

hwexam.sty/cls: An Infrastructure for formatting Assignments and Exams*

Michael Kohlhase
Jacobs University, Bremen
<http://kwarc.info/kohlhase>

October 13, 2010

Abstract

The `hwexam` package and class allows individual course assignment sheets and compound assignment documents using problem files marked up with the `problem` package.

Contents

1	Introduction	2
2	The User Interface	2
2.1	Package and Class Options	2
2.2	Assignments	2
2.3	Typesetting Exams	2
2.4	Including Assignments	3
3	Limitations	4
4	Implementation: The hwexam Class	5
4.1	Class Options	5
5	Implementation: The hwexam Package	6
5.1	Package Options	6
5.2	Assignments	7
5.3	Including Assignments	9
5.4	Typesetting Exams	9
5.5	Leftovers	11

*Version v0.9b (last revised 2010/06/25)

1 Introduction

The `hwexam` package and class supplies an infrastructure that allows to format nice-looking assignment sheets by simply including problems from problem files marked up with the `problem` package [Koh10c]. It is designed to be compatible with `problems.sty`, and inherits some of the functionality.

2 The User Interface

2.1 Package and Class Options

The `hwexam` package and class take the options `solutions`, `notes`, `hints`, `pts`, `min`, and `boxed` that are just passed on to the `problems` package (cf. its documentation for a description of the intended behavior).

`showmeta` If the `showmeta` option is set, then the metadata keys are shown (see [Koh10a] for details and customization options).

The `hwexam` class additionally accepts the options `report`, `book`, `chapter`, `part`, and `showignores`, of the `omdoc` package [Koh10b] on which it is based and passes them on to that. For the `extrefs` option see [Koh10d].

2.2 Assignments

`assignment` This package supplies the `assignment` environment that groups problems into
`number` assignment sheets. It takes an optional `KeyVal` argument with the keys `number`
(for the assignment number; if none is given, 1 is assumed as the default or —
in multi-assignment documents — the ordinal of the `assignment` environment),
`title` `title` (for the assignment title; this is referenced in the title of the assignment
`type` sheet), `type` (for the assignment type; e.g. “quiz”, or “homework”), `given` (for
`given` the date the assignment was given), and `due` (for the date the assignment is due).
`due`

2.3 Typesetting Exams

`multiple` Furthermore, the `hwexam` package takes the option `multiple` that allows to combine
multiple assignment sheets into a compound document (the assignment sheets
are treated as section, there is a table of contents, etc.).

`test` Finally, there is the option `test` that modifies the behavior to facilitate formatting
tests. Only in `test` mode, the macros `\testspace`, `\testnewpage`, and
`\testemptypage` have an effect: they generate space for the students to solve the
given problems. Thus they can be left in the \LaTeX source.

`\testspace` `\testspace` takes an argument that expands to a dimension, and leaves vertical
`\testnewpage` space accordingly. `\testnewpage` makes a new page in `test` mode, and
`\testemptypage` `\testemptypage` generates an empty page with the cautionary message that this
page was intentionally left empty.

`testheading` Finally, the `\testheading` takes an optional keyword argument where the keys
`duration` `duration` specifies a string that specifies the duration of the test, `min` specifies the
`min` equivalent in number of minutes, and `reqpts` the points that are required for a
`reqpts`

perfect grade.

```
\title{320101 General Computer Science (Fall 2010)}
\begin{testheading}[duration=one hour,min=60,reqpts=27]
  Good luck to all students!
\end{testheading}
```

formats to

Name: _____ Matriculation Number: _____

320101 General Computer Science (Fall 2010)
October 13, 2010

You have one hour(sharp) for the test;
Write the solutions to the sheet.
The estimated time for solving this exam is 58 minutes, leaving you 2 minutes for revising your exam.
You can reach 30 points if you solve all problems. You will only need 27 points for a perfect score, i.e. 3 points are bonus points.

Different problems test different skills and knowledge, so do not get stuck on one problem.

	To be used for grading, do not write here								
prob.	1.1	2.1	2.2	2.3	3.1	3.2	3.3	Sum	grade
total	4	4	6	6	4	4	2	30	
reached									

good luck

Example 1: A generated test heading.

2.4 Including Assignments

`\includeassignment` The `\includeassignment` macro can be used to include an assignment from another file. It takes an optional KeyVal argument and a second argument which is a path to the file containing the problem (the macro assumes that there is only one `assignment` environment in the included file). The keys `number`, `title`, `type`, `given`, and `due` are just as for the `assignment` environment and (if given) overwrite the ones specified in the `assignment` environment in the included file.

3 Limitations

In this section we document known limitations. If you want to help alleviate them, please feel free to contact the package author. Some of them are currently discussed in the `STEX` TRAC [Ste].

1. none reported yet

4 Implementation: The hwexam Class

The functionality is spread over the `hwexam` class and package. The class provides the `document` environment and pre-loads some convenience packages, whereas the package provides the concrete functionality.

`hwexam.dtx` generates four files: `hwexam.cls` (all the code between `<*cls`) and `</cls`), `hwexam.sty` (between `*package`) and `</package`) and their L^AT_EXML bindings (between `*ltxml.cls`) and `</ltxml.cls`) and `<*ltxml.sty`) and `</ltxml.sty`) respectively). We keep the corresponding code fragments together, since the documentation applies to both of them and to prevent them from getting out of sync.

4.1 Class Options

To initialize the `hwexam` class, we declare and process the necessary options by passing them to the respective packages and classes they come from.

```
1 <*cls>
2 \DeclareOption{test}{\PassOptionsToPackage{\CurrentOption}{hwexam}}
3 \DeclareOption{multiple}{\PassOptionsToPackage{\CurrentOption}{hwexam}}
4 \DeclareOption{showmeta}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
5 \DeclareOption{extrefs}{\PassOptionsToPackage{\CurrentOption}{sref}}
6 \DeclareOption{notes}{\PassOptionsToPackage{\CurrentOption}{problem}}
7 \DeclareOption{hints}{\PassOptionsToPackage{\CurrentOption}{problem}}
8 \DeclareOption{solutions}{\PassOptionsToPackage{\CurrentOption}{problem}}
9 \DeclareOption{pts}{\PassOptionsToPackage{\CurrentOption}{problem}}
10 \DeclareOption{min}{\PassOptionsToPackage{\CurrentOption}{problem}}
11 \DeclareOption{boxed}{\PassOptionsToPackage{\CurrentOption}{problem}}
12 \DeclareOption{extract}{\PassOptionsToPackage{\CurrentOption}{problem}}
13 \DeclareOption*{\PassOptionsToClass{\CurrentOption}{omdoc}}
14 \ProcessOptions
15 </cls>
16 <*ltxml.cls>
17 # -*- CPERL -*-
18 package LaTeXML::Package::Pool;
19 use strict;
20 use LaTeXML::Package;
21 use LaTeXML::Util::Pathname;
22 use Cwd qw(cwd abs_path);
23 DeclareOption('test',,sub {PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption'))))})
24 DeclareOption('multiple',sub {PassOptions('hwexam','sty',ToString(Digest(T_CS('\CurrentOption'))))})
25 DeclareOption('showmeta',sub {PassOptions('metakeys','sty',ToString(Digest(T_CS('\CurrentOption'))))})
26 DeclareOption('extrefs',sub {PassOptions('sref','sty',ToString(Digest(T_CS('\CurrentOption'))))})
27 DeclareOption('notes',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
28 DeclareOption('hints',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
29 DeclareOption('solutions',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
30 DeclareOption('pts',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
31 DeclareOption('min',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
32 DeclareOption('boxed',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
33 DeclareOption('extract',sub {PassOptions('problem','sty',ToString(Digest(T_CS('\CurrentOption'))))})
```

```

34 DeclareOption(undef,sub {PassOptions('omdoc','cls',ToString(Digest(T_CS('\CurrentOption')))); }
35 ProcessOptions();
36 \ltxml.cls)

```

We load `article.cls`, and the desired packages. For the L^AT_EXML bindings, we make sure the right packages are loaded.

```

37 \*cls)
38 \LoadClass{omdoc}
39 \RequirePackage{stex}
40 \RequirePackage{hwexam}
41 \RequirePackage{graphicx}
42 \RequirePackage{a4wide}
43 \RequirePackage{amssymb}
44 \RequirePackage{amstext}
45 \RequirePackage{amsmath}
46 \cls)
47 \*ltxml.cls)
48 LoadClass('omdoc');
49 RequirePackage('stex');
50 RequirePackage('hwexam');
51 RequirePackage('graphicx');
52 RequirePackage('amssymb');
53 RequirePackage('amstext');
54 RequirePackage('amsmath');
55 \ltxml.cls)

```

5 Implementation: The hwexam Package

5.1 Package Options

The first step is to declare (a few) package options that handle whether certain information is printed or not. Some come with their own conditionals that are set by the options, the rest is just passed on to the `problems` package.

```

56 \*package)
57 \DeclareOption{showmeta}{\PassOptionsToPackage{\CurrentOption}{metakeys}}
58 \newif\iftest\testfalse
59 \newif\ifsolutions\solutionsfalse
60 \DeclareOption{test}{\testtrue\solutionsfalse}
61 \newif\ifmultiple\multiplefalse
62 \DeclareOption{multiple}{\multipletrue}
63 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{problem}}
64 \ProcessOptions
65 \package)

```

Then we make sure that the necessary packages are loaded (in the right versions).

```

66 \*package)
67 \RequirePackage{keyval}[1997/11/10]
68 \RequirePackage{problem}
69 \package)

```

Here comes the equivalent header information for L^AT_EX_ML, we also initialize the package inclusions. Since L^AT_EX_ML does not handle options yet, we have nothing to do.

```
70 <*!xml>
71 # -*- CPERL -*-
72 package LaTeXML::Package::Pool;
73 use strict;
74 use LaTeXML::Package;
75 RequirePackage('problem');
76 </!xml>
```

Then we register the namespace of the requirements ontology

```
77 <*!xml>
78 RegisterNamespace('assig'=>"http://omdoc.org/ontology/assignments#");
79 RegisterDocumentNamespace('assig'=>"http://omdoc.org/ontology/assignments#");
80 </!xml>
```

5.2 Assignments

We will prepare the keyval support for the `assignment` environment.

```
81 <*package>
82 \addmetakey{assig}{number}
83 \addmetakey*{assig}{title}
84 \addmetakey{assig}{type}
85 \addmetakey{assig}{given}
86 \addmetakey{assig}{due}
```

The next three macros are intermediate functions that handle the case gracefully, where the respective token registers are undefined.

The `\given@due` macro prints information about the given and due status of the assignment. Its arguments specify the brackets.

```
87 \def\given@due#1#2{%
88 \ifx\assig@given\@empty\else\ifx\assig@due\@empty\else{#1}\fi\fi%
89 \ifx\assig@given\@empty\else{Given {\assig@given}}\fi%
90 \ifx\assig@given\@empty\else\ifx\assig@due\@empty\else{, }\fi\fi%
91 \ifx\assig@due\@empty\else{Due {\assig@due}}\fi%
92 \ifx\assig@given\@empty\else{\ifx\assig@due\@empty\else{#2}\fi}\fi}
```

With them, we can define the central `assignment` environment. This has two forms (separated by `\ifmultiple`) in one we make a title block for an assignment sheet, and in the other we make a section heading and add it to the table of contents.

`assignment@titleblock` This macro prints the title block of a section. If the `multiple` package option is given we make a section heading out of this, and if not, a title block. Note that as `problems` are numbered by section, we also set the section counter in the latter case.

```
93 \ifmultiple
94 \def\assignment@titleblock{%
```

```

95 \@ifundefined{assig@number}{\stepcounter{section}}{\setcounter{section}{\assig@number}}%
96 \section*{\document@hwexamtype~\arabic{section}:~\assig@title\given@due{\(\)}%
97 \addcontentsline{toc}{section}{\document@hwexamtype~\arabic{section}:~\assig@title}%
98 \setcounter{problem}{0}}
99 \else
100 \def\assignment@titleblock{%
101 \setcounter{section}{\assig@number}
102 \begin{center}\bf
103 \Large\@title\
104 \document@hwexamtype~\assig@number:~\assig@title\strut\
105 \large{\given@due()}
106 \end{center}}
107 \fi

```

`assignment@process@keys` this macro collects the keys from its argument and corrects them from the outside.

```

108 \def\assignment@process@keys#1{\metasetkeys{assig}{#1}
109 \ifx\inclassig@title\@empty\else\def\assig@title{\inclassig@title}\fi
110 \ifx\inclassig@type\@empty\else\def\assig@type{\inclassig@type}\fi
111 \ifx\inclassig@number\@empty\else\def\assig@number{\inclassig@number}\fi
112 \ifx\inclassig@due\@empty\else\def\assig@due{\inclassig@due}\fi
113 \ifx\inclassig@given\@empty\else\def\assig@given{\inclassig@given}\fi

for this to work we need to define the \inclassig macros in case no \includeassignment
is ever called.

114 \def\inclassig@title{}
115 \def\inclassig@type{}
116 \def\inclassig@number{}
117 \def\inclassig@due{}
118 \def\inclassig@given{}

```

`assignment`

```

119 \newenvironment{assignment}[1][\assignment@process@keys{#1}]%
120 \assignment@titleblock{}
121 \</package>

122 \<*ltxml>
123 DefEnvironment(' {assignment} OptionalKeyVals:assig',
124 "<omdoc:omgroup ?&KeyVal(#1,'id')(xml:id='&KeyVal(#1,'id')')() "
125 . "assig:dummy='for the namespace'"
126 . "?#locator(stex:srcref='#locator')()>"
127 . "<omdoc:metadata ?#locator(stex:srcref='#locator')()>"
128 . "<dc:title ?#locator(stex:srcref='#locator')()>"
129 . "Assignment ?&KeyVal(#1,'num')(&KeyVal(#1,'num').)()"
130 . "?&KeyVal(#1,'title')(&KeyVal(#1,'title')))"
131 . "</dc:title>"
132 . "?&KeyVal(#1,'given')(<omdoc:meta property='assig:given'>&KeyVal(#1,'given')</omdoc:meta>"
133 . "?&KeyVal(#1,'due')(<omdoc:meta property='assig:due'>&KeyVal(#1,'due')</omdoc:meta>)"
134 . "?&KeyVal(#1,'pts')(<omdoc:meta property='assig:pts'>&KeyVal(#1,'pts')</omdoc:meta>)"
135 . "</omdoc:metadata>"
136 . "#body"

```

```

137 . "</omdoc:omgroup>\n");
138 </ltxml>

139 <*package>
140 \addmetakey[\assig@default@type]{document}{hwexamtype}
141 </package>

```

5.3 Including Assignments

The next command is essentially a glorified `\include` statement, it just sets some internal macros first that overwrite the local points, ¹

EdNote(1)

```

142 <*package>
143 \addmetakey{inclassig}{number}
144 \addmetakey{inclassig}{title}
145 \addmetakey{inclassig}{type}
146 \addmetakey{inclassig}{given}
147 \addmetakey{inclassig}{due}
148 \newcommand{\includeassignment}[2] [] {\metasetkeys{inclassig}{#1}\include{#2}}
149 \newcommand{\inputassignment}[2] [] {\metasetkeys{inclassig}{#1}\input{#2}}
150 </package>
151 <*\ltxml>
152 DefMacro('\includeassignment [] {}', '\input{#2}');
153 DefMacro('\inputassignment [] {}', '\input{#2}');
154 </ltxml>

```

5.4 Typesetting Exams

```

155 <*package>
156 \addmetakey{quizheading}{tas}
157 \newcommand\quizheading[1]{\def\@tas{#1}%
158 \large\noindent NAME: \hspace{8cm} MAILBOX: \[2ex]%
159 \ifx\@tas\empty\else%
160 \noindent YOUR TA: \@for\@I:=\@tas\do{\Box$ \@I\hspace*{1em}}\[2ex]\fi}
161 </package>

162 <*package>
163 \addmetakey{testheading}{min}
164 \addmetakey{testheading}{duration}
165 \addmetakey{testheading}{reqpts}
166 \newenvironment{testheading}[1] [] {\metasetkeys{testheading}{#1}
167 {\noindent\large{Name: \hfill Matriculation Number:\hspace*{2cm}\strut\}[1ex]
168 \begin{center}\Large\textbf{\@title}\[1ex]\large\@date\][3ex]\end{center}
169 {\textbf{You have
170 \ifx\test@heading@duration\empty\testheading@min minutes\else\testheading@duration\fi
171 (sharp) for the test}};\ Write the solutions to the sheet.}\par\noindent
172
173 \newcount\check@time\check@time=\testheading@min
174 \advance\check@time by -\theassignment@totalmin

```

¹EDNOTE: these keys should be done with `\addmetakey`

```

175 The estimated time for solving this exam is {\theassignment@totalmin} minutes,
176 leaving you {\the\check@time} minutes for revising your exam.
177
178 \newcount\bonus@pts\bonus@pts=\theassignment@totalpts
179 \advance\bonus@pts by -\testheading@reqpts
180 You can reach {\theassignment@totalpts} points if you solve all problems. You will only need
181 {\testheading@reqpts} points for a perfect score, i.e.\ {\the\bonus@pts} points are
182 bonus points. \vfill
183 \begin{center}
184   {\Large\em
185 % You have ample time, so take it slow and avoid rushing to mistakes!\\[2ex]
186 Different problems test different skills and knowledge, so do not get stuck on
187 one problem.}\vfill\par\correction@table \\[3ex]
188 \end{center}}
189 {\newpage}
190 \</package>
191 \<*txml>
192 DefEnvironment('{\testheading}OptionalKeyVals:omdoc', '');
193 \</txml>
194 \<*package>
195 \def\testspace#1{\iftest\vspace*{#1}\fi}
196 \def\testnewpage{\iftest\newpage\fi}
197 \def\testemptypage{\iftest\begin{center}This page was intentionally left
198 blank for extra space\end{center}\vfill\eject\else\fi}
199 \</package>
200 \<*txml>
201 DefConstructor('\testspace{ }', '');
202 DefConstructor('\testnewpage', '');
203 DefConstructor('\testemptypage', '');
204 \</txml>

```

`\@problem` This macro acts on a problem's record in the *.aux file. Here we redefine it to generate the correction table.

```

205 \<*package>
206 \def\@problem#1#2#3{\stepcounter{assignment@probs}
207 \def\@test{#2}\ifx\@test\@empty\else\addtocounter{assignment@totalpts}{#2}\fi
208 \def\@test{#3}\ifx\@test\@empty\else\addtocounter{assignment@totalmin}{#3}\fi
209 \xdef\correction@probs{\correction@probs & #1}%
210 \xdef\correction@pts{\correction@pts & #2}
211 \xdef\correction@reached{\correction@reached &}}
212 \</package>

```

`\correction@table` This macro generates the correction table

```

213 \<*package>
214 \newcounter{assignment@probs}
215 \newcounter{assignment@totalpts}
216 \newcounter{assignment@totalmin}
217 \def\correction@probs{prob.}%
218 \def\correction@pts{total}%

```

```

219 \def\correction@reached{reached}%
220 \stepcounter{assignment@probs}
221 \def\correction@table{\begin{tabular}{|l|*{\theassignment@probs}{c|}|p{3cm}|}\hline%
222 &\multicolumn{\theassignment@probs}{c|}|%
223 {\footnotesize To be used for grading, do not write here} &\\\hline
224 \correction@probs & Sum & grade\\\hline
225 \correction@pts & \theassignment@totalpts & \strut\hspace{3cm}\strut\\\hline
226 \correction@reached & & \[.7cm]\hline
227 \end{tabular}}
228 \end{package}

```

5.5 Leftovers

at some point, we may want to reactivate the logos font, then we use

```

here we define the logos that characterize the assignment
\font\bierfont=../assignments/bierglas
\font\denkerfont=../assignments/denker
\font\uhrfont=../assignments/uhr
\font\warnschildfont=../assignments/achtung

\def\bierglas{{\bierfont\char65}}
\def\denker{{\denkerfont\char65}}
\def\uhr{{\uhrfont\char65}}
\def\warnschild{{\warnschildfont\char 65}}
\def\hardA{\warnschild}
\def\longA{\uhr}
\def\thinkA{\denker}
\def\discussA{\bierglas}

```

Finally, we need to terminate the file with a success mark for perl.

```

229 \txml1;

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

L^AT_EX^ML, 5–7

References

- [Koh10a] Michael Kohlhase. *metakeys.sty: A generic framework for extensible Metadata in L^AT_EX*. Self-documenting L^AT_EX package. Comprehensive T_EX Archive Network (CTAN), 2010. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/metakeys/metakeys.pdf>.
- [Koh10b] Michael Kohlhase. *omdoc.sty/cls: Semantic Markup for Open Mathematical Documents in L^AT_EX*. Self-documenting L^AT_EX package. Comprehensive T_EX Archive Network (CTAN), 2010. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/omdoc/omdoc.pdf>.
- [Koh10c] Michael Kohlhase. *problem.sty: An Infrastructure for formatting Problems*. Self-documenting L^AT_EX package. Comprehensive T_EX Archive Network (CTAN), 2010. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/problem/problem.pdf>.
- [Koh10d] Michael Kohlhase. *sref.sty: Semantic Crossreferencing in L^AT_EX*. Self-documenting L^AT_EX package. Comprehensive T_EX Archive Network (CTAN), 2010. URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/stex/sref/sref.pdf>.
- [Ste] *Semantic Markup for LaTeX*. Project Homepage. URL: <http://trac.kwarc.info/sTeX/> (visited on 12/02/2009).