

THE UNIVERSITY OF AKRON
Mathematics and Computer Science
The Web and Exerquiz Packages
Manual of Usage

D. P. Story

Directory

- **Table of Contents**
- **Begin Article**
- **List of Options**

Copyright © 1999-2000 dpstory@uakron.edu

Last Revision Date: January 28, 2000

Version 1.30

Table of Contents

1. Introduction

- Goals
- A Brief History
- Thanks

2. The Web Package

- Overview
- Package Requirements

2.1. Basic Usage

- Setting the Driver Option
- The pdf`tex` Option
- The tight Option

2.2. Hyperref Options

2.3. The Title Page and TOC

- Basic Information Macros
- Redesigning `\maketitle`
- The `nodirectory` option
- The `latexoc` option

2.4. Navigational Aids

- A Navigational Bar
- Direction Icons

2.5. The Language Options

2.6. The `forpaper` option

3. The Exerquiz Package

- Overview
- Package Requirements

3.1. Basic Usage

- The `pdftex` Option
- The `dviwindo` Option
- The Language Option

3.2. On-line Exercises

- The `forpaper` option
- The `nosolutions` option
- The `solutionsafter` option
- Moving the Solution Set
- Redesigning the exercise Environment

3.3. Quizzes with Immediate Response

3.4. Quizzes with Solutions

- The `questions` Environment
- The `forpaper` option
- The `solutionsafter` Option
- Moving the Solution Set
- Redesigning the `shortquiz` Environment

3.5. Graded Quizzes using JavaScript

- Link-Style Quizzes
- Form-Style Quizzes

3.6. Correcting the Quizzes with JavaScript

- The `nocorrections` Option

3.7. How to Modify the quiz Environment

- The Quiz Titles
- The check appearance
- The ‘Correction’ Button
- The Score Field

3.8. Greater Customization with \TextField

- Admissible key-value pairs of \TextField
- Some Examples of \Textfield
- Modifying \LayoutTextField

4. List of Options

Solutions to Exercises

Solutions to Quizzes

References

1. Introduction

This document describes how to use the `web` and `exerquiz` packages for L^AT_EX. It is intended that the output from a `.tex` source file that uses these packages be converted to Adobe's *Portable Document Format* (PDF). The full features and capabilities of the `web` and `exerquiz` packages can only be seen/experienced by viewing the prepared document using Adobe's *Acrobat Reader*.

- **Goals**

The goals of this work are two-fold: (1) create an attractive, easy-on-the-eye page layout suitable for the WWW (or classroom/conference presentations) (the `web package`); and (2) make it very easy (for educators) to create interactive exercises and quizzes in the PDF format (the `exerquiz package`).

These two packages should be useful to educators who want to post interactive materials for their students on the WWW.

Please contact me at dpstory@uakron.edu should you encounter any problems, or have suggestions to make.

- **A Brief History**

The `web` and `exerquiz` packages were written in preparation for a two-day workshop on \LaTeX /PDF that I gave at the **College of the Redwoods**, April 30-May 1, 1999, at the invitation of David Arnold. The workshop forced me to take many of the basic macros that I had developed in plain \TeX and convert them to \LaTeX .

Significant additions to the `exerquiz` immediately following the 20th Annual Conference of the \TeX User's Group (TUG), in August, 1999, Vancouver, British Columbia, which I attended.

- **Thanks**

Noel Vaillant, www.probability.net, deserves my thanks for his enthusiasm for the `web` style file and his initial work on it inspired me to make a serious effort at writing a \LaTeX package.

Thanks also goes out to Jean-Michel Sarlat for writing a French version of the `web` and `exerquiz` packages, see his **Syracuse** Web site. He urged me to include a language option. Thanks also goes to Michael Wiedmann who suggested a language option many months earlier, but

I'm afraid, it landed on deaf ears at the time. These two provided the translations for the `french` and `german` options. (January 1, 2000)

Now, I really must get back to work. $\mathcal{D}\mathcal{S}$

2. The Web Package

The purpose of the `web` package is to create a page layout for documents meant for screen presentation, whether over the WWW or classroom/conference presentations, in PDF. Such documents are *not* (necessarily) *intended to be printed*; consequently, the page layout is, in some sense, optimized for screen viewing.

- **Overview**

The `web` package redefines `\maketitle` and `\tableofcontents` in a more web friendly way; it colors the section headings, and inserts `\bullets` (●) at the `\subsubsection` level. This, to my eyes, is very attractive. Additionally, certain navigational devices—a navigational bar and some direction irons—are included in the package.

There are options for a small collection of drivers: `dvipsone`, `dvips` and `pdftex`. The language option redefines certain language dependent elements of the package to other languages, currently, `french`, `german` and `norsk`. There is even an option for reformatting the `web` style to a print format!

The capabilities of the `web` package and its options are discussed below. Any comments and suggested improvements (new features) would be greatly appreciated.

● Package Requirements

The `web` package was designed for screen presentations tutorials, such as classroom or conference lectures, short technical articles, etc.; consequently, the `article` class of L^AT_EX seems to be a sufficient for these purposes. Though you can use `web` with any of the standard classes that define the `\section`, `\subsection` and `\subsubsection` commands, the package is really meant to be used with the `article` class. It is **strongly** suggested!

The package heavily depends on Sebastian Rahtz' `hyperref` package

(now maintained and developed by Heiko Oberdiek). The `web` package was developed using version 6.56 of `hyperref`. Using prior versions of `hyperref` *may* lead to successful compilation—no guarantees offered. It is best to work with the most recent version of `hyperref`.

The `color` and `amssymb` packages are also required. The former is for obvious reasons, the later is to provide certain navigational symbols when the `navibar` option is invoked.

Finally, to create quality PDF document, type 1 fonts *must* be used. Fortunately, type 1 fonts in the Computer Modern font set are freely available, and come with all the major freeware, shareware and commercial \TeX systems. If you haven't done so already, learn how to use the type 1 fonts.

In this regard, I have written an article that may be of interest to you entitled “*Using \LaTeX to Create Quality PDF Documents for the WWW*”, see reference [9].

2.1. Basic Usage

To use the `web` package, insert into the preamble of your document the following:

```
\usepackage[<driver_option>]{web}
```

The optional argument above defines the driver to be used; for example,

```
\usepackage[dvipsones]{web}
```

Currently, the `web` package supports five drivers: `dvipsones`, the `dvi-to-ps` converter by Y&Y, Inc., (<http://www.yandy.com/>); `dviwindo`, Y&Y's `dvi`-previewer; `dvips`, the freeware `dvi-to-ps` converter; `pdftex`, the `tex-to-pdf` application; and `dvipdfm`, the `dvi-to-pdf` application by Mark Wicks, (<http://odo.kettering.edu/dvipdfm/>).

- ▶ The package has been tested using `\documentclass{article}` and it is *strongly* recommended that this class be used.
- ▶ The cropping parameters assumes `letter` paper size. When using option `dvips`, use the following command line switch `-t letter`. My

command line for `dvips` is

```
dvips.exe -j0 -Pcm -Pams -t letter -o <foo>.ps <foo>.dvi
```

The default paper setting for `dvips` is A4. If you find changing the command line inconvenient, you can include the command

```
\hypersetup{pdfpagescrop=53 486 389 754}
```

in the preamble of your document.

• Setting the Driver Option

You can set your driver option in one of three ways:

- Pass as a local option:

```
\usepackage[<driver_option>]{web}
```
- Pass as a global option:

```
\documentclass[<driver_option>]{article}
```
- Create the file `web.cfg` with the single command in it:

```
\ExecuteOptions{<driver_option>}
```

Place the file `web.cfg` in any folder where L^AT_EX looks for input files. Then, you need only type `\usepackage{web}`.

Where `<driver_option>` is any of the following options: `dvipsone`, `dviwindo`, `dvips`, `pdftex`, or `dvipdfm`.

The macros of the `web` package have been extensively tested using the Y&Y T_EX System (www.yandy.com) for the `dvipsone` and `dviwindo` options and a MikT_EX System (www.miktex.de) for the `dvips`, `pdftex` and `dvipdfm` options.

• The `pdftex` Option

The `pdftex` option requires `hyperref` version 6.60 or greater. (The most recent version is preferred.) The `web` and `exerquiz` packages will probably run correctly with slightly earlier versions, but you may see ‘underfull hbox’ error messages.

The `web` package uses the AMS Font set for the black triangle that appears in the navigational bar, see [Section 2.4](#).

The AMS Font sets comes with MikT_EX, if you don’t have pdfT_EX configured the use them, here are some instructions that work for the

MikTeX system.

1. Copy the file `amsfonts.map` from the folder `texmf/dvips/ams` to the folder `texmf/pdftex/base`. (MikTeX may have restructured the directory tree since the time of this documentation.)
2. Edit the file `pdftex.cfg` to include the lines:

```
map +amsfonts.map      % to get AMSFonts
map  standard.map      % to get dings
```

3. The `pdftex.cfg` file now looks like this:

```
output_format 1
compress_level 9
decimal_digits 2
page_width 210mm
page_height 297mm
horigin 1in          % <-- important, see 4.
vorigin 1in          % <-- important, see 4.
map standard.map
map +cm.map
```

```
map +amsfonts.map
```

4. Take note of the settings of `horigin` and `vorigin`. Set these two to 1 in. This will align the text on the page where the cropping numbers expects the text to be; the result will be a properly cropped page.

- **The `tight` Option**

In an effort to compact more material per page, I've introduced a `tight` option. When this option is used, many of the list parameters are redefined so that is not so much space around these environments, and between items.

```
\usepackage[<driver_option>,tight,<otther_options>]
```

This screen version of this manual was typeset with the `tight` option, the print version was typeset without it.

2.2. Hyperref Options

The web package loads `hyperref` into the document and sets some selected options of that package; therefore, including the `hyperref` package is not needed in the preamble of your own document.

Any additional `hyperref` options that are needed can be introduced into the package using `hyperref`'s `\hypersetup` macro, for example,

```
\documentclass{article}
\usepackage[dvipsone]{web}    % or dvips or pdftex
```

```
% Declare additional hyperref options using \hypersetup
\hypersetup{pdfpagemode=None,bookmarksopen=false}
```

Documentation of the options that `hyperref` recognizes can be had by either \LaTeX ing the file `hyperref.dtx`, or by getting a copy of the *The \LaTeX Web Companion* [4] by Michel Goossens *et al.*

2.3. The Title Page and TOC

The title page is constructed from the values of the macros: `\title`, `\author`, `\university`, `\email`, and `\version`. The values of some of the macros `\title` and `\author` are also transferred to the PDF-DocInfo section of the Acrobat Reader/Exchange.

Additionally, the values of `\subject` and `\keywords` are inserted into the PDFDocInfo section.

- **Basic Information Macros**

Just fill in the values of all the basic macros briefly described above. For example, the following is a copy of the title information for this document:

```
% \title,\author,\subject,\keywords are sent to DocInfo
\title{The Web and Exerquiz Packages Manual of Usage}
\author{D. P. Story}
\subject{How to create on-line exercises and quizzes}
\keywords{LaTeX,hyperref,PDF,exercises,quizzes}
```



```
% \university,\email,\version are used only on title page
\university{THE UNIVERSITY OF AKRON\\
  Mathematics and Computer Science}
\email{dpstory@uakron.edu}
\version{1.30}
\copyrightyears{1999-2000}
```

► The `\title`, `\author`, `\subject`, `\keywords` are a convenient way of entering information in the Document Information fields—see

File > Document Info > General ... (Ctrl+D)

in the Acrobat Reader/Exchange.

If `\title` contains control sequences that do not expand to the Standard PDFDocEncoding character set, Distiller will be thrown into a tailspin; `hyperref` defines the `\texorpdfstring` macro¹ to avoid these kinds of problems. For example,

```
\title{The \texorpdfstring{\e~x$}{EXP} Function}
```

¹The code for handling PDFDocEncoding for `hyperref` is due to Heiko Oberdiek

The first argument is the one that is typeset (on the title page, the title of the document will be ‘The e^x Function’); the second argument is the one that is sent to the title field of DocInfo in the Acrobat Reader (and will read ‘The EXP Function’).

Of all the **Basic Information Macros**, use `\texorpdfstring` only with the `\title`, `\author`, `\subject` and `\keywords`, all of which are used in the DocInfo field of the Acrobat Reader.

▶ `\texorpdfstring` works for `\section`, `\subsection`, etc. as well.

Having entered the information you can now type the standard sort of L^AT_EX commands of `\maketitle` and `\tableofcontents`:

```
\begin{document}
\maketitle
\tableofcontents
...
\end{document}
```

▶ Use the file `webeqtst.tex`, which comes with the distribution, as a prototype or template for your own document.

- **Redesigning `\maketitle`**

The arguments of the **Basic Information Macros** macros, as just discussed, are used to define text macros with no parameters; for example, when you type `\title{Web Package}`, the macro `\title` takes its argument and defines a macro `\webtitle` that expands to ‘Web Package’.

You can redesign the title page to suit your needs simply by redefining the `\maketitle`: rearrange the macros listed in the second column of **Table 1** on the page, or include a graphic, or change the background color. To redefine `\maketitle`, use the commands:

```
\renewcommand\maketitle{...your design...}
```

See the definition of `\maketitle` in the `web.sty` file for an example.

When making the design, it is useful to know that the `web` package uses `\hypertarget` to create a named destination, ‘`webtoc`’, in the table of contents, Use this `webtoc` to jump to the table of contents using the macro `\hyperlink`.

Lastly, I have included a macro, `\optionalpagematter`, you can

This macro	defines this macro
<code>\title</code>	<code>\webtitle</code>
<code>\author</code>	<code>\webauthor</code>
<code>\subject</code>	<code>\websubject</code>
<code>\keywords</code>	<code>\webkeywords</code>
<code>\university</code>	<code>\webuniversity</code>
<code>\email</code>	<code>\webemail</code>
<code>\version</code>	<code>\webversion</code>
<code>\copyrightyears</code>	<code>\webcopyrightyears</code>

Table 1: The Basic Information Macros

use to include additional material on the title page. Here is an example of usage:

```
\renewcommand\optionalpagematter{\vfill
  \begin{center}
  \fcolorbox{blue}{webyellow}{
  \begin{minipage}{.67\linewidth}
```

```
\noindent\textcolor{red}{\textbf{Abstract:}} This
file attempts to teach you how to create a simple
\LaTeX\ document.
\end{minipage}}
\end{center}}
```

The above definition will create the framed box seen below.

Abstract: This file attempts to teach you how to create a simple L^AT_EX document.

The `\optionalpagematter` appears just below the `\webauthor` and above the directory listing. See the sample file `webeqtst.tex` for an example of this feature.

► Of course, you can rearrange everything to suite your taste.

• The `nodirectory` option

The inclusion of `\tableofcontents` is optional. The article you write may be short, or perhaps it may just be a collection of exercises and quizzes. In this case, you may not want a table of contents.

If you do not want a table of contents, you would not include `\tableofcontents` just after `\begin{document}`. Without a table of contents, you may as well turn off the directory listing on the cover page as well. Use the `nodirectory` option to do this:

```
\usepackage[dvips,nodirectory]{web} % dvipsone, pdftex
```

The directory listing does not appear on the title page.

- **The `latexoc` option**

If you don't like the default design for the table of contents, you can always recover the standard L^AT_EX table of contents by using the `latexoc` option with the `web` package:

```
\usepackage[latexoc]{web}
```

Should you want to go with this option, you might consider including

```
\hypersetup{linktocpage}
```

Look at the table of contents with and without this `hyperref` option to decide which you prefer.

2.4. Navigational Aids

The web package offers a couple of navigational aids to help you move around: the `navibar` Option, and some **direction icons**.

- **A Navigational Bar**

Use the `navibar` option of `web` to add a navigational toolbar, as seen at the bottom of this page. Usage:

```
\usepackage[<driver_option>,navibar]{web}
```

the result is the navigation bar you see at the bottom of the page.

▶ The toolbar can be turned on or off by the following commands: `\NaviBarOn` and `\NaviBarOff`. The navigational toolbar at the bottom of the page was generated by the `\NaviBarOn`. `\NaviBarOff` was placed on the next to turn off the bar.

- **Direction Icons**

The up arrow you see in the upper right-hand corner was constructed using colored rules and the AMS symbol font, `amssymb`. The uparrow



icon was produced by the command:

```
\insertnaviiconhere{\ArrowUp{\hyperlink{webtoc}}}
```

Or, more generally,

```
\insertnaviiconhere{\ArrowUp{link_command}}
```

```
\insertnaviiconhere{\ArrowDown{link_command}}
```

This will insert direction icons on the current page (I hope).

If you want a running direction icon you can use

```
\insertnaviiconhereafter{\ArrowUp{link_command}}
```

or

```
\insertnaviiconhereafter{\ArrowDown{link_command}}
```

▶ To discontinue a running arrow icon type

```
\defaultpageheader
```

one the page you want the arrow(s) to disappear.

2.5. The Language Options

The language options redefine all of the language dependent text macros that appear on the title page, in the table of contents and in the running headers. Invoke these options in the usual way:

```
\usepackage[<driver_opt>,<lang_opt>]{web}
```

Where, <lang_opt> is one of the following: `french`, `german`, or `norsk`. The files `webeqtst-fr.pdf`, `webeqtst-de.pdf` and `webeqtst-no.pdf` are the demonstration file `webeqtst.pdf` with the `french` and `german` options invoked, respectively.

The `web` and `exerquiz` packages seem to be compatible with the `babel` package; you can use

```
\documentclass{article}
\usepackage[french]{babel}
\usepackage[dvips,french]{web}
\usepackage{exerquiz}
```

subject to the usual restrictions on these language packages. (Don't use characters declared active by these languages within a `\label`, or

as a field name for a quiz.

The translations for the `french` option is due to the tremendous efforts of Jean-Michel Sarlat, and Michael Wiedmann did the translations for the `german` option.

2.6. The `forpaper` option

Some people may want to create exercises using the `exercise` environment for a paper document; this is certainly possible.

To do this, you simply reset `\textwidth` and `\textheight` of the document and use the `forpaper` option with `exerquiz`. Here is a sample listing:

```
\documentclass{article}
\usepackage[dvipsone,nodirectory,latextoc]{web}
\usepackage[forpaper]{exerquiz}
% forpaper, change some parameters
\ifeqforpaper
% reset \textheight
```

```
\setlength\textheight\oldltxttextheight  
% reset cropping (for letter size paper)  
\hypersetup{pdfpagescrop={0 0 612 792}}  
\fi
```

The `forpaper` options does two things: (1) It redefines the basic color macros `\textcolor`, `\color` and `\pagecolor` so that they do nothing; and (2) changes the `\newpage` command to `\par\medskip` at the end of each solution—we don't want to waste paper now do we.

Notice also (1) the boolean switch `\ifeqforpaper`, which you are free to use to refine the look your `forpaper` version; and (2) the length `\oldltxttextheight` (and `\oldltxttextwidth`), which equals the `\textheight` (respectively, `\textwidth`) as set by the class file, such as the `article` class.

3. The Exerquiz Package

• Overview

The `exerquiz` package provides environments for creating the following interactive elements in a PDF document.

- On-line Exercises: Macros for creating on-line exercises.
- Quizzes with Immediate Response: Macros for creating interactive quizzes with immediate feedback.
- Quizzes with Solutions: Macros for creating interactive quizzes with immediate feedback and a link to the solutions to the quizzes.
- Graded Quizzes using JavaScript: Macros for creating quizzes graded by JavaScript, with an option to have the quizzes corrected using JavaScript.

The `exerquiz` provides the above listed environments for the `dvipson`, `dvips`, `pdftex` and `dvipdfm` options; only the `exercise` environment is available with the `dviwindo` option.

There are options for reformatting the exercises to a print format; for excluding the solutions to the exercises; for writing the solutions to the exercises so they follow the question; for different languages, and much more.

The `exerquiz` also allows you to rearrange the order and location of the solutions to the exercises and quizzes; to redefine many running headers; to customize the exercises and quizzes; and to use the `exercise` environment to create a new environment with its own counter—or with no counter at all.

All the above mentioned macros and the options of the package are discussed in this section.

• Package Requirements

The `exerquiz` package is independent of the `web` package; however, `exerquiz` utilizes `hyperref` just as `web` does. Use the latest version of `hyperref`. In addition to the `color` package, also used by `web`, `exerquiz` also uses the `verbatim` package, this is used to write verbatim solutions to exercises and quizzes to certain auxiliary files.

The `exerquiz` package uses a feature of `hyperref` that `web` does not use, the *form features* of PDF. For the interactive features to properly work, use [Acrobat Reader 3.0](#) with [Forms 3.5](#) or higher. (Acrobat Reader 4.0 or above preferred.)

3.1. Basic Usage

Place in the preamble of your document

```
\usepackage{exerquiz}
```

► Use `exerquiz` with the `web` package:

```
\usepackage[<driver_option>,<more_options>]{web}  
\usepackage{exerquiz}
```

No driver option with `exerquiz` is needed if you are using the `web` package. (The driver options for the `web` package are `dvipsone`, `dvips`, `pdftex`, `dvipdfm` and `dviwindo`.)

For the `dvipdfm` option to work properly you will need `dvipdfm`, version 0.12.7b or later, and `hyperref`, version 6.68a or later.

- ▶ Use `hyperref` and `exerquiz` with either `dvipsone` or `dvips`:

```
\usepackage[<driver_options>,<more_options>]{hyperref}  
\usepackage{exerquiz}
```

Permissible driver options are `dvipsone` and `dvips`.

- ▶ Use `hyperref` and `exerquiz` with `pdftex`, `dviwindo` or `dvipdfm`

```
\usepackage[<driver_options>,<more_options>]{hyperref}  
\usepackage[<driver_option>]{exerquiz}
```

See the next few paragraphs for more details.

• The `pdftex` Option

The `exerquiz` package is independent of the `web` package. Therefore, you can create your own page layout package and use `exerquiz` to help you create exercises and quizzes. Of course, `hyperref` must be used.

Should you want to use the `exerquiz` package using pdf \TeX without the `web` package, use the `pdftex` option:

```
\usepackage[pdftex,<more options>]{hyperref}  
\usepackage[pdftex]{exerquiz}
```

In particular, `pdfscreen`², a screen design package written for pdfTEX by C. V. Radhakrishnan, has been tested and works correctly with `exerquiz`. For example,

```
\usepackage[screen,article,sidebar]{pdfscreen}
\usepackage[pdftex]{exerquiz}
```

See the sample file `eq_pdfs.tex` already set up for use with `pdfscreen`, obtained by downloading the zipped file `eq_pdfs.zip`.

- **The `dviwindo` Option**

Beginning with version 1.3 of `web` and `exerquiz`, `dviwindo` (the `.dvi` previewer by **Y&Y, Inc.**) is supported. This means that hypertext links will be active from within the `dviwindo` previewer—as will as from within the Acrobat Reader after the file has been converted to PDF.

Should you want to use the `exerquiz` package without the `web` package, in this case, the usage is

```
\usepackage[dviwindo,<more_options>]{hyperref}
```

²CTAN:macros/latex/contrib/supported/pdfscreen


```
\usepackage[dviwindo]{exerquiz}
```

► **Important Note:** *Only* the `exercise` environment (the material described in [Section 3.2](#)) is supported by these two options. None of the quiz environment can be used with these two options at this time.

• The Language Option

The `language option`, available in the `web` package, can be invoked even when the `web` package is not used.³ Currently, only `french`, `german` and `norsk` are the supported options. For example, with `hyperref`, you could use:

```
\usepackage[<driver_option>,<more_options>]{hyperref}  
\usepackage[<driver_option>,french]{exerquiz}
```

Where `<driver_option>` is any of the supported drivers: `dvipson`, `dvips`, `pdftex`, `dviwindo` or `dvipdfm`. *Note:* the `<driver_option>` is not needed with the `exerquiz` package with `dvipson` or `dvips`.

³Otherwise, the `language option` is introduced as an option of the `web` package.

3.2. On-line Exercises

The `exerquiz` package defines `exercise` and `solution` environments, the latter being nested inside the former. With these environments, you can create questions (exercises) with solutions. A hypertext link is created to connect the exercise with the solution.

The syntax for the `exercise` and `solution` environments is as follows:

```
\begin{exercise}
Your Question.
\begin{solution}
The Solution to Your Question
. . . . .
. . . . .
. . . . .
\end{solution}
\end{exercise}
```

Here is an example of the usage.

EXERCISE 1. Evaluate the integral $\int x^2 e^{2x} dx$.

► Questions and solutions are kept together *à la Knuth*. The solutions are written to the file `\jobname.sol` verbatim then input back using the macro `\includeexersolutions`.

► You can redefine the counter to include the section number. For example.

```
\renewcommand{\theeqexno}{\thesection.\arabic{eqexno}}
```

can be placed in the preamble of your document. In this case, the above exercise would appear as EXERCISE 3.1.

► The usual cross-referencing mechanisms for L^AT_EX, i.e., using `\ref` and `\pageref`, work as expected.

For example, the label `'\label{ex:int}'` was placed just after `\begin{exercise}` on the previous page, let us now reference Exercise 1, on page 35.

```
let us now reference Exercise~\ref{ex:int},
on~\pageref{ex:int}.
```

Of course, the nicer looking variations can be done as well: For example, see [EXERCISE 1](#).

```
\hyperref[ex:int]{\textsc{Exercise~\ref*{ex:int}}}
```

The *-form of `\ref` was used to turn off the redundant link creation. (`hyperref` would normally make the `\ref` macro into a link.)

- **The `forpaper` option**

The `forpaper` option, also available in the [web](#) package, is needed in the `exerquiz` package if your are using `exerquiz` without `web`. The option is invoked in the usual way

```
\usepackage[<options>]{hyperref} % or pdfscreen
\usepackage[forpaper]{exerquiz}
```

See the discussion of the [forpaper](#) on page [26](#) given earlier.

- **The nosolutions option**

Some educators may initially want to post a series of exercises on the Web without the solutions. Then, at a later date, repost the exercises with the solutions included. For this application there is the `nosolutions` option for the `exerquiz` package.

```
\documentclass{article}
\usepackage[pdftex]{web} % dvipsone or dvips
\usepackage[nosolutions]{exerquiz}
```

For this kind of application, it might make sense to publish the exercises with the `forpaper` option.

- **The solutionsafter option**

For additional flexibility with how you want the solutions to the exercises presented, there is a `solutionsafter` option with `exerquiz`. Should you invoke this option

```
\documentclass{article}
\usepackage[dvipsone]{web} % dvips or pdftex
```

`\usepackage[solutionsafter]{exerquiz}`

the solutions to the exercises appear just *after* the exercise question. For example

EXERCISE 2. Let V be a vector space, show that the zero vector, $\mathbf{0}$, is unique.

Solution: Let $\mathbf{0}'$ be a vector that satisfies the axiom of being a zero of the vector space V . We want to show $\mathbf{0} = \mathbf{0}'$. Since $\mathbf{0}$ is a zero, we have $\mathbf{0} + \mathbf{0}' = \mathbf{0}'$. But we are assuming $\mathbf{0}'$ is a zero vector as well, hence, $\mathbf{0}' + \mathbf{0} = \mathbf{0}$. Finally,

$$\mathbf{0}' = \mathbf{0} + \mathbf{0}' = \mathbf{0}' + \mathbf{0} = \mathbf{0}$$

and this completes the proof.

Exercise 2

The option `solutionsafter` is global; all exercises will be typeset this way—unless you change it within the document using the macros `\SolutionsAfter` and `\SolutionsAtEnd`. This manual was typeset without the `solutionsafter` option. The above example was typeset as follows:

```
\SolutionsAfter % show solution following exercise
\begin{exercise}
Let  $V$  be a vector space, show ...
\begin{solution}
.....
\end{solution}
\end{exercise}
\SolutionsAtEnd % turn back on solutions at of document
```

Normally, a typical document might have all solutions at the end of the document (the default behavior), or all solutions following each exercise (`solutionsafter` option). Mixtures of these two types can be obtained by using the two commands `\SolutionsAfter` and `\SolutionsAtEnd`.

This feature might be an easy way of typesetting examples. See the paragraph ‘[Redesigning the exercise Environment](#)’ on page 40 for an example of setting up an `example` environment.

▶ See the files `webeqtst.pdf` and `hw02.pdf` for examples

• Moving the Solution Set

The solution set, by default, comes last in the file. You can move its positioning by including the command `\includeexersolutions` at any point *after* the last exercise. You'll note, that I have moved the solutions in this file before the **References** section, as indicated, for example, by its position in the table of contents.

• Redesigning the exercise Environment

You can customize the `exercise` environment to suite your own needs. To customize, you need to change some or all of the following six commands. In the listing below, the L^AT_EX definition of each follows a short description.

1. `\exlabel`: This command expands to the name of the exercise label, the default string is 'Exercise'.

```
\newcommand\exlabel{Exercise}
```

2. `\exlabelformat`: Typesets the exercise label; use it to introduce additional type style such as boldface, italic, small caps, etc.


```
\newcommand\exlabelformat{%
    {\scshape\exlabel\ \theeqexno.}}
```

3. `\exlabelsol`: Expands to the name of the exercise label in the solutions section. Usually its value is the same as `\exlabelsol`.

```
\newcommand\exlabelsol{\exlabel}
```

4. `\exsllabelformat`: The format of the solutions label, the default is `'\bfseries\exlabel'`.

```
\newcommand\exsllabelformat
    {\noexpand\textbf{\exlabelsol\ \theeqexno.}}
```

5. `\exrtnlabelformat`: This is the label you click on to return from the solution of the exercise.

```
\newcommand\exrtnlabelformat{\exlabelsol\ \theeqexno}
```

6. `\exsectitle`: The section title of the solutions to the exercises.

```
\newcommand\exsectitle{Solutions to \exlabel s}
```

7. `\exsecrunhead`: The running header for the solution section for the exercises.

```
\newcommand\exsecrunhead{\exsectitle}
```

► The counter `eqexno` is used to count exercises. When the `exercise` environment starts, this counter is incremented. Normally, the values of this counter figures into the definitions of `\exlabelformat`, `\exsllabelformat` and `\exrtnlabelformat`. Still, the use of `eqexno` is optional; for example, you might want to state a problem just as ‘Special Exercise’, without an associated exercise number.

Below is an example of redefining the `exercise` environment. We define a `problem` environment based on the `exercise` environment.

```
\newenvironment{problem}{%
\renewcommand\exlabel{Problem}
\renewcommand\exlabelformat{\textbf{\exlabel\ \theeqexno.}}
\renewcommand\exsllabelformat
  {\noexpand\textbf{\exlabel\ \theeqexno}}
\renewcommand\exrtnlabelformat{${\blacktriangleleft}$}
\renewcommand\exsecrunhead{\exsectitle}
\begin{exercise}}%
{\end{exercise}}
```

See any standard L^AT_EX reference on how to define a new environment, for example [2].

Here is an example of the new `problem` environment:

Problem 3. This is a question.

The code for this problem was simply:

```
\begin{problem}
This is a question.
\begin{solution}
This is the solution.
\end{solution}
\end{problem}
```

► Two of these commands must be handled with special care, they are `\exsllabelformat` and `\exrtnlabelformat`; formatting such as `\textbf` or `\scseries` must be preceded by a `\noexpand`. These commands are written to a file, and must be prevented from expanding.

When you use the `exercise` environment, the counter `eqexno` is automatically incremented by default. The `exercise` does have an

optional argument for inserting your own counter.

```
\begin{exercise}[<ctr>]
.....
\end{exercise}
```

Where `<ctr>` is a counter already defined. This option is useful if you want to use the `exercise` environment to create a new environment with its own numbering scheme, as the following example illustrates.

This example demonstrates how to define an `example` environment with its own counter. For examples, we don't want the solutions to appear at the end of the file, so we'll use `\SolutionsAfter` and `\SolutionsAtEnd`. All changes are local.

```
% put a counter in preamble
\newcounter{exampleno}

\newenvironment{example}{%
\renewcommand\exlabel{Example}
\renewcommand\exlabelformat
  {\textbf{\exlabel\ theexampleno.}}
```

```
\renewcommand\exrtnlabelformat{${\square$}  
\SolutionsAfter  
\begin{exercise}[exampleno]}%  
{\end{exercise}  
\SolutionsAtEnd}
```

Now we simply type

```
\begin{example}  
What is  $2+2$ ?  
\begin{solution}  
It is well known that  $2+2=4$ .  
\end{solution}  
\end{example}
```

to obtain

Example 1. What is $2 + 2$?

Solution: It is well known that $2 + 2 = 4$.

□

Example 2. What is $2 + 2$?

Solution: It is well known that $2 + 2 = 4$. □

The changes are local to the new `example` environment. If we have another exercise, we get a correctly numbered exercise.

EXERCISE 4. What is $2 + 2$?

▶ The command `\exsolafter` typesets the solution label to the exercise in the case the `solutionsafter` option is in effect. The default value of `\exsolafter` is `\textit{Solution}:/` You can redefine it as follows:

```
\renewcommand\exsolafter{\textsl{L}"osung}:
```

This redefinition yields:

Example 3. What is $2 + 2$?

Lösung: It is well known that $2 + 2 = 4$. □

▶ There is a special option to the `exercise` environment as well,

```

\begin{exercise}[0]
.....
\end{exercise}

```

When the optional argument is 0 rather than a counter. In this case, no counter is associated with the environment. For example,

```

\newenvironment{project}{%
\renewcommand\exlabel{Project}
\renewcommand\exlabelformat{\textbf{\exlabel. }}
\renewcommand\exsllabelformat
  {\noexpand\textbf{\exlabel\ Hint:}}
\renewcommand\exrtnlabelformat{${\blacktriangleleft}$}
\begin{exercise}[0]}%
{\end{exercise}}

```

Thus, we obtain,

Project. Find a shorter proof of FERMAT'S LAST THEOREM. Do not look at the project hints until you have finished the project.

The code:

```
\begin{project}
Find a shorter proof of \textsc{Fermat's Last Theorem}. Do not
look at the project hints until you have finished the project.
\begin{solution}
There, you didn't need my help after all.
\end{solution}
\end{project}
```

Note that the solutions are typeset at the end of the file in the ‘Solutions to Exercises’ section. At this time, there is no feature for sorting out these different types of environments; they are all **exercise** environments, which is what they are.

▶ Finally, see the sample file `hw01.tex` that illustrates how to change all the labels. The file also demonstrates how `web` and `exerquiz` can be used to post problems on the Internet, or on paper, with or without solutions included.

3.3. Quizzes with Immediate Response

The `shortquiz` environment is used to create a multiple choice question with immediate response. The environments allow redefinition to customize the look you the quizzes. (See the paragraph entitled ‘[Redesigning the shortquiz Environment](#)’ on page 60.)

The syntax for this environment is as follows:

```

\begin{shortquiz}                % begin shortquiz
...Question goes here...
\begin{answers}{num_cols}       % begin proposed answers
...
\Ans0 <an incorrect answer> &   % a wrong answer
...
\Ans1 <a correct answer> &      % the right answer
...
\end{answers}                   % end listing of answers
\end{shortquiz}                 % end shortquiz

```

The parameter `num_cols` is the number of columns you want to typeset

your multiple choice responses in.

This type of quiz is suitable as a quiz in-line question of the reader, perhaps after explaining some concept. Quizzes can be used to direct the reader's attention to an important point.

► Here is an example of the `shortquiz` environment. Responses are graded without comment using JavaScript.

Quiz Which of the following is the $\frac{d}{dx}\sin(x^3)$?

(a) $\sin(3x^2)$ (b) $\cos(x^3)$ (c) $3x^2 \cos(x^3)$ (d) $3x^2 \cos(3x^2)$

The verbatim listing follows:

```
\begin{shortquiz}           % begin shortquiz environment
Which of the following is the  $\frac{d}{dx}\sin(x^3)$ ?
\begin{answers}{4}         % 4 columns of answers
  \Ans0  $\sin(3x^2)$  & % \Ans0 is a false answer
  \Ans0  $\cos(x^3)$  &
  \Ans1  $3x^2\cos(x^3)$  & % \Ans1 is the correct answer
```

```
\Ans0  $3x^2\cos(3x^2)$ 
\end{answers}           % end answers environment
\end{shortquiz}        % end shortquiz environment
```

The `answers` environment uses `tabular` with `p{<width>}` to set up the columns. The `\parboxes` are typeset ragged right. For example,

Quiz Which of the following best describes Augustin Cauchy?

- | | |
|--|---|
| (a) He developed the Calculus while his University was closed for the plague. | (b) Given credit for first using the functional notation $f(x)$. |
| (c) He created the “bell-shaped curve” and first used the method of least squares. | (d) He first formulated a precise definition of the limit and continuity of a function. |
| (e) Gave a rigorous definition of the definite integral—an integral that now bears his name. | (f) His notation for the derivative and the integral is used even to this day. |

► See the sample files `webeqtst.tex` and `qz01.tex` for examples. The later file gives examples of how to redefine some of the standard `shortquiz` labels.

3.4. Quizzes with Solutions

Another type of quiz that is easy to implement in PDF is the multiple choice quiz with immediate response with solution given. This too is a `shortquiz` environment:

```
\begin{shortquiz}
...Question goes here...
\begin{answers}[<name>]{<num_cols>}
...
\Ans0 <an incorrect answer> &
...
\Ans1 <a correct answer> &
...
\end{answers}
```

```

\begin{solution}
...Solution to correct answer goes here...
\end{solution}
\end{shortquiz}

```

The `<name>` is a name used to create a hypertext jump to the solution; `<name>` will be the “named destination.” As before, `<num_cols>` is the number of columns to typeset the answers in.

The following example illustrates the quiz with solution.

Quiz Define a function $f(s) = 4s^3$ and another function $F(t) = t^4$. Is F an antiderivative of f ?

(a) Yes (b) No

The verbatim listing:

```

\begin{shortquiz}
Define a function  $f(s)=4s^3$  and another
function  $F(t)=t^4$ . Is  $F$  an antiderivative of  $f$ ?
\begin{answers}[quiz:anti]{4}

```

```
\Ans1 Yes &\Ans0 No
```

```
\end{answers}
```

```
\begin{solution}
```

The answer is 'Yes'. The definition requires that

$$F'(x) = f(x) \quad \text{for all } x,$$

well, let's check it out.

.....

.....

Therefore,

$$F'(x) = 4x^3 = f(x) \quad \text{for all } x,$$

as required by the definition.

```
\end{solution}
```

```
\end{shortquiz}
```

- **The questions Environment**

The `questions` environment was designed to work with the `quiz` environment—taken up in [Section 3.5](#) below—but it works equally well with `shortquiz`.

Using the `questions` environment, quizzes defined by `shortquiz`, with/without solutions, can be mixed together and combined to make a “mini-quiz”. For example,

Quiz Determine the LCD for each of the following.

1. $\frac{3x}{2y^2z^3} - \frac{2}{xy^3z^2}$.

(a) LCD = $2xy^5z^5$

(c) LCD = $2xy^3z^3$

(b) LCD = $2y^3z^3$

(d) LCD = $2xy^3z^5$

2. $\frac{x+y}{3x^{3/2}y^2} - \frac{x^2+y^2}{6xy^4}$.

(a) LCD = $18x^{3/2}y^4$

(c) LCD = $18xy^4$

(b) LCD = $6x^{3/2}y^4$

(d) LCD = $6xy^4$

The first question is given without a solution, the second has a solution attached to it. An abbreviate verbatim listing follows.

```

\begin{shortquiz}
Determine the LCD for each of the following.
\begin{questions}
\item  $\frac{3x}{2y^2z^3} - \frac{2}{xy^3z^2}$ $.
\begin{answers}2
...
\end{answers}
\item  $\frac{x+y}{3x^{3/2}y^2} - \frac{x^2+y^2}{6xy^4}$ $.
\begin{answers}[quiz:LCB]2
...
\end{answers}
\begin{solution}
If you erred on this one, ... ..
\end{solution}
\end{questions}

```



```
\end{shortquiz}
```

- **The `forpaper` option**

The `forpaper` option has already been described. The solutions to a `shortquiz` questions are not typeset on separate pages, but are separated by a `\medskip`.

Following up on the pretest angle first discussed in the paragraph [Redesigning the `shortquiz` Environment](#), page 60, a single document can be constructed that can be published on-line, or published for paper distribution. This feature may be useful to some educators.

By the way, if you want to create a series of multiple choice questions with solutions, you must make up a lot of named destinations (the optional argument of the `answers` environment). Alternately, you can let L^AT_EX assign the names for you, which provides for you a uniform naming system. You can use `questionno` to do this:

```
\begin{shortquiz} Answer each, then look at solutions.  
  \begin{questions}  
    \item ...
```

```
\begin{answers}[quiz:\thequestionno]{4}
...
\end{answers}
\begin{solution}
...
\end{solution}
\item ...
\begin{answers}[quiz:\thequestionno]{4}
...
\end{answers}
\begin{solution}
...
\end{solution}
\end{questions}
\end{shortquiz}
```

- **The solutionsafter Option**

The `solutionsafter` option works as described for the `exercise` environment. The option just sets a boolean switch. This switch can be controlled locally with the macros `\SolutionsAfter` and `\`. Here is a simple example.

Quiz In what year did Columbus sail the ocean blue?

- (a) 1490 (b) 1491 (c) 1492 (d) 1493

Solution: Columbus sailed the ocean blue in 1492. Some say he discovered San Salvatore, others say he first sighted Cat Island in the Bahamas.

End Quiz

Here, I have surrounded the `shortquiz` environment with the command `\SolutionsAfter` before the environment, and with the command `\SolutionsAtEnd` just after.

This option may be useful in publishing an answer key to a multiple choice quiz. The quiz and solutions can be created together. The quiz can be published, then later, the quiz with complete solutions.

- **Moving the Solution Set**

The solution set, by default, comes last in the file. You can move its positioning by including the command `\includequizsolutions` at any point *after* the last exercise. You'll note, that I have moved the solutions in this file before the **References** section, as indicated, for example, by its position in the table of contents.

- **Redesigning the shortquiz Environment**

You can temporarily change the title for the `shortquiz` environment by redefining the macro `\sqlabel`; for example, the default definition of this macro is

```
\newcommand\sqlabel{\textcolor{red}{Quiz.}}
```

The syntax for redefining `\sqlabel` is

```
\renewcommand\sqlabel{...new code goes here...}
```

You can redefine the *default* label as well; the default label is the title label that `shortquiz` uses when `\sqlabel` is *not present*. The default label is `\eq@sqlabel` and must be redefined using the macro

`\renewcommand`. The best place for this to be done is the preamble. The syntax:

```
\makeatletter      % make 'at'=@ a normal letter
\renewcommand\eq@sqlabel{...new code goes here...}
\makeatother      % make 'at'=@ something special(other)
```

To change the entire document to use ‘Exam’ instead of ‘Quiz’, make the following changes in the preamble:

```
\makeatletter
% change default quiz title to 'Exam'
\renewcommand\eq@sqlabel{\textcolor{red}{Exam.}}
% change quiz solutions return label
\renewcommand\eq@sqsrlrtnlabel{End Exam}
% change solutions label
\renewcommand\eq@sqsllabel{%
  \string\textbf{Solution to Exam:}}
\renewcommand\eq@sqsllsectitle{Solutions to Exams}
% change default running header for solutions
```

```
\renewcommand\eq@qslsecrunhead{Solutions to Exams}
\makeatother
```

► The above commands are ‘global’—they are in effect throughout the entire document. You can temporarily change these labels using the `\sqlabel`, `\sqlrtnlabel`, `\qsllabel` and `\qslsectitle`. Note that you cannot temporarily change `\eq@qslsecrunhead`, the running label—this should be set in the preamble.

Should you want to make a series of multiple choice questions (using the `questions` environment) and combine them into a sort of review or pretest, a useful idea would be to number the solutions. The counter that maintain the question number is called `questionno`. You can then, for example, define

```
\renewcommand\eq@qsllabel{%
  \string\textbf{Solution to Question \thequestionno:}}
```

► See the sample files `webeqtst.tex` and `qz01.tex` for examples. The later file gives examples of how to redefine some of the standard `shortquiz` labels.

3.5. Graded Quizzes using JavaScript

Use the `quiz` environment to create graded quizzes. In this case, several (many) questions are bundled together. The student takes the quiz, his/her responses are recorded by JavaScript. Upon completion of the quiz, the total score is reported to the student.

There are two types of quizzes, the `link-style` and `form-style`. In [Section 3.6](#), we see that the `quiz` environment can also correct the quizzes.

The `quiz` environment consists of a series of nested environments. Inside the `quiz` environment is the `questions` environment (an enumerated list), and within that environment is the `answers` environment. Symbolically, we can express this as

$$\text{quiz} \supseteq \text{questions} \supseteq \text{answers}$$

The term ‘answers’ is, perhaps, not sufficiently descriptive; ‘alternatives’ would be more appropriate, but it requires more typing. :-)

This environment has the following syntax:

```
\begin{quiz}{quizfieldname}
The preamble to the questions goes here.
\begin{questions}
\item State first question....
\begin{answers}4                                % <- 4 column format
\Ans0 ... &\Ans1 ... &\Ans0 ... &\Ans0 ...
\end{answers}
...
\item n th question....
\begin{answers}4                                % <- 4 column format
\Ans0 ... &\Ans1 ... &\Ans0 ... &\Ans0 ...
\end{answers}
\end{questions}
\end{quiz}
```

► Following the quiz, or anywhere in the document, place the macro `\ScoreField`, defined in `exerquiz`, to display the results of the quiz:

```
\ScoreField{quizfieldname}
```


Important. The value of the parameter of the macro `\ScoreField` must match the `quizfieldname` defined in the argument of the quiz environment.

Read the paragraph entitled ‘**The Score Field**’ on page 80 for more details on this macro.

An alternative to `\ScoreField` is to use `\TextField`, a macro defined by `hyperref`. See more details later.

● **Link-Style Quizzes**

This style uses links to record the choices to the alternatives. The link method takes up less space in the pdf file than does the form-style.

Below is an example of a link-style quiz. Instructions should be given to guide the student in operating the quiz correctly.

Instructions. You must click on ‘Begin Quiz’ to initialize the quiz. Not doing so, brings forth an error message. When finished, click on ‘End Quiz’.

Begin Quiz Using the discriminant, $b^2 - 4ac$, respond to each of the following questions.

1. Is the quadratic polynomial $x^2 - 4x + 3$ irreducible?
(a) Yes (b) No
2. Is the quadratic polynomial $2x^2 - 4x + 3$ irreducible?
(a) Yes (b) No
3. How many solutions does the equation $2x^2 - 3x - 2 = 0$ have?
(a) none (b) one (c) two

End Quiz

► While you are taking the test, and before you click on ‘End Quiz’, you can change your answers. A message box comes out, gives you your original choice, and asks you whether you really want to change your answer.

```
\begin{quiz}{qz:discr-1} % qz:discr=quiz field name  
Using the discriminant,  $b^2-4ac$ , respond to each of the
```

following questions.

```
\begin{questions}
```

```
\item Is the quadratic polynomial  $x^2-4x + 3$  irreducible?
```

```
\begin{answers}4
```

```
\Ans0 Yes &\Ans1 No
```

```
\end{answers}
```

```
\item Is the quadratic polynomial  $2x^2-4x+3$  irreducible?
```

```
\begin{answers}4
```

```
\Ans1 Yes &\Ans0 No
```

```
\end{answers}
```

```
\item How many solutions does the equation  $2x^2-3x-2=0$  have?
```

```
\begin{answers}4
```

```
\Ans0 none &\Ans0 one &\Ans1 two
```

```
\end{answers}
```

```
\end{questions}
```

```
\end{quiz}\par
```

```
\ScoreField{qz:discr-1} % matching quiz field name
```

• Form-Style Quizzes

You may be thinking that such a quiz format—one in which the student cannot see his/her choices—is not very good. It is perhaps adequate for two or three quick questions. For a longer quiz format, one would like to see a “checkbox” format. A quiz with a checkbox format can be obtained using the `*-form` of the `quiz` environment:

```
\begin{quiz}*{quizfieldname}  
...same format as before...  
\end{quiz}
```

Here is the same sample quiz with the form-style option. The only change in the code is the insertion of the `*-option`.

Instructions. You must click on ‘Begin Quiz’ to initialize the quiz. Not doing so, brings forth an error message. When finished, click on ‘End Quiz’.

Begin Quiz Using the discriminant, $b^2 - 4ac$, respond to each of the following questions.

1. Is the quadratic polynomial $x^2 - 4x + 3$ irreducible?
Yes No
2. Is the quadratic polynomial $2x^2 - 4x + 3$ irreducible?
Yes No
3. How many solutions does the equation $2x^2 - 3x - 2 = 0$ have?
none one two

End Quiz

▶ Before completing the quiz, a student can easily change alternatives.

▶ This type is more suitable for longer quizzes. The choices student make are visually recorded for the student to review and change before clicking on 'End Quiz'. A partial verbatim listing:

```
\begin{quiz}*{qz:discr-f}
```

Using the discriminant, b^2-4ac , respond to each of the following questions.

```
\begin{questions}
.....
.....
\end{questions}
\end{quiz}\par
\ScoreField{qz:discr-f}
```

▶ See the sample files `webeqtst.tex` and `qz02.tex` for examples. The later file gives examples of how to customize `quiz`.

3.6. Correcting the Quizzes with JavaScript

Beginning with `exerquiz`, version 1.2, you can now correct quizzes created by the `quiz` environment. To correct the quizzes, simply include an additional element into your quiz, a correction button. The correction button is installed using the macro `\eqButton`.

The following is a link-style quiz.

Instructions: Click on ‘Begin Quiz’ to initialize the quiz. When finished, click on ‘End Quiz’. Then, click on the ‘Correct’ button.

Begin Quiz Using the discriminant, $b^2 - 4ac$, respond to each of the following questions.

1. Is the quadratic polynomial $x^2 - 4x + 3$ irreducible?
(a) Yes (b) No
2. Is the quadratic polynomial $2x^2 - 4x + 3$ irreducible?
(a) Yes (b) No
3. How many solutions does the equation $2x^2 - 3x - 2 = 0$ have?
(a) none (b) one (c) two

End Quiz

Legend: A ✓ indicates a correct response; a ✗, indicates an incorrect response, in this case, the correct answer is marked with a ●.

A partial verbatim listing of this quiz follows:

```
\begin{quiz}{qz:discr1-1} Using the discriminant,  $b^2-4ac$ ,  
respond to each of the following questions.
```

```
\begin{questions}
.....
.....
.....
\end{questions}
\end{quiz}

\ScoreField{qz:discr1-1}\raisebox{3.5pt}%
  {\eqButton{qz:discr1-1}}
```

Here, I have aligned the button with the score field. The button needed to be raised up 3.5 pt. The button does not have to be aligned in this way. You can place it wherever you wish.

▶ The macro `\eqButton` is used to create a nice “correction” button. JavaScript is used to correct the quiz. The only required argument is the field label that uniquely defines the field in which the total score is placed. See the section entitled ‘[The ‘Correction’ Button](#)’ on page 77 for more details on how to use this macro.

▶ The `\eqButton` will not work until the user has clicked on ‘End

Quiz’. The user can re-take the quiz simply by clicking on ‘Begin Quiz’, the form fields and JavaScript variables will be cleared.

► It is possible to take this form data and submit it to a CGI script for processing (The data can be saved to a database, for example.) However, there is no built-in capability for this in the `exerquiz` package.

The same quiz can be written in form-style simply by inserting the `*`-option.

Instructions. You must click on ‘Begin Quiz’ to initialize the quiz. Not doing so, brings forth an error message. When finished, click on ‘End Quiz’.

Begin Quiz Using the discriminant, $b^2 - 4ac$, respond to each of the following questions.

1. Is the quadratic polynomial $x^2 - 4x + 3$ irreducible?

Yes

No

2. Is the quadratic polynomial $2x^2 - 4x + 3$ irreducible?

Yes

No

- 3.** How many solutions does the equation $2x^2 - 3x - 2 = 0$ have?
 none one two

End Quiz

► In the partial verbatim listing that follows, notice the field name as been changed from `qz:discr1-l` to `qz:discr1-f`. The different quizzes must have a unique field name.

```
\begin{quiz}*{qz:discr1-f} Using the discriminant,  $b^2-4ac$ ,
respond to each of the following questions.
```

```
\begin{questions}
```

```
.....
.....
.....
```

```
\end{questions}
```

```
\end{quiz}\quad\ScoreField{qz:discr1-f}\raisebox{3.5pt}
{\eqButton{qz:discr1-f}}
```

► Notice that in this example, the `\ScoreField` and the `\eqButton`

are positioned following the ‘End Quiz’; this makes the design more compact and nicer looking.

- **The `nocorrections` Option**

Including the corrections adds quite a bit more JavaScript code to the `.pdf` document, this feature is ‘on’ by default. If you have a document in which you do not want to have the option of offering corrected quizzes, then just specify `nocorrections` is the option list of `exerquiz`.

There are also a couple of macros you can use to override the option switch: `\CorrectionsOn` and `\CorrectionsOff`. Each remains in affect until the other is invoked.

3.7. How to Modify the quiz Environment

There are four ways the appearance of the quizzes can change:

- change the titles
- change the ‘check’ appearance
- change the text field in which the score appears,

- change the appearance of the ‘Correction’ button.

This section discusses each of these four in turn.

● **The Quiz Titles**

It is possible to redefine the quiz titles and other labels if desired.

▶ Locally:

```
\renewcommand\bqlabel{Begin Exam}
\renewcommand\eqlabel{End Exam}
```

▶ Globally:

```
\makeatletter
\renewcommand\eq@bqlabel{Begin Exam}
\renewcommand\eq@eqlabel{End Exam}
\makeatother
```

● **The check appearance**

The appearance of the ‘check’ can be chosen using the `\symbolchoice` macro of the `exerquiz` package. The permissible values for the ar-

gument of `\symbolchoice` are `check` (the default), `circle`, `cross`, `diamond`, `square`, and `star`.

This quiz was generated by inserting `\symbolchoice{diamond}` before the quiz.

Begin Quiz Answer each of the following. Passing is 100%.

1. Who created T_EX?

Knuth

Lamport

Carlisle

Rahtz

2. Who originally wrote L^AT_EX?

Knuth

Lamport

Carlisle

Rahtz

End Quiz

• The ‘Correction’ Button

The ‘Correction’ button is defined by the `\eqButton`, which takes one argument; namely, the field name that contains the total score for the quiz, see the above examples. It also has one optional argument that can be used to modify the appearance of the button.

Local	Global	Default	Description
<code>\BC</code>	<code>\eq@BC</code>	1 0 0	border color
<code>\BG</code>	<code>\eq@BG</code>	.7529 .7529 .7529	face color
<code>\CA</code>	<code>\eq@CA</code>	Correct	button text
<code>\RC</code>	<code>\eq@RC</code>	My Answers!	rollover
<code>\AC</code>	<code>\eq@AC</code>	Please!	pushed text
<code>\DA</code>	<code>\eq@DA</code>	/Helv 10 Tf 0 g	text format
<code>\BS</code>	<code>\eq@BS</code>	/W 1 /S /B	button spec

Table 2: `\eqButton` Parameters

The macros listed in the first column of [Table 2](#) are permitted in the optional parameter field of `\eqButton`—there is no parameter checking, $\text{T}_{\text{E}}\text{X}$ or Distiller/Reader will find the errors.

The meaning of these values is beyond the scope of this manual. Refer to the internet article “Pdfmarks: Links and Forms”, [8], for

details; in particular, see the “Forms” article.

Begin Quiz Answer each of the following. Passing is 100%.

1. What T_EX System does Thomas Esser maintain?

MikT_EX csT_EX teT_EX fpT_EX

2. What T_EX System does Fabrice Popineau maintain?

MikT_EX csT_EX teT_EX fpT_EX

3. What T_EX System does Christian Schenk maintain?

MikT_EX csT_EX teT_EX fpT_EX

End Quiz

The new part is the customized ‘Correction’ button. Here is a verbatim listing of the `\ScoreField` and `\eqButton` macros.

```
\raisebox{-2.5pt}{\ScoreField{qz:TeX-c}\raisebox{3.5pt}{%
  \eqButton[\BC{0 0 1}%   blue border color
  \CA{TeX}%             Button text
  \RC{Users}%           rollover text
  \AC{Group}%           pushed text
```

```
\DA{/TiRo 10 Tf 0 0 1 rg}% times roman, 10 pt, blue text
\BS{/W 1 /S /I}%      % border width 1, inset button
]{qz:TeX-c}}
```

▶ It is important to suppress any spaces following these macro arguments, thus,

```
\eqButton[\BC{0 0 1}\BC{0 0 1}\CA{TeX}\RC{Users}%
\AC{Group}\DA{/TiRo 10 0 0 1 rg}\BS{/W 1 /S /I}]{qz:TeX}
```

would work as well.

▶ This example—as well as others—appears in `webeqstst.tex`, a test file that accompanies the `web/exerquiz` distribution.

• The Score Field

The score field is the text field to which the quiz (and its underlying JavaScript) report the score. This field can be constructed using the `\ScoreField` macro; e.g.,

```
(\ScoreField{qz:TeX-c})
```


We have seen many examples of the use of this macro.

In the simplest case, `\ScoreField` takes one argument, as above, the `quizfieldname` of the associated quiz. It's expansion produces a read-only text field that is 1.5 inches in width with a red border. The initial text that appears in the field is the expansion of the macro `\eqScore`. The expansion of `\eqScore` depends on the language option: `\eqScore` expands to 'Score:' by default, to 'Punkte:' for the `german` option and to 'Score : ' for the `french` option.

The macro `\ScoreField` also has an optional parameter that can be used to modify the appearance of the text field. Should want to change the basic look of the text field produced by `\ScoreField`, just introduce the changes through this optional parameter. For example, the field

was produced by the code

```
\ScoreField[width=2in,bordercolor={0 0 1},  
backgroundcolor=.753 .753 .753,borderstyle=B]{qz:TeX-c}
```

Some of the more useful ‘key-value’ pairs are listed in the series of tables in the paragraph ‘[Admissible key-value pairs of `\TextField`](#)’ on page [83](#); however, you *do not* use the optional parameter field to define either the `name` (this is done with the required parameter) or the `default` value (this is done automatically with `\eqScore`).

The macro `\ScoreField` uses the `hyperref` macro `\TextField` to produce the score field. If you want a score field fancier than the ones that `\ScoreField` can produce just revert to using the `\TextField` macro directly. (See [Section 3.8](#).)

► See the file `qz02.tex` for details and examples of how to modify the quiz titles. The language files, e.g., `eqfr.def` and `eqde.def`, demonstrate how to redefine all variables, including those listed above.

3.8. Greater Customization with `\TextField`

The `\ScoreField` macro is defined as

```
\newcommand{\ScoreField}[2][width=1.5in]{%  
  \TextField[name=#2,default=\eqScore,align={0 /Ff 1},#1]{}}
```

You can see the command takes one required argument and one optional argument (the required one is label #2 and the optional one is #1). The required argument becomes the **name** of the field, and the optional parameter is just added into the parameter list of `\TextField`. You can also see the **value** of the field is set to `\eqScore`, which expands to ‘Score:’ in English.

More sophisticated score fields can be obtained by manipulating `\TextField` directly.

We can modify the appearance of the text field in a couple of ways: (1) by changing some of the options in `\TextField`; (2) changing the macro definition of `\LayoutTextField`.

- **Admissible key-value pairs of `\TextField`**

In this section we will outline methods of modifying the appearance of `\TextField` text box, which is a macro defined in the `hyperref` package.

The following table lists several useful options for `\TextField`.

▶ \TextField Options: Dimension Related

height	dimen		height of box
width	dimen		width of field box

▶ \TextField Options: Text Related

align	name	0	0 = left-aligned, 1 = centered 2 = right-aligned
charsize	dimen	10pt	Font size of text

▶ \TextField Options: Color Related

backgroundcolor	RGB	1 1 1	Color of background color.
bordercolor	RGB	1 0 0	Color of the border.
color	RGB	0 0 0	Color of text within field

<code>borderstyle</code>	name	S	S (solid), B (beveled), I (insert), U (underlined)
--------------------------	------	---	--

▶ `\TextField` Options: Style Related

<code>borderwidth</code>	number	1	Width of the border of the box (in points)
--------------------------	--------	---	--

▶ `\TextField` Options: Field Related

<code>default</code>			default value of the field
<code>name</code>	name		name of the field
<code>readonly</code>	boolean	false	set read-only switch

A complete list of options comes with the `hyperref` documentation; or see Michel Goossens *et al* [4], entitled *The L^AT_EX Web Companion*

We next present several examples that illustrate the use of these options.

- **Some Examples of `\TextField`**



```
\TextField[width=2in,name=QuizTotal,
  bordercolor=0 0 1,default=\eqScore]{}
```

The `readonly` is an important attribute to have—the students cannot modify their score. Unfortunately, `readonly` *does not work* with `hyperref`. To overcome this “bug” in `hyperref`, you have to be a little sneaky.



```
\TextField[width=2in,name=Example,bordercolor=0 0 1,
  backgroundcolor=0.98 0.92 0.73,color=1 0 0,
  align={0 /Ff 1}, % readonly is /Ff 1 in pdf code
  default=Text Field with Various Options]{}
```

The `read only` key-value (`/Ff 1`) can “ride” into the code on the “back” of the `align` parameter. (`align=0` means left-aligned in the text field.)



```
\TextField[width=1.25in,name=Example,bordercolor=1 1 1,
  backgroundcolor=.753 .753 .753,color=0 0 1,
  borderstyle=B,align={1 /Ff 1}, % /Ff 1 = readonly
  default=Your Score goes here]{}
```

- **Modifying \LayoutTextField**

The macro `\LayoutTextField` allows you to design how your text box is formatted by $\text{T}_{\text{E}}\text{X}$. The default definition for `hyperref` is

```
\newcommand\LayoutTextField[2]{#1 #2}
```

The first argument is the label, the second is the textfield. For example, above definition yields:

Enter your name:

```
\TextField[width=2.5in,name=Name]{Enter your name: }
```

The `exerquiz` package redefines this to

```
\renewcommand\LayoutTextField[2]{% label, field
  \medskip\noindent#2}
```

Finally, the field examples given earlier had the following definition for `\LayoutTextField`:

```
\newcommand\redpoint{\par\medskip\noindent
  \makebox[\parindent][1]{\large\textcolor{red}
    {$\blacktriangleright$}}}
\renewcommand\LayoutTextField[2]{\redpoint
  \raisebox{-3pt}{#2}}
```

I lowered the text field to line up better with the `amssymb` symbol `\blacktriangleright` (▶), which is painted red. In this way we obtain



▶ See the file `qz02.tex` for examples of the usage of `\TextField` and `\LayoutTextField`.

4. List of Options

Options of the Web/Exerquiz Packages	
Options of the Web Package	
<code>dvipsone</code>	dvi-to-ps driver by Y&Y, Inc.
<code>dvips</code>	dvi-to-ps driver
<code>pdftex</code>	tex-to-pdf application
<code>dviwindo</code>	Y&Y's dvi previewer (links work in previewer)
<code>dvipdfm</code>	dvi-to-pdf application
<code>navibar</code>	inserts a menu bar at the bottom or each page
<code>latexdoc</code>	displays the standard toc
<code>nodirectory</code>	eliminates the directory listing on the title page

Options of the Web/Exerquiz Packages (cont.)	
<code>forpaper</code>	this turns off color, and does not put solutions on separate pages.
<code>tight</code>	redefines list environment parameters so lists don't take up so much space
<code>french</code>	French for web, passed to exerquiz. (Thanks to Jean-Michel Sarlat)
<code>german</code>	German for web, passed to exerquiz. (Thanks to Michael Wiedmann)
<code>norsk</code>	Noregian for web, passed to exerquiz. (Thanks to Hans Fredrik Nordhaug)

Options of the Web/Exerquiz Packages (cont.)	
Options of the Exerquiz Package	
<code>pdftex</code>	tex-to-pdf application
<code>dviwindo</code>	Y&Y's dvi previewer (<code>exercise</code> environment only)
<code>dvipdfm</code>	dvi-to-pdf application (<code>exercise</code> environment only)
<code>nosolutions</code>	removes the solutions to the exercises
<code>nocorrections</code>	removes the ability to correct the quizzes
<code>solutionsafter</code>	solutions to exercises are typeset just after the question
<code>forpaper</code>	same function as in <code>web</code> . Needed when <code>exerquiz</code> is not used with <code>web</code>

Options of the Web/Exerquiz Packages (cont.)	
<code>french</code>	JavaScript messages in French (Thanks to Jean-Michel Sarlat)
<code>german</code>	JavaScript messages in German (Thanks to Michael Wiedmann)
<code>norsk</code>	JavaScript messages in Noregian (Thanks to Hans Fredrik Nordhaug)

Solutions to Exercises

Exercise 1. We evaluate by integration by parts:

$$\begin{aligned}\int x^2 e^{2x} dx &= \frac{1}{2} x^2 e^{2x} - \int x e^{2x} dx && u = x^2, dv = e^{2x} dx \\ &= \frac{1}{2} x^2 e^{2x} - \left[\frac{1}{2} x e^{2x} - \int \frac{1}{2} e^{2x} dx \right] && \text{integration by parts} \\ &= \frac{1}{2} x^2 e^{2x} - \frac{1}{2} x e^{2x} + \frac{1}{2} \int e^{2x} dx && u = x^2, dv = e^{2x} dx \\ &= \frac{1}{2} x^2 e^{2x} - \frac{1}{2} x e^{2x} + \frac{1}{4} e^{2x} && \text{integration by parts} \\ &= \frac{1}{4} (2x^2 - 2x + 1) e^{2x} && \text{simplify!}\end{aligned}$$

Exercise 1

Problem 3. This is the solution.



Exercise 4. It is well known that $2 + 2 = 4$.

Exercise 4

Project Hint: There, you didn't need my help after all.



Solutions to Quizzes

Solution to Quiz: The answer is ‘Yes’. The definition requires that

$$F'(x) = f(x) \quad \text{for all } x,$$

well, let’s check it out.

The definition of f is $f(s) = 4s^3$ and so $f(x) = 4x^3$.

The definition of F is $F(t) = t^4$ and so, by the rules of differentiation, $F'(t) = 4t^3$. Thus, $F'(x) = 4x^3$. Therefore,

$$F'(x) = 4x^3 = f(x) \quad \text{for all } x,$$

as required by the definition.

End Quiz

Solution to Quiz: If you erred on this one, more than likely it was on the appropriate multiplicative constant: 6 not 18. At least that's what I'm betting on.

The instructions of the LCD Algorithm said to *completely factor the denominator*. Here's a list of the factors

$$\underbrace{3, x^{3/2}, y^2}_{\text{first term}}, \underbrace{2, 3, x, y^4}_{\text{second term}}$$

Let's rearrange them

$$2, 3, 3, x, x^{3/2}, y^2, y^4$$

Now drop duplicate factors—that's the 3. Oops! I did mention dropping identical factors, didn't I?

$$2, 3, x, x^{3/2}, y^2, y^4$$

Now, group together all terms which have the same base, then drop, from each of these groups all terms but the one with the highest power. We obtain then,

$$2, 3, x^{3/2}, y^4$$

The LCD is the product of same:

$$\text{LCD} = (2)(3)x^{3/2}y^4 = 6x^{3/2}y^4.$$

Solution Notes: Alternative (a) will work as a common denominator, but it is not the least common denominator. If you use (a), you will be working with larger numbers than is really necessary. **End Quiz**

References

- [1] Leslie Lamport, *L^AT_EX: A Document Preparation System* (2nd ed.), Addison-Wesley Publishing Company, 1994, ISBN 0-201-52983-1.
- [2] Michel Goossens, Frank Mittelbach and Alexander Samarin, *The L^AT_EX Companion*, Addison-Wesley Publishing Company, 1994, ISBN 0-201-54199-8. [43](#)
- [3] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach, *The L^AT_EX Graphics Companion*, Addison-Wesley Publishing Company, 1997, ISBN 0-201-85469-4.
- [4] Michel Goossens, and Rahtz, Sebastian, with Gurari, Eitan, Moore, Ross, and Sutor, Robert, *The L^AT_EX Web Companion*, Addison Wesley Longman, Reading, Massachusetts, USA, 1999. ISBN: 0-201-43311-7. [15](#), [85](#)

- [5] Helmut Kopka and Patrick W. Daly, *A Guide to L^AT_EX2e* (2nd ed.), Addison-Wesley Publishing Company, 1995, ISBN 0-201-43777-X.
- [6] Donald E. Knuth, *The T_EXbook*, Addison-Wesley Publishing Company, 1987, ISBN 0-201-13448-9.
- [7] Thomas Merz, *Web Publishing with Acrobat/PDF*, Springer-Verlag Berlin Heidelberg New York, 1998, ISBN 3-540-63762-1.
- [8] D.P. Story, *Pdfmarks: Links and Forms*, AcroTeX Web Site, 1998, www.math.uakron.edu/~dpstory/acrotex.html 78
- [9] D.P. Story, *Using L^AT_EX to Create Quality PDF Documents for the WWW*, AcroTeX Web Site, 1998, www.math.uakron.edu/~dpstory/acrotex.html 9