

Using the skills package

Pierre-Amiel Giraud
History & geography teacher
Chambery Middle School
33140 Villenave d'Ornon (France)
pierre-amiel.giraud@ac-bordeaux.fr

Copyright © 2020
Pierre-Amiel Giraud

October 21, 2020

This is the user's guide for version 1.0.0 of the `skills` package.

Contents

1	Introduction	2
1.1	License	2
1.2	Version Notes	2
2	Usage instructions	3
2.1	Loading the <code>skills</code> package	3
2.2	Defining skills	4
2.3	Declaring skills	4
2.3.1	The easy way (exam document class only)	4
2.3.2	The other but also easy way (any document class)	5
2.4	Printing the skills table	6
3	Customizing the display of the skills	7
3.1	The separator in the <code>\skills</code> and <code>\skillquestion</code> commands	7
3.2	Where the skills will be printed	8
3.3	Surrounding the skills: parentheses, brackets, a box, and more	9
3.3.1	Very quick way	9
3.3.2	Using the <code>\skillsenclosure</code> command	10
3.4	The skills table	11

4	Technical informations	11
4.1	Loaded packages	11
4.2	Possible improvements (this is not a roadmap)	11

1 Introduction

The file `skills.sty` provides the `skills` package, which attempts to make it easy for even a \LaTeX novice to prepare proficiency tests, especially in combination with the `exam` document class.

Here, what is called a proficiency test is an exam where questions are assigned one or more skills, and where the proficiency of the pupil is evaluated for each skill. It seems it can also be called, depending on the context and the country, skills assessment or skill-based assessment. As far as I know, proficiency tests are more often used in French-speaking areas, where they are called "*évaluations par compétences*". This package has been designed with the French approach in mind. If you are using proficiency tests but this package doesn't fill your needs, your suggestions or code contributions are welcome.

The `skills` package has also been designed to be best used within the `exam` document class. Thus, almost all command names are very similar. However, the package can be used within a variety of document classes: only a very small subset of the package commands is specific to the `exam` document class.

Some other packages, such as the `competences` packages by Christophe Bares, follow the same goal but with different approaches. They might be of some interest if the present `skills` package doesn't fill your needs.

The latest version of `skills.sty` (possibly a beta test version) should always be available at <https://framagit.org/pagiraud/skills/>.

1.1 License

- This work may be distributed and/or modified under the conditions of the \LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of \LaTeX version 2003/12/01 or later.
- This work has the LPPL maintenance status "maintained".
- The Current Maintainer of this work is Pierre-Amiel Giraud
- This work consists of the files `skills.sty` and `skillsdoc.tex`.

1.2 Version Notes

v1.0.0 Initial release

2 Usage instructions

This section details how to use the package and the commands it provides, from basic usage to customization. Some features are only available when the package is loaded within the exam document class. They will be indicated as such.

2.1 Loading the skills package

To use the skills package, you must load it, like every package, with the command

```
\usepackage{skills}
```

Package options (advanced use only)

The package has only three options:

- `makenoidxglossaries` (default)
- `donotmakenoidxglossaries`
- `counter=<value>`

The `donotmakenoidxglossaries` option is of any use only if you want to use the `glossaries` package, upon which `skills` is based, in your document. It deactivates the `\makenoidxglossaries` command included in the `skills.sty` file, so that you can handle your glossaries with the indexing option of your choice. Please refer to the `glossaries` package documentation for more details.

So, if you want to use the `glossaries` and the `skills` packages together in your document, it is advised to load the `skills` package as follows:

```
\usepackage[donotmakenoidxglossaries]{skills}
```

The `counter` option is to be used only if you to change the default counter for grouping the skills in the skills table (question for the exam class, section for other classes). The section counter was chosen because it is probably the most widespread logical counter (as opposed to the page counter). This very document, written with the exam class, loads the `skills` package with

```
\usepackage[counter=section]{skills}
```

Indeed, multiple `questions` environments are used all along the documentation, and using `question` as a counter would needlessly clutter the skills table. Any available counter can be used.

2.2 Defining skills

Skills must be defined in the preamble, using the `skilldef` command:

```
\skilldef{label}{reference}{description}
```

The first argument, `label`, is a simple keyword that will be used in your document to indicate that you are evaluating mastery of this skill. The `reference` is the code of the skill in your competency framework. Finally, `description` is the place where you can put the title of your skill along with some description.

Defining a skill doesn't imply that it will be used. Thus, you can define your whole competency framework (or a significant subset) in a file and then simply load it with the `\input` command in every proficiency test you make: all skills will be available for declaration, but only those explicitly declared in the document body will appear on the output file.

For the sake of example, we will define three skills that will be used all along this manual:

```
\skilldef{writing}{L1.1}{Writing to argue and write to communicate  
and share ideas}  
\skilldef{vocabulary}{H3.2}{Learning specific vocabulary and using  
it in context}  
\skilldef{situationInTime}{C1.1}{Situating in time and elaborating  
chronological landmarks}
```

2.3 Declaring skills

There are two ways for declaring skills. The first, and the simplest, is available only if the loaded document class is `exam`. The second one is class agnostic.

2.3.1 The easy way (exam document class only)

When the `skills` package is used together with the `exam` document class, the `skillquestions` environment and the `\skillquestion` command become available to the user.

They behave like the `questions` environment and `\question` command from the `exam` document class, except they have one more optional argument, used for declaring skills:

```
\begin{skillquestions}[optional list of comma separated skills]  
\skillquestion[optional list of comma separated skills][optional number of  
points for the question] Some question  
\end{skillquestions}
```

The skills declared as arguments to the `skillquestions` environment are global: they are not linked to any specific question in the exam, but are rather evaluated throughout the exam and will be flagged as such in the skills table (see subsection 2.4). Those declared as arguments to a `\skillquestion` command are evaluated only in the questions where they are declared.

Some details need specific attention:

- The label chosen when defining a skill is used to declare it in the exam.
- If you don't want to declare global skills or if, for any reason, you don't want to declare skills for a question, you should omit the square brackets containing them: don't use empty brackets.
- Don't use spaces in the skills list: write `writing,vocabulary`, not `writing, vocabulary`.
- A skill can't be declared as global and question specific in the same exam. If you try to do so, the question-level declaration will overwrite the global one.
- The `skillquestions` environment and the `\skillquestion` command are only wrappers around the `questions` environment and the `question` command from the exam document class. As such, they can be freely mixed: `\question` and `\skillquestion` can be used together both within the `questions` and `skillquestions` environments. Points of `\question` and `skillquestion` add up nicely.

A very basic example, using the three previously defined skills, could thus be:

```
\begin{skillquestions}[writing]
\skillquestion A first question without any skill declared
\skillquestion[situationInTime,vocabulary] A second question with two skills
\skillquestion[vocabulary][2] A third question with one skill and 2 points.
\question[3] A last question without any skill declared but 3 points.
\end{skillquestions}
```

And would results in:

1. A first question without any skill declared
2. (C1.1 H3.2) A second question with two skills
3. (2 points) (H3.2) A third question with one skill and 2 points.
4. (3 points) A last question without any skill declared but 3 points.

As you can see, the declared global skill isn't printed: it will be shown only in the skills table (see subsection 2.4), which summarizes all declared skills in the exam.

2.3.2 The other but also easy way (any document class)

Skills and global skills can also be declared with the `\skills` and `\globalskills` commands:

```
\skills{list of comma separated skills}
\globalskills{list of comma separated skills}
```

As for `skillquestion` and `skillquestions`, the separator must be only a comma: don't use spaces. In the default configuration (see subsection 3.2 for customization), the skills will appear right where you write them. If you type:

```
Some text \skills{vocabulary} that goes on and on and on.
```

You will get:

```
Some text (H3.2) that goes on and on and on.
```

So, the `\skills` command can be used in any class. That includes the `exam` document class, if you want, for instance, to declare skills at the `subquestion` or `part` level. This won't have any effect on the way skills are grouped in the skills table. By default, they are grouped per question in the `exam` class and per section for other document classes.

The `\globalskills` command doesn't print anything. If you type:

```
Some text \globalskills{writing} that goes on and on and on.
```

You will get:

```
Some text that goes on and on and on.
```

But the place where you put the `\globalskills` command has a direct effect on the skills table, as the skills appear in the order they are firstly declared. What's more, typing the `globalskills` command anywhere in the document will likely result in undesired commas in the skills table. Therefore, as a rule of thumb, all global skills should be declared at once, in the same `\globalskills` command:

- right after `\begin{document}` if you want the global skills to be printed at the beginning of the skills table;
- right before `\end{document}` if you want them to be printed at the end of the skills table.

The `\globalskills` command shouldn't be used at all in the `exam` class, at least in the `skillquestions` environment is used.

2.4 Printing the skills table

There would be very little interest in defining then declaring skills evaluated in an exam if they are not listed somewhere. With the `skills` package, this is done with the command:

```
\skillstable[optional sorting argument]
```

So, for most users, a simple `\skillstable` is enough. If you are unhappy with the default sorting (skills sorted in the order they are firstly declared in the document), which is equivalent of typing `\skillstable[use]`, some other options are available: `word`, `letter`, `standard`, `def`, `nocase`, `case`. For more information on these, please refer to the `glossaries` package documentation (chapter 10, Displaying a glossary).

Here, if I type:

`\skillstable`

You get:

Skill	Questions	Proficiency
C1.1 Situating in time and elaborating chronological landmarks.	2, 3	
H3.2 Learning specific vocabulary and using it in context.	2, 3	
L1.1 Writing to argue and write to communicate and share ideas.		

The global skills can easily be seen as no question number appears in the Questions column.

If the `\globalskills` command is used to declare global skills, the place of the skills in the table is determined by the place the command is put in the document (see subsection 2.3.2). If the global skills are declared with the optional argument of the `skillquestions` environment, by default, they are put after the other skills in the table, but this can be changed by typing:

```
\renewcommand{\putglobalskills}{before}
```

3 Customizing the display of the skills

For now, we have only seen how to setup a proficiency test, without modifying the default appearance, that is:

- In the `\skills` and `\skillquestion` commands, if several skills are declared, a blank space is used as a separator.

3.1 The separator in the `\skills` and `\skillquestion` commands

The default blank space can be changed by renewing the `\skillssep` command. For instance, if you want the skills to be separated with a comma followed by a blank space, type:

```
\renewcommand{\skillssep}{, }
```

And now, if you type:

```
\skills{vocabulary,situationInTime}
```

You get:

(H3.2, C1.1)

3.2 Where the skills will be printed

The default is that the skills references will be inserted at the beginning of the question with `\skillquestion` or right where the `\skills` command is used, but, for one-sided documents

- the command `\skillsinmargin` (or, equivalently, the command `\skillsinleftmargin`) will cause the skills references to be set in the left margin,
- the command `\skillsinrightmargin` will cause the skills references to be set in the right margin,
- the commands `\noskillsinmargin` and `\noskillsinrightmargin` are equivalent, and either of them will revert to the default situation.

For two-sided documents, `\skillsinmargin` and `\skillsinleftmargin` print skills in the inner margin, while `\skillsinrightmargin` prints them in the outer margin.

If you chose to print the skills references inline (default), you can move where they appear simply by changing the place you type the `\skills` command. For instance, if you want to print the skills at the end of a question, in the exam class, you can type:

```
\begin{questions}
  \skillquestion A very good question. \skills{vocabulary}
\end{questions}
```

You will get:

1. A very good question. (H3.2)

When the skills are set in a margin, they start on the line they were declared. Sometimes, for instance if you declare bot points and skills for a `\skillquestion` to be shown in the same margin, the skills and points overlap, so you need to move vertically the skills using:

```
\skillsinmarginadjust{some length}
```

Let's see an example:

```
\begin{questions}
  \pointsinrightmargin
  \skillsinrightmargin
  \skillquestion[vocabulary][2] Some question
  \skillsinmarginadjust{\baseline}
  \skillquestion[vocabulary][2] Some question
\end{questions}
```

1. Some question
2. Some question

(H3.2)

(2)

(H3.2)

`\skillsinmarginadjust` can be changed as often as you need.

3.3 Surrounding the skills: parentheses, brackets, a box, and more

By default, skills references are enclosed in parentheses. The package provides two ways for changing this.

3.3.1 Very quick way

Following the syntax used in the exam class (but not requiring you to use it), you can very quickly change the way skills are enclosed.

For instance, if you want to have the skills references enclosed in brackets:

```
\bracketedskills
```

For example, if you give the command `\bracketedskills`, typing some questions will produce something as follows:

1. [H3.2] Why is there air?
2. [H3.2 C1.1] What if there were no air?

If you prefer having the skills enclosed in a box instead of in parentheses, give the command

```
\boxedskills
```

For example, if you give the command `\boxedskills`, then the questions typed above will produce

1. H3.2 Why is there air?
2. H3.2 C1.1 What if there were no air?

If you give the commands `\boxedskills` and `\skillsinmargin`, then the above questions will produce

H3.2 1. Why is there air?

H3.2
C1.1 2. What if there were no air?

If you want the skills to be not enclosed at all, you can give the command:

```
\onlyskills
```

If you give the commands `\onlyskills` and `\skillsinrightmargin`, then the above questions will produce

1. Why is there air? H3.2
2. What if there were no air? H3.2
C1.1

Other combinations of these commands will produce similar effects.

If you want to switch back and forth between formats during the proficiency test, you can do so by giving one of the commands

```
\bracketedskills
\nobacketedskills
\boxedskills
\noboxedskills
\onlyskills
\notonlyskills
```

whenever you want to switch. The commands `\nobacketedskills`, `\noboxedskills` and `\notonlyskills` are equivalent: they all return to the default of putting parentheses around the skills references.

3.3.2 Using the `\skillsenclosure` command

It is possible to enclose the skills references with any string of characters, using the `\skillsenclosure` command:

```
\skillsenclosure{opening string}{closing string}
```

Actually, `\bracketedskills` and `\onlyskills` definitions are

```
\newcommand{\bracketedskills}{\skillsenclosure{[]}{[]}}
\newcommand{\onlyskills}{\skillsenclosure{}{}}
```

But one can be more imaginative with, for instance

```
\skillsenclosure{**}{**}
```

or even

```
\skillsenclosure{oO}{Oo}
```

This last example would produce

1. oO H3.2 Oo Why is there air?
2. oO H3.2 C1.1 Oo What if there were no air?

The `\boxedskills` command uses another mechanism: after triggering an `\onlyskills`, it puts a `\fbox` or a `\parbox` in a `\fbox` around the skills, depending on whether the latter are printed inline or in a margin. It means one can use `\skillsenclosure` after `\boxedskills`. For instance, typing the following commands

```
\boxedskills
\skillsenclosure{oO }{ Oo}
```

before the questions printed above would produce

1. oO H3.2 Oo Why is there air?
2. oO H3.2 C1.1 Oo What if there were no air?

3.4 The skills table

We have already talked about the `\putglobalskills` command (see subsection 2.4).

It is also possible to rename the columns of the table by renewing some commands before typing `\skillstable`:

- `\renewcommand{\descriptionname}{Some name}` for the first column (default: Skill)
- `\renewcommand{\pagelistname}{Some name}` for the second column (default: Questions)
- `\renewcommand{\skilllevelname}{Some name}` for the third column (default: Proficiency)

If you are using the package `polyglossia` or `babel` with French as main language, these names are localized by the `skills` package as "Compétences", "Questions" and "Maîtrise".

It is also possible to totally change the look of the skills table with the command

```
\renewglossarystyle{skillstable}{A new skills table look}
```

This requires some knowledge about the `glossaries` package. Please refer to its documentation, especially section 15.2 (Defining your own glossary style).

4 Technical informations

4.1 Loaded packages

The `skills` package loads some other packages, namely and unordered: `glossaries`, `marginnote`, `xparse`, `tabularx`, `etoolbox`, `iftex` and `kvoptions`.

They are all loaded without any option.

4.2 Possible improvements (this is not a roadmap)

In next versions, if any, some improvements will be made:

- A code cleanup is needed, because I started coding the package with the assumption that the naming convention for commands in the \LaTeX world was all lowercase. Then, I understood that the convention consisted rather in using lowercase names for user commands and camel case names for other commands.
- A deeper code refactoring would replace \LaTeX commands by plain TeX control sequences and limit the dependency to external packages. This lower dependency would also help mitigating the probability of side effects.
- Allowing for multiple skills tables on the same document, for instance on a per part or per section basis.

- Offering some competency framework, that could be chosen with a key/value system as a package option.