

exam-rtl — An Arabic/RTL Exam Package*

Nadeer Aljaroudi

2026-08-20

Abstract

The exam-rtl package is the Arabic/RTL version of exam-ltr. It provides a comprehensive framework for creating professional exams in Arabic and Right-to-Left (RTL) languages. It supports a wide variety of question types, including Multiple Choice, True/False, Fill in the Blank, Ordering, Matching, Essay, Short Answer, and Reading Passages.

Key features include automatic answer key generation, controlled randomization of options and matching items, and full support for RTL text alignment and Eastern Arabic numerals.

Contents

1	Introduction	2
2	Requirements	2
3	Installation	2
4	Quick Start	3
5	Mode Commands	4
6	Question Environment	5
7	Question Types	6
7.1	Multiple Choice (MC)	6
7.2	True/False (TF)	6
7.3	Fill in the Blank (FillBlank)	7
7.4	Ordering	8
7.5	Matching	8
7.6	Essay	9
7.7	Short Answer	10
7.8	Reading Passage	10
8	Complete Example	11
9	Common Mistakes	11

*This file has version v1.0, dated 2026-08-20.

10 Known Limitations	12
11 License	12
12 Source Code	13

1 Introduction

The `exam-rtl` package is the Arabic/RTL version of `exam-ltr`. It provides a clean and robust environment for building professional exams without manual formatting. Built with educators in mind, it allows you to focus on creating high-quality question content while the package handles question layout and RTL alignment automatically.

Key features:

- **8 Dedicated Question Types:** MC, TF, FillBlank, Ordering, Matching, Essay, ShortAnswer, and ReadingPassage.
- **Automated Answer Keys:** Answers are collected and displayed directly after each question in teacher mode.
- **Smart Randomization:** Shuffles options, matching columns, and ordering items in a reproducible manner.
- **Metadata Display:** Show or hide marks, difficulty levels, and question type tags.
- **Dual-Mode Output:** Switch between student version (answers hidden) and teacher version (answers shown).
- **Arabic Support:** RTL text alignment and Eastern Arabic numerals.

2 Requirements

- **LaTeX engine:** XeLaTeX (PDFLaTeX and LuaLaTeX are **not** supported)
- **Fonts:** IBM Plex Sans Arabic (default) – ensure it is installed on your system
- **Required packages** (loaded automatically): `xcolor`, `tcolorbox`, `fontspec`, `polyglossia`, `bidi`, `expl3`, `xparse`

3 Installation

The `exam-rtl` package is distributed through CTAN and is included in all major LaTeX distributions (TeX Live, MiKTeX, MacTeX). In most cases, no manual installation is needed; the system's package manager will handle it automatically.

If manual installation is required, use one of the following methods.

Quick Installation (Single Project)

1. Place `exam-rtl.sty` in the same folder as your `.tex` file.
2. Add `\usepackage{exam-rtl}` to your preamble.

Permanent Installation (All Projects)

Linux / macOS

1. Create the target directory:
`mkdir -p ~/texmf/tex/latex/exam-rtl`
2. Copy the package file into this directory:
`cp exam-rtl.sty ~/texmf/tex/latex/exam-rtl/`
3. Update the LaTeX file database:
`texhash ~/texmf`

Windows

MiKTeX:

1. Create the folder: `C:\texmf\tex\latex\exam-rtl`
2. Copy `exam-rtl.sty` into it.
3. Refresh the database via the MiKTeX Console “Refresh FNDB”.

TeX Live:

1. Create the folder: `C:\texlive\texmf-local\tex\latex\exam-rtl`
2. Copy `exam-rtl.sty` into it.
3. Refresh the database by running `texhash`.

4 Quick Start

This section outlines the standard workflow for creating an exam with the `exam-rtl` package.

Step 1: Load the Package and Configure Mode

Include the package in your document preamble and set the desired output mode:

```
\documentclass{article}
\usepackage{exam-rtl}

\TeacherMode    % Show answer keys

\ShowMarks      % Display marks
\ShowCorrection % Show answer boxes
\ExamID{123}    % Fix randomization seed
```

Step 2: Structure Your Questions

Wrap every question in the `Question` environment. You must specify four arguments: $\langle Type \rangle$, $\langle Difficulty \rangle$, $\langle Marks \rangle$, and $\langle Question\ text \rangle$:

```
\begin{Question}{\langle Type \rangle}{\langle Difficulty \rangle}{\langle Marks \rangle}{\langle Question\ text \rangle}
    question\ content
\end{Question}
```

Step 3: Choose the Environment for Your Question

Inside the `Question` environment, select the environment that matches your question type: `MC` (Multiple Choice), `TF` (True/False), `FillBlank`, `Ordering`, `Matching`, `Essay`, `ShortAnswer`, or `ReadingPassage`.

Step 4: Compile and View Output

Compile your document with `xelatex`. The output depends on the mode set in Step 1:

- **Student Mode (default):** Answer keys are hidden.
- **Teacher Mode:** Answer keys appear only when `\ShowCorrection` is also enabled.

For complete examples in Arabic, please refer to the `exam-rtl-doc.tex` file included with the package.

5 Mode Commands

The package provides several switches that control the appearance of the final exam sheet. These switches allow you to toggle between student and teacher views, display metadata, and show answer boxes.

Student and Teacher Modes

`\StudentMode` These commands toggle between student and teacher views.

```
\TeacherMode
\StudentMode    % Hide answer keys (default)
\TeacherMode    % Show answer keys
```

The default mode is `\StudentMode`. When `\TeacherMode` is active, answer keys become visible.

Metadata Visibility

```
\ShowMarks      These commands control the visibility of question metadata.
\ShowDifficulty
\ShowType
\ShowMarks      % Display marks
\ShowDifficulty % Display difficulty levels
\ShowType       % Display question type
```

Each switch is independent; any combination may be enabled.

Correction Boxes

`\ShowCorrection` Enables the display of correction boxes for teacher use.

`\ShowCorrection`

Only takes effect when `\TeacherMode` is explicitly enabled.

Randomization Seed

`\ExamID` Sets a fixed seed for all randomization operations (shuffling MC options, ordering items, and matching pairs).

`\ExamID{<number>}`

Using the same seed guarantees identical output across compilations.

6 Question Environment

`Question (env.)` The Question environment is the primary container for all question types. It handles automatic numbering, manages metadata display (type, difficulty, marks), and integrates answer key collection in teacher mode.

Syntax

```
\begin{Question}{<Type>}{<Difficulty>}{<Marks>}{<Question Text>}{
  question content
}\end{Question}
```

Arguments

- `<Type>`: Question category (MC, TF, FillBlank, Ordering, Matching, Essay, ShortAnswer, ReadingPassage).
- `<Difficulty>`: Descriptive label (e.g., Easy, Medium, Hard).
- `<Marks>`: Numeric score assigned to the question.
- `<Question Text>`: Main prompt or instruction.

How It Works

- **Automatic Numbering:** Questions are numbered sequentially.
- **Metadata Display:** When enabled via `\ShowType`, `\ShowDifficulty`, and `\ShowMarks`, the corresponding labels are displayed alongside the question.
- **Answer Key Collection:** When `\TeacherMode` and `\ShowCorrection` are enabled, all answers are collected and displayed in a dedicated answer key box.
- **Consistent Formatting:** All question types placed inside this environment share consistent spacing and layout.

7 Question Types

7.1 Multiple Choice (MC)

`MC (env.)` The MC environment creates multiple-choice questions with automatically shuffled options. Multiple questions can be placed inside a single MC environment, each with its own set of options. The correct answer is stored internally and appears in the answer key in teacher mode. Options are displayed in a grid layout with a customizable number of columns.

Syntax:

```
\begin{MC} [columns]  
  \MCQuestion{question text}{correct_index}  
  \MCOption{option 1}  
  \MCOption{option 2}  
  \MCOption{option 3}  
  ...  
\end{MC}
```

How It Works:

- **Column Layout (default: 4):** The optional [*columns*] argument sets the number of columns for displaying options. Any number can be used as long as it fits the page width.
- **Correct Answer Index:** The {*correct_index*} refers to the position of the correct option as written (e.g., 1, 2, 3). This index is used internally to track the correct answer after shuffling.
- **Randomization:** All options inside `\MCQuestion` are shuffled based on the global exam seed set by `\ExamID`. This ensures reproducibility when a fixed seed is provided.
- **Answer Key Integration:** The correct answer is collected automatically and displayed in the answer key when `\TeacherMode` and `\ShowCorrection` are enabled.

7.2 True/False (TF)

`TF (env.)` The TF environment creates True/False questions. Multiple statements can be placed inside a single TF environment, each with its own correct answer. Two display styles are available: bracket style (default) and options style, allowing flexibility for different assessment needs.

Syntax:

```
\begin{TF} [style]  
  \TFQuestion{statement}{answer}  
  \TFQuestion{statement}{answer}  
  ...  
\end{TF}
```

How It Works:

- **Visual Style:** The optional [*style*] argument selects the display format:
 - **bracket** (default): Places an empty answer slot () after each statement.
 - **options**: Places “True” and “False” labels to the right of each statement.
- **Statement Input:** The $\{\langle statement \rangle\}$ argument receives the text that students must evaluate.
- **Answer Specification:** The $\{\langle answer \rangle\}$ argument must contain either “True” or “False”.
- **Answer Key:** When `\TeacherMode` and `\ShowCorrection` are enabled, the correct answers are collected and displayed in the answer key.
- **Multiple Questions:** Multiple `\TFQuestion` commands can be placed sequentially within one TF environment.

7.3 Fill in the Blank (FillBlank)

`FillBlank` (*env.*) The FillBlank environment creates fill-in-the-blank questions where students complete sentences with appropriate words. It supports single or multiple blanks per question and an optional word bank to assist students during assessment.

`\WordBank`
`\FillQuestion`
`\Blank`

Syntax:

```
\begin{FillBlank}[\langle wordbank \rangle]
  \WordBank[\langle separator \rangle]{\langle word1, word2, ... \rangle}
  \FillQuestion{\langle text with \Blank \rangle}{\langle answer \rangle}
  ...
\end{FillBlank}
```

How It Works:

- **Blank Creation:** The `\Blank` command creates an underline field where students write their answers. Its optional argument [*width*] sets the line length (default: 15mm).
- **Word Bank Activation:** The optional [*wordbank*] argument controls whether a word bank is displayed. Set to `true` to show a framed list of words above the questions (default is `false`).
- **Word Bank Configuration:** The `\WordBank` command defines the vocabulary list. Words must be separated by commas. The optional [*separator*] controls spacing between words (default: `\quad`).
- **Multiple Questions:** Multiple `\FillQuestion` commands can be placed sequentially within one FillBlank environment.
- **Answer Key:** When `\TeacherMode` and `\ShowCorrection` are enabled, correct answers are collected and displayed in a dedicated answer key box.

7.4 Ordering

Ordering (*env.*) The Ordering environment creates sequencing questions where students must arrange items in their correct logical order. Items must be written in their proper sequence inside the environment, but are presented to students in a shuffled layout. The original order is preserved for the teacher's answer key. Options are displayed in a grid layout with a customizable number of columns.

Syntax:

```
\begin{Ordering}[\langle columns \rangle]
  \OrderQuestion{\langle instruction \rangle}
  \OrderOption{\langle item 1 \rangle}
  \OrderOption{\langle item 2 \rangle}
  \OrderOption{\langle item 3 \rangle}
  ...
  \OrderAnswer[\langle width \rangle]
\end{Ordering}
```

How It Works:

- **Correct Sequence:** Items must be written inside `\OrderOption` in their correct logical order. This order is preserved for the answer key.
- **Randomization:** Items are shuffled for the student based on the global exam seed (`\ExamID`), ensuring consistent randomization across compilations.
- **Column Layout (default: 4):** The optional `[\langle columns \rangle]` argument sets the number of columns for displaying shuffled items. Any number can be used as long as it fits the page width.
- **Answer Space:** The `\OrderAnswer` command adds an optional answer space with customizable width (default: 5mm) where students write their final ordered sequence.
- **Answer Key:** When `\TeacherMode` and `\ShowCorrection` are enabled, the correct sequence is collected and displayed in the answer key.

7.5 Matching

Matching (*env.*) The Matching environment creates two-column association questions. Items must be written in their correct pairs inside the environment. The right column remains fixed and numbered, while the left column is automatically shuffled for students. The original pairings are preserved for the teacher's answer key.

Syntax:

```
\begin{Matching}[\langle right.ratio \rangle]
  \MatchPair{\langle left item \rangle}{\langle right item \rangle}
  \MatchPair{\langle left item \rangle}{\langle right item \rangle}
  ...
\end{Matching}
```


How It Works:

- **Pair Definition:** Each pair is defined using `\MatchPair`. The order of definition determines the correct answer key. The right column is fixed and numbered, while the left column is automatically shuffled.
- **Column Width Control:** The optional `[\right_ratio]` adjusts the width of the right column as a decimal fraction between 0 and 1. Default is 0.5 (equal split). This is useful when the right column contains short terms and the left column has lengthy descriptions (e.g., 0.3 for 30% width).
- **Randomization:** The left column is shuffled based on the global exam seed (`\ExamID`), ensuring consistency across compilations.
- **Answer Key:** When `\TeacherMode` and `\ShowCorrection` are enabled, the correct mappings are displayed as numbered-to-lettered pairs.
- **Capacity:** Supports a maximum of 28 pairs per question (limited by Arabic alphabet).

7.6 Essay

Essay (*env.*) The Essay environment creates open-ended, long-form written response questions.
\EssayQuestion It provides a writing area with either ruled lines or blank space, depending on the selected style. The instructor's answer summary is stored and displayed in the answer key in teacher mode.

Syntax:

```
\begin{Essay}[\langle lines \rangle]
  \EssayQuestion[\langle line_count \rangle]{\langle question \rangle}{\langle answer summary \rangle}
  ...
\end{Essay}
```

How It Works:

- **Writing Space Style:** The optional environment argument `[\langle lines \rangle]` controls the appearance of the writing area:
 - **true:** Generates ruled notebook lines (default).
 - **false:** Provides blank white space for free-form text or diagrams.
- **Writing Space Size (default: 4):** The optional `[\langle line_count \rangle]` argument specifies the number of lines allocated for the student's answer.
- **Question Components:** `\EssayQuestion` command takes two mandatory arguments: the question prompt and a concise answer summary for the instructor's reference.
- **Multiple Questions:** Multiple `\EssayQuestion` commands can be placed sequentially within one Essay environment.
- **Answer Key:** When `\TeacherMode` and `\ShowCorrection` are enabled, all answer summaries are collected and displayed in a dedicated answer key box.

7.7 Short Answer

ShortAnswer (*env.*) The ShortAnswer environment creates questions that require brief, direct responses — such as single words, short phrases, or concise statements. The answer space can be customized (0, 1, or 2 lines) to keep the exam layout clear and space-efficient. The instructor’s answer is stored and displayed in the answer key in teacher mode.

Syntax:

```
\begin{ShortAnswer}[\langle lines \rangle]
  \ShortQuestion[\langle size \rangle]{\langle question \rangle}{\langle answer \rangle}
  ...
\end{ShortAnswer}
```

How It Works:

- **Writing Space Style:** The optional environment argument `[\langle lines \rangle]` controls the appearance of the answer space:
 - **true:** Generates ruled notebook lines (default).
 - **false:** Provides blank white space without lines.
- **Answer Space Size:** The optional `[\langle size \rangle]` argument specifies the number of lines for the student’s answer:
 - **0:** Prints a short underline directly after the question text — ideal for single-word answers. (Prints a line regardless of the environment style.)
 - **1:** Generates a one-line writing space below the question.
 - **2:** Generates a two-line writing space below the question.
- **Question Components:** `\ShortQuestion` command takes two mandatory arguments: the question prompt and the model answer or keyword solution.
- **Multiple Questions:** Multiple `\ShortQuestion` commands can be placed sequentially within one ShortAnswer environment.
- **Answer Key:** When `\TeacherMode` and `\ShowCorrection` are enabled, all answers are collected and displayed in a dedicated answer key box.

7.8 Reading Passage

ReadingPassage (*env.*) The ReadingPassage environment provides a container for reading comprehension texts. The passage is printed exactly as written — preserving all formatting, paragraph breaks, and inline styling — without any modification or randomization. The passage is displayed in a standalone box, and authors can place any question type after it to assess comprehension.

Syntax:

```
\begin{ReadingPassage}{\langle title \rangle}
  passage text
\end{ReadingPassage}
```

How It Works:

- **Passage Rendering:** The text is displayed in a framed box, exactly as entered, preserving all formatting.
- **Title:** The mandatory `{\title{}}` argument sets the centered heading above the passage. If left empty `{}`, the default title “Reading Passage” is used.
- **Content Flexibility:** The passage can include multiple paragraphs, inline text styles (bold, italic, etc.), and custom lists.
- **Question Integration:** Any question type can follow a `ReadingPassage` within the same `Question` block to assess comprehension.

8 Complete Example

A fully functional sample exam file, `example.tex`, is included with the package to demonstrate its capabilities in a realistic academic context. The example file covers all question types supported by the package: `MC`, `TF`, `Ordering`, `FillBlank`, `Matching`, `ShortAnswer`, `Essay`, and `ReadingPassage`.

Usage Instructions:

1. Ensure you have the file `example.tex` in your working directory (usually found in the package documentation folder).
2. Open it in your $\text{\LaTeX}$ editor and compile with `xelatex`.
3. To switch between student and teacher views, uncomment or comment the `\TeacherMode` and `\ShowCorrection` commands in the preamble. This toggles the display of answer keys and correction boxes.

Note: Arabic examples are provided in the `exam-rtl-doc.tex` file included with the package.

9 Common Mistakes

The following list addresses frequent errors encountered during exam creation. Adhering to these guidelines ensures correct compilation and output.

- **Enabling `\ShowCorrection` without `\TeacherMode`:** `\ShowCorrection` alone does not display answers. Both `\TeacherMode` and `\ShowCorrection` must be active for solutions to appear.
- **Omitting the `Question` Environment:** All question types must be wrapped inside the `Question` environment. Failing to do so prevents metadata and answer keys from appearing.
- **Using Invalid Type Identifiers:** The first argument of `Question` must be a valid identifier: `MC`, `TF`, `FillBlank`, `Ordering`, `Matching`, `Essay`, `ShortAnswer`, or `ReadingPassage`. Using non-standard identifiers (e.g., `MCQ`) will result in missing or incorrect metadata labels.

- **Using the Wrong Engine:** This package requires XeLaTeX. PDFLaTeX and LuaLaTeX are not supported due to the `bidi` package dependency.
- **Incorrect Column Placement in Matching:** The `Matching` environment always shuffles the left column. Items intended for randomization must be placed in the first argument.
- **Incorrect Input Sequence in Ordering:** The order in which you write the `\OrderOption` commands in your document defines the correct answer. Entering them in the wrong order will result in incorrect grading.
- **Omitting the Title in ReadingPassage:** The `ReadingPassage` environment requires a title argument. If no custom title is needed, use empty braces `{}` . Omitting the argument causes LaTeX to misinterpret the first character of the passage text as the title.
- **Font not installed:** The default font is IBM Plex Sans Arabic. Ensure it is installed on your system.

10 Known Limitations

The `exam-rtl` package has the following limitations in its current version.

- **Directionality:** The package is optimized for Right-to-Left (RTL) languages (e.g., Arabic). For LTR support, use the companion package `exam-ltr` .
- **Mandatory Randomization:** Option shuffling cannot be fully disabled. Even without the `\ExamID` macro, a default randomization is applied.
- **Scope of Randomization:** Randomization is limited to internal elements: choices in `MC` , items in `Ordering` , and the left-hand column in `Matching` . Shuffling the entire sequence of question blocks is not supported in this version.
- **Item Capacity:** In `Ordering` , each question is limited to a maximum of 28 options (items to order). In `Matching` , each question is limited to a maximum of 28 pairs. Both limits are due to the 28 Arabic letters used for labeling.
- **Short Answer Lines:** Configuring more than 2 lines in the `ShortAnswer` environment caps the display at 2.
- **Answer Key Placement:** Answer keys are displayed individually beneath each question. A centralized answer sheet at the end of the document is not supported in this version.

11 License

This package is distributed under the LaTeX Project Public License (LPPL), version 1.3c or later.

The latest version of this license is available at: <http://www.latex-project.org/lppl.txt>

12 Source Code

The complete source code of the `exam-rtl` package is included in this `.dtx` file. To view the code:

- Open `exam-rtl.dtx` in any text editor.
- The code is located after the line: `% \begin{macrocode}`.
- To extract the package only: `latex exam-rtl.ins`.
- The extracted `exam-rtl.sty` will preserve all Arabic text correctly.
- This documentation does not include the source code to avoid compilation issues with Arabic text.

Change History

v1.0	release	1
General: Initial Arabic/RTL		