `pdftex` option of `web`, and `pdflatex` the document.

Go to it, and be creative! 