

Exam Papers

Posing Questions To Students

Abstract

Exam is a module for consistent production and maintenance of student examinations. Provided for are various types of questions such as with long and small answers, yes/no questions and multiple choice.

Keywords

examination, multiple choice, ConTeXt

Introduction

The ConTeXt *hvd-m-exm* module is a means for easy and consistent typesetting of exams and collections of exam questions. It especially facilitates questions with short answers and multiple choice. Many aspects of the typesetting are configurable.

This module is an adapted and upgraded version of my previous LaTeX package called the *exam* package. Its last version 3.30 of 1997/03/14 should still be available on CTAN (macros/latex/contrib/exams) but is no longer maintained by me. Changing to ConTeXt proved an efficient route to enhanced behaviour.

Exam structure

Exams can be typeset in two different formats. First of course in the format as presented to students undergoing the examination; this version can be typeset with nothing more than the bare examination questions. The other extreme is a collection of all questions with answers, annotations and points awarded. In between these is a lot of flexibility.

Customization may be done by setting various parameters, either specific to the macros in this module or through those inherent to the ConTeXt macros on top of which they are built. As a last resort one can redefine macros at will; there is a fair amount of modularisation to facilitate this.

A small builtin vocabulary of common language dependent terms is used. Defining them for another language allows one to typeset these terms automatically if that is ConTeXt's current active language. Looking at the code it will be apparent what is needed for additional languages.

All typesetting of exams, either for an examination or for the whole collection, has the following simple

overall structure:

```
\startexam[options]
  \setexamdirectory[first series]
  \question[options]{file-1}
  \question[options]{file-2}
  ...
  \setexamdirectory[second series]
  \question[options]{file-3}
  \question[options][buffer]{\getbuffer}
  ...
\stopexam
```

Each question resides in its own file, but these may be put in different directories. With `\setexamdirectory` the reading directory is switched. One can intersperse the typesetting with all sort of text material. As an alternative to file reading, the questions can also come from ConTeXt buffers; this makes their inclusion in expository texts easy. The last question in the example above uses text defined between ConTeXt's simple `\startbuffer ... \stopbuffer` construct. If its second parameter is `buffer` the argument within the braces is interpreted as the callup of a buffer instead of a filename. The questions themselves are defined with:

```
\startquestion[options]
  ... text question followed by answer ...
\stopquestion
```

At the end of the exam a titlepage is made. Typesetting this after the questions makes for easy production of a summary with data such as the number of questions and the total number of points to be earned. For old hands: typesetting the titlepage after the contents was already seen in the early days of TeX processing, when program text was done with the *webmac* macro package.

Filenames may omit the `.tex` suffix, as is usual in the TeX world. Without an explicit `\setexamdirectory` the file is taken from the current directory.

More often than not — at least in my experience — one mistypes the name of some file in a long series of them. Instead of aborting the production run, or worse

forgoing the mistake silently, there is a prominent message drawing attention to the mistake. Below is the warning that `missing-file.tex` wasn't located:

FILE missing-file UNKNOWN

Examples

Following are some examples of the capabilities of the module.

Short answer. First a simple question where a short answer has to be given. The definition is here:

```
\startquestion[%
  date=23-07-2006,
  score=2,
  subject={Short answer}]
What is the most famous question?
\shortanswer To be or not to be.\par
\stopquestion
```

The arguments of the `\startquestion` serve the following purpose. The `date` field is for the creation date of the question. The `score` is the maximum attainable number of points that can be earned by giving a correct answer (though this value may be overridden at production of the exam, if so required). The `subject` provides a short description for the question. The body of the question contains its text, followed by the predetermined answer. Being a fairly short answer the `\shortanswer` form is chosen; note the `\par` that must finish it. If called with option `showanswer=no` (a natural choice for the production of a student copy) the typeset question looks like this:

Problem 1: What is the most famous question?
Answer:

The answer has to be filled in on the dotted line. Setting the option to `showanswer=yes` in the option field of `\startexam` lets the dotted line disappear and the answer appears instead.

Problem 1: What is the most famous question?
Answer: To be or not to be.

There is an option (not demonstrated here) for showing the scorebox: in the margin a square is placed, containing the maximum score set for the question. The size of the square is easily changed by redefinition:

```
\def\scoreboxsize{dimension}.
```

Below one can observe the extra information at the top of the question for the `series=yes` option.

— Short answer *** file = buffer:23-07-2006 —
Problem 1: What is the most famous question?
Answer: To be or not to be.

In the next examples the output is given twice, with respectively `showanswer=no` and `yes`. Furthermore the parameters `date`, `score` and `subject` of the `\startquestion` are omitted from the accompanying code as shown.

Standard answer. Opposed to the short answer stands the standard answer. The body of the answer is enclosed in the pair `\startanswer ... \stopanswer`. Enough vertical room must be reserved to allow the student to give a complete answer. That space is specified on the definition of the question in the first option argument; it can however be overridden when necessary. Omitting it takes the last default that has been set with option `answerspace`. In case answers are produced, the vertical space parameter is ignored and the answer will be typeset in its natural height.

```
\startquestion
\annotation Short question, long answer.\par
Elaborate on the question
\quotation{To be or not to be}.
\startanswer[2cm]
... sample answer text ...
\stopanswer
\annotation A second note.\par
\stopquestion
```

Problem 2: Elaborate on the question “To be or not to be”.
Answer:

In the next printout, containing the answer, the `notes` option has been set to show the `\annotation` macro. Annotations are useful as a form of comment, being neither part of the question nor of the answer. Annotations are collected and typeset together at the end of the question-answer body. If more than one annotation is present, they are automatically itemized (see also the note in example 6).

Problem 2: Elaborate on the question “To be or not to be”.

Answer: This is a famous quote from *Hamlet*, a play of the English writer William Shakespeare. One really has to suspect that nowadays not many students have seen even one of Shakespeare’s plays, let alone having read one. Most might not even know when or where William Shakespeare is supposed to have been born.

— Notes —

1. Short question, long answer.
2. A second note.

There is a third possibility in the way a long answer is treated; it is illustrated in the code below. The option parameter of `\startanswer` is given the value `force`. As a result the answer block is typeset, regardless whether answer showing is on or off. This is useful in those cases where a template must be filled in by the student. The `\ifanswers.. \fi` macro allows one to show or hide the answer parts.

```
\startquestion
Finish the following ..
\startanswer[force]
\startitemize[2]
\item this .. \ifanswers Hamlet\fi
\item of .. \ifanswers William Shakespeare\fi
\item .. at \ifanswers Stratford-on-Avon\fi
\stopitemize
\stopanswer
\stopquestion
```

Problem 3: Finish the following three statements on the quotation “To be or not to be”.

Answer:

- this is a famous quote from
- of the English writer
- supposed to be born at

Problem 3: Finish the following three statements on the quotation “To be or not to be”.

Answer:

- this is a famous quote from Hamlet
- of the English writer William Shakespeare
- supposed to be born at Stratford-on-Avon

Alternating answer. For questions that can be answered simply by crossing out or underlining one of two alternatives (such as yes/no, right/wrong) there is the `\altanswer` construct. Its two arguments carry

the alternatives. In answer mode the correct one is underlined; its position being the first one by default, to be changed with the option. The coding is:

```
\startquestion
Was something rotten in the state of Denmark?
\altanswer[left]{\Yes}{\No}
\stopquestion
```

Problem 4: Was something rotten in the state of Denmark?

Answer: yes or no

Problem 4: Was something rotten in the state of Denmark?

Answer: yes or no

Boxed answer. Sometimes an answer is just one thing to put inside a box. The `\answerbox` macro is made for this type of question. The length of the box can be specified in the optional argument; omitting it defaults to the natural width of the answer text. As with the long answer, here too one can change the default width examwide; the relevant option is `boxwidth`. The example:

```
\startquestion
Who posed that question?
\answerbox[3cm]{\quotation{Hamlet}}
\stopquestion
```

Problem 5: Who posed that question?

Answer:

Problem 5: Who posed that question?

Answer:

Parametrized question. When calling up a question it is possible to customize it by means of a parameter. To choose for example between several alternative formulations of essentially the same question. This helps to vary successive exams, while still keeping them similar. The parameter is given its default value on the `\startquestion` with the `parameter=value` option and can be redefined on the `\question` call. An explanation of parameter usage might come in handy and may find its place inside an annotation. In the example below `\question[parameter=2]` is used to select the second alternative whereas 1 is the default.

```

\startquestion[parameter=1]
Was something rotten in the state of
\ifx\parameter\parameterdefault
  Denmark? \altanswer[left]{\Yes}{\No}
\else Britain? \altanswer[right]{\Yes}{\No}
\fi
\startannotation ... \stopannotation
\stopquestion

```

Problem 6: Was something rotten in the state of Britain?

Answer: yes or no

— Notes —

The actual value 2 of `\parameter` is used; the default `\parameterdefault` has value 1.

Multiple choice question. Finally an example of a multiple choice question. In multiple choice there is no inviting “Answer” prompt, because the intention is obvious. When the necessary modules for random generation (*hvdm-rng*) and list sorting (*hvdm-1st*) are installed, one may automatically shuffle the items. The itemlist appearance is determined by the `setupmc` option; there one can give a `\setupitemize` to be executed just before the typesetting of the itemlist.

```

\startquestion
Which person is found in \quote{Hamlet}?
\startmultiplechoice
\wrong Rosalind\par
\wrong Desdemona\par
\right Ophelia\par
\wrong Juliet\par
\stopmultiplechoice
\stopquestion

```

Problem 7: Which person is found in ‘Hamlet’?

- Rosalind
- Desdemona
- Ophelia
- Juliet

Macros and their parameters

A summary of the most important macros and their arguments follows.

`\setupexams[...]=...]`. Does the setup for the subsequent `\exam`’s. The options are summarized in the table. Because questions are put inside a `framedtext`, their appearance can be influenced

with `\setupframedtexts[]`. Some `framedtext` parameters can be set from the option list of `\exam` and `\question`. These options are `rulethickness`, `offset`, `frameoffset`, `radius`, `corner`, `framecolor`, `background`, `backgroundoffset`, `backgroundscreen`, `backgroundcolor`, `backgroundcorner`, `backgroundradius`.

`\startexam[...]=...]`...`\stopexam`. Specify an exam between this start and stop pair. The same options as with `\setupexams` apply here. Special values to mention are:

- ▷ `language` to choose the language in which to typeset the fixed terms;
- ▷ `marks` to choose the marks for multiple choice, customize these yourself by redefinition of `\rightmark`, `\wrongmark` and `\choosemark`;
- ▷ `numbering` switches numbering on/off;
- ▷ `prompt` set this option to no in order to suppress typesetting the “Answer” prompts for exams to be answered on separate answer sheets;
- ▷ `random` for random processing and checking with `\ifrandom.. \else.. \fi` when module *hvdm-rng* is installed, for randomization of multiple choice install *hvdm-1st* too;
- ▷ `separator` separation of questions with hairline, nothing or placement on separate page;
- ▷ `setupmc` setup for multiplechoice itemlist;
- ▷ `showanswer` for typesetting the answers;
- ▷ `showscore` for showing the score values;
- ▷ `standalone` typesets each question apart as is done in this text;
- ▷ `series` produces a catalogue of questions.

`\setexamdirectory[path]`. Set current directory to the (possibly empty) value of the argument.

`\setnumberstart[number]`. Sets the initial value for the numbering of questions. The numbering starts at 1 by default, but one can imagine numbering schemes where different topics each start at a round number. The option value `reset` restarts the numbering at the beginning.

`\question[...]=...]{filename}`. Callup question from a file. The available options are those marked Q in the table. It is possible to include frame parameters in the option list for individual changes. The `number=#` option allows one to have questions individually numbered. Note however that the regular number is still advanced by 1.

`\question[...]=...][buffer]{\getbuffer}`. Get the question from buffer instead of from file.

after	QE	<i>command</i>	execute after question
answerspace	DQE	<i>dimension force</i>	set default answer space
before	QE	<i>command</i>	execute before question
boxwidth	QE	<i>dimension</i>	set default width answerbox
date	DE	<i>text</i>	examination (E) or creation (D) date
frame	QE	<i>on off</i>	put frame around question
hang	QE	<i>number</i>	hang parameter on question and answer
introstyle	QE	<i>stylecommand normal</i>	fontstyle of question, answer intro
language	E	<i>2-letter currentlanguage</i>	languagecode for fixed vocabulary
marks	E	<i>square circle squarev circlev</i>	multiplechoice marks
notes	QE	<i>yes no</i>	place annotations
numbering	E	<i>yes no</i>	show question numbering
parameter	QD	<i>any</i>	actual (Q) or default (D) parameter value
prompt	QE	<i>yes no</i>	place answer part or always suppress
random	QE	<i>yes no</i>	random processing turned on or off
score	QD	<i>number</i>	the question score
separator	E	<i>yes no page</i>	separator or pagebreak after question
series	E	<i>yes no</i>	typeset catalogue of questions
setupmc	QE	<i>\setupitemize</i>	format multiplechoice item list
showanswer	E	<i>yes no</i>	show answers
showscore	E	<i>yes no</i>	show score and possibly subscores
standalone	E	<i>yes no</i>	typeset isolated question
style	QE	<i>stylecommand normal</i>	fontstyle of question body
stopper	QE	<i>char</i>	placed behind question and answer
subject	E	<i>text</i>	field of the exam
width	QE	<i>fit broad dimension</i>	width of question

Setups effective on question definition (D), question (Q) exam (E)

`\startquestion[...]\stopquestion.` Put the definition of a question fully inside this macro pair. The most obvious pattern is to start with the text of the question and let this follow by the answer. Typesetting begins with the “Question” header and an optional sequence number, any answer macro following then triggers the “Answer” header. The options applicable are summarized below.

date	<i>text</i>	<u>empty</u>	creation date
subject	<i>text</i>	<u>empty</u>	descriptive text
score	<i>number</i>	<u>0</u>	points allotted
parameter	<i>any</i>	<u>empty</u>	parameter default

When a question has more than one part, it is handy to allot a fixed number of points to each subquestion. The macro `\subscore[#]` places them in the margin when `showanswer=yes`. It helps maintaining consistency by reporting an error if the sum of the subscores does not equal the score given to the question as a whole. The number option makes it possible to number each question individually or reset the numbering from a given question on.

`\shortanswer... \par.` Answer text of one para-

graph to be placed on a dotted line. The closing `\par` that may not be omitted.

`\startanswer[length]... \stopanswer.` Answer text is put inside this pair. The optional length parameter specifies the vertical amount of space to reserve for the answer. If it is not given the default value is used. Set the length parameter to *force* in order to force typesetting of the answer block in its natural height unless suppressed completely with the `prompt=no` setting. The value given here always prevails, because it is clearly something the designer of the question had in mind. It is therefore advised to use this parameter sparingly and preferably tune these lengths on `\startexam` (global) and/or on the `\question` (individual).

`\answerbox[width]{...}`. Answer to be placed inside a box of given width. If width is not given the default value is used. The default width can be specified by the `boxwidth` parameter.

`\altanswer[left,right]{1st}{2nd}`. Two alternating answers. Defaults to the first alternative as the correct one, any option value other than `left` or `empty` switches this to the second argument. Redefine macro

`\altanswerseparator` for the separator between the two possibilities.

`\startmultiplechoice... \stopmultiplechoice`
`\right... \par`, `\wrong... \par`. A multiple choice item list should be put inside the start/stop pair. Place the correct item after `\right` and the wrong items after `\wrong`, closing them with `\par`; if you like a synonym like `\ok` better, just define `\ok` as `\right`. The symbols marking the items can be chosen with the `marks` option on `\startexam`.

`\annotation ... \par`
`\startannotation... \stopannotation`. If put inside the question, the annotations will be added at the end of it according to the value of the `notes` option. For example useful to explain `\parameter` usage. The

first format limits the contents to one paragraph; do not forget the closing `\par` then.

`\examtitlepage`. Produces a standard titlepage. Redefine this macro at will. Macros `\thetitle`, `\thedate`, `\thetotalquestions`, `\thetotalscore` are available for use; the latter two are filled in by `\stopexam`.

`\endefines`. English language vocabulary. Others may be added by defining a macro of the form `\xxdefines` similar to `\endefines`. Showing the vocabulary for a language can be done with `\meaning\xxdefines`.

Hans van der Meer
 hansm@science.uva.nl