CurVe – a LaTeX 2ε class package for making **Cur**ricula **V**ita**e***

Didier Verna
mailto:didier@lrde.epita.fr
http://www.lrde.epita.fr/~didier

v1.16 (2010/12/14)

Abstract

CurVe provides a LaTeX 2ε class that hopefully will make your life easier when you want to write your CV. It provides you with a set of commands to create rubrics, entries in these rubrics etc. CurVe will then properly format your CV for you (possibly splitting it onto multiple pages), which is usually the most painful part of CV writing. Another nice feature of CurVe is its ability to manage different CV "flavors" simultaneously. It is in fact often the case that you want to maintain slightly divergent versions of your CV at the same time, in order to emphasize on different aspects of your background.

Contents

*CurVe homepage: http://www.lrde.epita.fr/~didier/software/latex.php#curve

1 Installation

1.1 Extraction

If you are building CurVe from the tarball you need to execute the following steps in order to extract the necessary files. CurVe also requires the DoX package (version 2.0, release date 2009/09/21 or later), to build. It is not required to use the package.

```
[pdf]latex curve.ins
[pdf]latex curve.dtx
[pdf]latex curve.dtx
makeindex -s gind curve.idx
[pdf]latex curve.dtx
[pdf]latex curve.dtx
```

After that, you need to install the generated documentation and style files to a location where LaTeX can find them.

1.2 TDS-compliant layout

For a TDS-compliant layout, the following locations are suggested:

```
[TEXMF]/tex/latex/curve/curve.cls
[TEXMF]/doc/latex/curve/curve.[pdf|dvi]
```

1.3 AUC-TeX support

AUC-TeX is a powerful major mode for editing TeX documents in Emacs. In particular, it provides automatic completion of command names once they are known. CurVe supports AUC-TeX by providing a style file named `curve.el` which contains AUC-TeX definitions for the relevant commands. This file should be installed in a place where AUC-TeX can find it (usually in a subdirectory of your LaTeX styles directory). Please refer to the AUC-TeX documentation for more information on this.

As of version 1.2, CurVe has an improved AUC-TeX support. Most notably, the command `M-Ret` will insert an `\entry*` macro within a `rubric` environment. Also, the `\makerubric` macro handling now removes both the file extension and the file flavor extension.

1.4 Examples

Once you have installed CurVe, you might want to start with processing the examples in the `examples/` directory. This will give you an idea of what a non customized CV looks like with CurVe. You can also throw an eye to my own CV, which is written with CurVe and has some more fancy hackery on top of it. It's in French, but only the appearance is important for you. My CV can be found at `http://www.lrde.epita.fr/~didier/about/cv.pdf`.

2 Overview

2.1 Document Layout

2.1.1 Headers

A CurVe CV begins with two optional headers (upper left and upper right) in which you usually put your name, address, email, whether you're married and so on. These headers will respectively be left and right aligned. As of version 1.4,

CurVe lets you insert a small identity photo in the headers, either on the left, on the right, or between them.

2.1.2 Titles

After these headers come an optional title and/or subtitle, which can be centered on the page, or flushed either left or right.

2.1.3 Rubrics

The remaining of the document is composed of sections called "rubrics" in the CurVe terminology. A rubric represents a major topic that you want to detail in your CV. Typical rubrics are "Education", "Professional Experience" and the like. Rubrics have a title (centered by default) and appear under the form of properly aligned "entries" (see below). If a rubric has to be split across different pages, its title will be repeated automatically.

2.1.3.1 Entries

An entry is an item of information related to the rubric under which it appears. An entry has a "contents", and an optional "key" under which it is classified. For instance, under the "Education" rubric, you could state that you got a Ph.D. in computer science in the year 2000. In that case, the year would be the entry's key, and the "Ph.D. in computer science" part would be the entry's contents. CurVe aligns both keys and contents together. Keys are optional in order for you to classify several entries together (without repeating the same key over and over again).

2.1.3.2 Subrubrics

Additionally, you might want to further split your rubrics into "subrubrics". For instance, in my own CV, I have a "Professional Experience" rubric, with three subrubrics: "Teaching", "Research" and "Development". This can be accomplished with a special command. Subrubrics are displayed in alignment with the entries' contents by default, but are formatted differently so that they remain distinguishable.

2.2 Document Structure

2.2.1 Source File Splitting

CurVe is based on the `LTXtable` package by David Carlisle. I won't go into gory details, but this has an important implication: **each rubric must be in its own separate file**. In other words, your CV's main source file is really a skeleton whose major task is to include the different rubrics from their respective source files.

This is not much of a hassle, really, and it actually made my life easier when I implemented the "flavor" mechanism described below.

2.2.2 Flavors

It is often desirable to maintain several slightly divergent versions of one's CV at the same time. For instance, when I was looking for a job some time ago, I had a version of my CV emphasizing on Artificial Intelligence, and another emphasizing on Distributed Virtual Reality. Only the title and some entries in the "Professional Experience" rubric were a bit different; the main skeleton basically remained the same.

CurVe provides an easy-to-use mechanism for maintaining different "flavors" of your CV at the same time. You basically write different versions of (some of) your rubrics in different files, tell CurVe which flavor you want to format (CurVe can even ask you which one to use directly) and that's it. CurVe will use the global skeleton, and whenever it finds a rubric file specialized for that particular flavor, it will use it. Otherwise, it will simply fall back to the default one (no particular flavor).

3 Using CurVe

First of all, please note that CurVe requires the presence of `ltxtable`, `ifthen`, `calc` and `filehook`. If you're using the identity photo feature, the `graphicx` package is also needed. You don't have to load them explicitly though. As long as LaTeX can locate them, they will be used automatically.

3.1 Skeleton File

Say `\documentclass[`⟨*options*⟩`]{curve}` at the beginning of your skeleton file in order to use CurVe. The available options are described along the text, where appropriate.

3.1.1 Headers

3.1.1.1 Header Components

`\leftheader` `{`⟨*contents*⟩`}`
`\rightheader` ...

The `\leftheader` and `\rightheader` macros define respectively the contents of the upper left and upper right headers. They can be used in the document's preamble only. The headers will respectively be flushed to the left and to the right.

`\photo` `[`⟨*hpos*⟩`]{`⟨*file*⟩`}`

If you want to insert a small identity photo into the header part, you can use the `\photo` macro (available since version 1.4). It takes a mandatory argument in which you pass the image file name, as you would to `\includegraphics`. This macro also takes an optional argument which lets you specify the horizontal position of the photo: the values can be `l` (the default), `c` or `r` meaning that the photo will appear on the left, center, or right.

3.1.1.2 Header Layout

The headers' horizontal layout is further controled by three additional macros.

`\photoscale` `{`⟨*scale*⟩`}`

The `\photoscale` macro specifies the amount of text width that the photo should

occupy. This should be a number between 0 and 1. By default, 0.1 is used (meaning 10% of `\textwidth`).

`\photosep` The `\photosep` macro is a LaTeX length that specifies the space to leave between the side of the photo and the next headers's text. This is used only when the photo is on the left or right. By default, `10pt` is used.

`\headerscale` `{`⟨*scale*⟩`}`
Finally, `\headerscale` specifies the proportion of the *remaining* space that the *left* textual header should occupy. It works like `\photoscale` and amounts to 0.5 by default.

Let me take an example to make this clearer. Suppose you have a `\photoscale` of 0.1 and a `\photosep` of `10pt`. The *remaining* space, that is, the space occupied by the textual headers, amounts to 90% of the text width, minus 10 points. If you then specify a `\headerscale` of 0.6, then the left header will take 60% of that remaining space, and the right one the other 40%.

`\headerspace` `\headerspace` is the amount of extra vertical space to put after the headers. This is a LaTeX length that defaults to `10pt`.

3.1.1.3 Making Headers

`\makeheaders` `[`⟨*vpos*⟩`]`
If you have defined headers, make them appear by calling `\makeheaders` just after the beginning of your document. Note that calling this macro assumes that you have previously defined both headers (possibly empty, though). Otherwise, an error will be signaled. As of version 1.4, the `\makeheaders` command accepts an optional argument that controls the vertical alignment. When given, this argument must be either `t` (for top), `b` (for bottom) or `c` (for center; the default).

3.1.2 Titles

3.1.2.1 Title Components

`\title` `{`⟨*text*⟩`}`
`\subtitle` ...
The `\title` and `\subtitle` macros take one mandatory argument which define respectively your CV's title and subtitle. They can be used in the document's preamble only.

3.1.2.2 Title Layout

`\titlealignment` `{`⟨*hpos*⟩`}`
By default, titles are centered on the page. However, you can also have them aligned to the left or right side of the page. To specify your preferred title alignment, call `\titlealignment` with an argument of either `l`, `c` or `r`, the meaning of which should be obvious. You can also achieve the same effect by passing an optional argument to `\maketitle` (see below).

`\titlespace` `\titlespace` is the amount of extra vertical space to put after the title(s). This is a LaTeX length that defaults to `0pt`.

`\titlefont` `{`⟨*cmds*⟩`}`
`\subtitlefont` ...
The `\titlefont` and `\subtitlefont` macros take one mandatory argument which redefine the fonts to use for the title and the subtitle. They can be used in the

document's preamble only. By default, `\Huge\bfseries` and `\Huge\itshape` are used respectively.

3.1.2.3 Making Titles

`\maketitle` `[⟨hpos⟩]`

If you have defined a title (and possibly a subtitle), make it (them) appear by calling `\maketitle` after the beginning of your document, and just after `\makeheaders` if you happen use it. It is possible to omit the subtitle, but if you call `\maketitle` without having defined at least a title, an error will be signaled. `\maketitle` accepts an optional argument for specifying the title alignment scheme. This argument is the same as in `\titlealignment`, and takes precedence over it.

3.1.3 Flavors

As you already know, each rubric must reside in its own separate file. For instance, if you have a "Professional Experience" rubric, you would write its contents into a file named `experience.tex`. The flavor mechanism works by assigning a pre-extension to rubric file names. For instance, suppose you want to make a special flavor of your CV emphasizing on "distributed virtual reality". You would call this flavor "dvr", and write the modified "Professional Experience" rubric into a file named `experience.dvr.tex`.

`\flavor` `{⟨key⟩}`

The `\flavor` macro takes one mandatory argument which specifies the flavor to use (in our example, `dvr`). Although this might be of little use, it is possible to change the flavor anywhere, even right in the middle of your CV's skeleton.

`ask` Instead of using the `\flavor` macro, you can make CurVe ask you at run-time which flavor to use by passing the `ask` option to it.

3.1.4 Rubrics

Apart from making headers and titles, the body of your skeleton file will usually contain nothing but directives to include the different rubrics of your CV.

`\makerubric` `{⟨name⟩}`

To include a rubric in your document, use `\makerubric`. This macro takes one mandatory argument which specifies the rubric to include at that point. The argument actually corresponds to the rubric file name **without any extension**. Continuing our previous example, you would say `\makerubric{experience}`. First, CurVe will try to find such a rubric file specific for the current flavor in use, (e.g. `experience.dvr.tex`). If that fails, it will fall back to a non-flavored file (here, `experience.tex`). This allows you to specialize only the required rubrics and use the default ones otherwise.

`\rubricafterspace` As of version 1.12, CurVe provides a new LaTeX length, `\rubricafterspace` controlling the amount of extra vertical space to put after a rubric (hence, before the next one). The default is `0pt`.

3.2 Rubric Files

3.2.1 The `rubric` Environment

`rubric` `{`⟨*title*⟩`}`
The whole contents of a rubric file must be enclosed in a `rubric` environment. This environment takes one mandatory argument which specifies the rubric's title. When a rubric crosses several pages, its title is restated with a "continuation" text appended.

`\rubricalignment` `{`⟨*hpos*⟩`}`
As of version 1.6, the rubric titles horizontal alignment can be changed thanks to the `\rubricalignment` macro. Possible values for its mandatory argument are `l`, `c` and `r` (meaning left, centered, or right relative to the whole text width), and `cl` and `cc` (meaning left or centered relative to the entries'contents). By default, rubric titles are centered (`c`).

`\rubricfont` `{`⟨*cmds*⟩`}` The `\rubricfont` macro takes one mandatory argument which redefines the font to use for rubric titles. By default, `\Large\bfseries` is used.

`\rubricspace` `\rubricspace` is the amount of extra vertical space to put after the rubric title. This is a LaTeX length that defaults to `10pt`.

3.2.2 Standard Entries

3.2.2.1 Creating rubric entries

`\entry` `[`⟨*key*⟩`]{`⟨*contents*⟩`}`
You create entries in your rubrics by calling the `\entry` macro. The first (optional) argument specifies the key, and the second (mandatory) one specifies the contents. Both keys and contents are aligned within each rubric.

`\entry*` `[`⟨*key*⟩`]`
Actually, the `\entry` macro was somewhat ill-designed in the first place. The `rubric` environment pretty much behaves as an `itemize` one, hence the idea of using an `\item`-like syntax. As of version 1.2, CurVe provides an `\entry*` macro which behaves like `\item` in lists: it takes the same first optional argument as the non starred version, but has no other argument. The entry's contents simply consists of the text following the macro call, up to the next `\entry`, `\entry*` or `\subrubric` (see below) call.

3.2.2.2 Entries layout

`\keyalignment` `{`⟨*hpos*⟩`}`
As of version 1.7, entries'keys horizontal alignment can be changed thanks to the `\keyalignment` macro. Possible values for its mandatory argument are `l`, `c` and `r` (meaning left, centered, or right). By default, keys are left aligned (`l`).

`\keyfont` `{`⟨*cmds*⟩`}`
The `\keyfont` macro takes one mandatory argument which redefines the font to use for the entries' keys. By default, the standard document font is used.

`\prefix` `{`⟨*cmds*⟩`}`
Each entry's contents can be prefixed with a visual clue (a symbol for instance). This comes in handy to make a clear distinction between different entries sharing the same key (which is not repeated). The `\prefix` macro takes one mandatory argument which redefines the prefix to use. By default, `\textbullet` is used.

Note that as of version 1.11, CurVe forces the prefix to be empty in bibliographic entries (see section 3.3).

`skipsamekey` While maintaining your CV, you might end up reorganizing your entries and even get entries with the same key. Normally, CurVe blindly prints the keys regardless of their values. If you don't want repetition, you would have to remove keys by hand which can be cumbersome. As of version 1.10, CurVe can skip all but the first of a series of identical keys automatically, provided that you use the `skipsamekey` option. Note that as of version 1.11, CurVe disables this mechanism in bibliography rubrics (see section 3.3).

3.2.3 Plain text entries

`\text` `{`⟨*contents*⟩`}`

Sometimes, you may feel the need to insert plain text in the middle of a rubric, without the fancy key/prefix formatting stuff. As of version 1.16, CurVe provides a `\text` macro for that. Plain text occupies the whole page width. If you whish to further distinguish it from the surrounding entries, you can use `\par` to add vertical space before and/or after the text.

3.2.4 Invisible entries

The most frequently asked question about CurVe is probably whether it is possible to align entries across several rubrics. This is (currently) not possible automatically because rubrics are typeset as independant tables. However, a manual solution boils down to enlarging too narrow entries (keys, actually).

`\noentry` `{`⟨*key*⟩`}`

As of version 1.11, CurVe provides a convenience macro to ease this process: `\noentry`. This macro takes one mandatory argument, a key that will be used in the entries alignment calculation. However, this command will not produce any text. So if you want all your rubrics to share the same alignment, you typically spot the longest key in your CV, and issue a `\noentry{this long key}` call in all other rubrics.

3.2.5 Subrubrics

`\subrubric` `{`⟨*title*⟩`}`

Within a single rubric, you can further separate entries into subrubrics. In order to do this, the `\subrubric` macro is provided. Its mandatory argument specifies the subrubric's title.

`\subrubricalignment` `{`⟨*hpos*⟩`}`

As of version 1.6, the subrubrics horizontal alignment can be changed thanks to the `\subrubricalignment` macro. Possible values for its mandatory argument are `l`, `c` and `r` (meaning left, centered, or right relative to the whole text width), and `cl` and `cc` (meaning left or centered relative to the entries'contents). By default, subrubrics are left-aligned with the entries' contents (`cl`).

`\subrubricfont` `{`⟨*cmds*⟩`}`

The `\subrubricfont` macro takes one mandatory argument which redefines the font to use for the subrubrics. By default, `\Large\itshape` is used.

`\subrubricspace` `\subrubricspace` controls the amount of extra vertical space to put after subrubrics. This is a LaTeX length that defaults to `5pt`. `\subrubricbeforespace`
`\subrubricbeforespace`

controls the amount of extra vertical space to put *before* a subrubric when there are entries above. This is a LaTeX length that defaults to `20pt`.

3.3 Bibliography

Most scientists include their own list of publications in their CV, so CurVe has support for different forms of bibliography.

3.3.1 Manual bibliography

The first thing you can do is create your own bibliography manually (I mean, just like an ordinary rubric), and although this may appear boring, I actually encourage people to do so for at least three reasons (only my opinion of course):

- A CV should be strictly formatted and coherent in layout. Bibliography is no exception to this rule. In other words, it is prettier to have your publications formatted like the rest of your CV.
- Automatic bibliography generation tools produce references, which is silly in a CV because you don't actually reference your papers anywhere (or do you?). So better to sort them another way, like, by year of publication as I do in my own CV.
- Manually adding, like, what? Half a dozen papers a year in your CV is not that much of a burden after all.

3.3.2 The `bibliography` environment

`thebibliography` `{`⟨*key*⟩`}`

`\bibitem` `[`⟨*label*⟩`]{`⟨*ref*⟩`}`

Some people however have expressed the wish of having standard bibliography support in CurVe. Version 1.2 provides that. The standard `thebibliography` environment is now supported along with its `\bibitem` companion. The behavior is actually that of a `rubric` environment with its `\entry*` companion (with an empty prefix however). This fact has two implications: firstly, the argument to the environment is unused in CurVe (but remains for compatibility with the rest of LaTeX) because CurVe itself formats the keys and contents properly aligned. Secondly, the bibliographic environment **must** reside in its own file, as any other rubric. Don't forget that if you happen to write the environment manually.

`openbib` As of version 1.16, the `openbib` option is supported. When used, bibliographic blocks appear on different lines instead of in a single paragraph. Note that prior to version 1.16, this was actually the default behavior, which was not quite right (the standard classes don't openbib by default).

`revbib` As of version 1.14, it is possible to count bibliographic items in a reverse order, which comes in handy when you display your publications from the most recent to the oldest one. The `revbib` option triggers this behavior. Note that this also works when you use BibTeX (see below).

3.3.3 BibTeX

`\nocite` `{`⟨*citations*⟩`}`

`\bibliographystyle` `{`⟨*style*⟩`}`

\bibliography {⟨*files*⟩}
If you want to use BibTeX, that's also possible of course. Do it as you would do in a random paper. You will probably issue a `\nocite{*}` command followed by a call to `\bibliography`. CurVe uses the `bbl` file as a rubric one.

3.3.4 Compatibility concerns

- `bibentry`: CurVe is compatible with the `bibentry` package since version 1.9. Note that there is an incompatibility between `bibentry` and `hyperref` with a workaround described in the `hyperref` README file. However, CurVe takes care of this problem for you so you *must not* use the workaround described there.
- `multibbl`: CurVe is compatible with the `multibbl` package since version 1.12.
- `splitbib`: CurVe is compatible with the `splitbib` package since version 1.16. Note however that only one level of category is allowed (categories appear as subrubrics).

3.4 Standard Class Features

3.4.1 Page Size and layout

*paper
landscape
The `a4`, `a5`, `b4`, `letter`, `legal` and `executive` "paper" options allow you to select the type of page format you want. By default, `letterpaper` is used. The `landscape` options switches the horizontal and vertical settings. I'm not sure why I propose this option. Nobody wants to write a CV in landscape mode, right?

oneside
twoside
As of version 1.6, CurVe also supports the standard `oneside` and `twoside` class options. By default, `oneside` is used. In `twoside` mode, odd and even pages have a different geometry and headings layout.

3.4.2 Font Size

10pt
11pt
12pt
The `10pt`, `11pt` and `12pt` options let you choose the size of the default font you want to use. By default, `10pt` is used.

3.4.3 Output Mode

final
draft
In `draft` mode, a black rule will be drawn at the end of overfull lines (as done by standard classes). Due to CurVe using the `LTXtable` package (and in case `longtable` prior to version 4 is used by it), a call to `\setlongtables` is performed in `final` mode. Please refer to the next section for more information on this. By default, `final` is used.

3.4.4 Page styles

\pagestyle {⟨*style*⟩}
\thispagestyle {⟨*style*⟩}
As of version 1.6, CurVe supports the standard LaTeX page style mechanism. Available styles are `empty`, `plain`, `headings` and `myheadings`. These styles have their usual meaning, given that rubric and subrubric names are used for marking purpose (the equivalent of chapters and sections in books for instance). By default, the page style is `empty`.

3.4.5 Internationalization

CurVe currently supports 12 languages via the following options: `english`, `french` (or `francais`), `spanish`, `portuguese` (or `portuges`), `brazilian` (or `brazil`), `italian`, `german`, `ngerman`, `dutch`, `danish`, `swedish` and `polish`. The `german` and `ngerman` options are currently equivalent; so are the `portuguese` and `brazilian` ones.

If you want a finer grain on the language-dependent parts of CurVe, the following macros are provided.

`\continuedname` `{`⟨*continuation text*⟩`}`

The `\continuedname` macro takes one mandatory argument which redefines the continuation text output when rubrics extend across several pages. By default, "⟨*space*⟩(continued)" is used in English. Although this might be of little use, it is possible to change the continuation text in the middle of your document, provided that you do so outside the `rubric` environment.

`\listpubname` `{`⟨*title*⟩`}`

The `\listpubname` macro takes one mandatory argument which redefines the title of the bibliographic section (when you use the provided bibliography support). By default, "List of Publications" is used in English. Note that for compatibility with the `multibbl` and `multibib` packages, CurVe honors the existence of `\bibname` or `\refname` macros prior to `\listpubname` for deciding which title to give to the bibliographic rubric.

4 Hints, Tips'n Tricks

Here are some tips that I use for my own CV. You might find them of some interest.

4.1 Page Geometry

First of all, it is common to have very thin margins in curricula vitae. CurVe does not do anything special about this because I don't think that belongs to its duty. The `geometry` package comes in handy if you want to reduce your margins.

4.2 Vertical spacing

Although they might look a bit like itemize environments, CurVe rubrics are implemented with tables. This has an important consequence: empty lines in rubrics do affect the vertical spacing of your document (at least for the time being).

You might be tempted to leave such empty lines here and there for readability, or for this precise effect it has on vertical spacing, but I advise you against this. Better to stick entries and subrubrics together, and play with the spacing commands to achieve the desired layout. This will ensure a more consistent layout with future versions of CurVe.

If you are using BibTeX, you should also be aware of the fact that some BibTeX styles output empty lines between `\bibitem`'s, and this has an unfortunate influence on vertical spacing for the same technical reason. If this vertical space annoys you, what you can do is modify the BibTeX style in order to avoid the production of these empty lines. For instance, in `plain.bst`, this simply boils down to removing the call to `newline$` at the beginning of the `output.bibitem` function.

4.3 The ltx Extension

Personally, I prefer to keep `.tex` for TEX files, and use the `ltx` extension for LATEX. This is supported by CurVe which will actually prefer `ltx` files over `tex` ones, especially when including rubrics. To be more precise, suppose you are building a flavor `flv` of your CV. A call to `\makerubric{foo}` will try to use the following files in that order:

```
foo.flv.ltx
foo.flv.tex
foo.ltx
foo.tex
```

4.4 Longtable

CurVe users should be aware of the fact that the layout implementation is based on the `LTXtable` package, which in turn is a mix of `tabularx` and `longtable`. This has several implications, most notably that when writing a rubric, you are actually inside a tabular environment. Here are some things to keep in mind:

- You are not allowed to use the \\ command to start a new line. However, you're free to use `\par` in your entries'contents instead. Note that CurVe sets `\parskip` to `0pt` so that starting a new paragraph looks like just starting a new line.
- You can use `\raggedright` and `\raggedleft` in your entries.
- You can use `\pagebreak`, `\nopagebreak` and `\newpage` at the beginning of a line, just before starting a new entry.
- Prior to version 4, `longtable` used an alignment mechanism involving calls to `\setlongtables` (see its documentation). CurVe retains this for backward compatibility and still calls `\setlongtables` in final (not draft) mode. If your version of `longtable` is recent enough, you shouldn't be concerned by this. If it is older, you might need to process your document a few times in draft mode, and then one last time in final mode. However, keep in mind that in both cases, you might still need up to 3 or 4 passes of LATEX on your document.

4.5 Managing Different Flavors

If you maintain different flavors of your CV at the same time, you probably want to rebuild all of them after any modification. Since you have a single skeleton file for all of them (say, `cv.tex`), the output file will have the same name for all flavors (say, `cv.dvi`). This can bother you if you want all flavors of your formatted CV available at the same time.

To remedy this problem, I usually use the `ask` option and a makefile to build the different flavors and move the output file to flavor-specific name. Here is a typical makefile target that should clarify (or maybe darken?) what I am saying:

```
cv.$(FLAVOR).dvi: cv.ltx $(RUBRICS)
	echo $(FLAVOR) | latex cv.ltx
	mv cv.dvi $@
```

As you can see, the shell is responsible for answering the question.

4.6 More On Flavors

In order to implement the flavor mechanism, the LaTeX macro `\input` has been redefined to look for "flavored" files first. This is actually very nice because you can use it if you want to make different flavors of text that does not belong in rubrics.

For instance, suppose you want a special version of the subtitle of your CV for the flavor `flv`. Create a file called `subtitle.flv.ltx` and put something like "`\subtitle{special subtitle}`" in it. Do something similar for the default subtitle. Now go to the skeleton of your CV, and write `\input{subtitle}` in the preamble. That's it. You'll have different subtitles in your different CV flavors.

A Appendix

A.1 Frequently Asked Questions

1. **Is there a way to align entries across several (all) rubrics?**
   Not automatically because rubrics are typeset as individual tables. There are many ways to manually "trick" too narrow keys in order to enlarge them however. As of version 1.11, CurVe provides a new convenience macro to do something similar: see section 3.2.4.

2. **How can I change the interline spacing?**
   Internally, CurVe uses LaTeX tabular-based environments. As such, you can play with `\arraystretch` to modify the space between rows.

3. **When a page break occurs in the middle of a rubric, the same alignment is kept on both pages, which might result in suboptimal layout.**
   This is a technical limitation of the automatic alignment computation process in longtables and I don't think there will be a solution anytime soon (page breaking is orthogonal to column width calculation). What you can do, once your CV is finalized, is manually split the concerned rubric into different ones, starting at the appropriate entries to avoid page breaking in the middle.

4. **How can I make multi-line subrubrics?**
   Here are two ideas:

   - Put your text in several consecutive subrubrics (one per line). However, this might not give you the desired vertical spacing.
   - Probably better, put your material in a parbox:
     `\subrubric{\parbox{width}{first blah blah\\next blah blah}}`
     This is a bit dirty because you have to figure out a suitable width for your parbox, but this will work.

5. **How can I make multi-line keys?**
   The trick is to temporarily change the key cell type to a paragraph one (remember that we're in a tabular environment).

(a) Recover the key formatting by doing something like this near the beginning of your document:
`\makeatletter\let\mykeyfont\@keyfont\makeatother`

(b) Use something like this where you need a multi-line key (you will have to adjust the paragraph width manually):

```
\entry*[\multicolumn{1}{@{}>{\mykeyfont}p{2cm}}{%
  long key\newline long key}]
  Entry text. Entry text. Entry ...
```

6. **How to deal with long keys?**
The best thing to do is to make them multi-line manually. Please refer to the previous question.

7. **Can I change the prefix locally?**
Yes and no. The `\prefix` command can only be used in the preamble or between rubrics. Otherwise, there is currently no way to change the prefix for a single entry. This limitation will disappear in a future release.

A.2 Changes

v1.16 New `\text` macro, suggested by Alexandre Duret-Lutz and Franco Callegati.
Support for the `openbib` option which was implicit before, reported by Vasilis Boucharas.
Fix incompatibilities with the `splitbib` package, reported by Jean Taucol.
Handle the `bibentry`/`hyperref` incompatibiity directly.
Implement old font commands, letting packages using them (e.g. `fancyhdr`) work correctly.

v1.15 Support for itemize environments, suggested by Mirko Hessel-von Molo
Added some documentation about vertical spacing problems in `bbl` files, suggested by Seweryn Habdank-Wojewódzki

v1.14 Support for reverse counting bibliographic entries, suggested by Joseph Wright
Support for Polish thanks to Radek Dominiak
`<radoslaw.dominiak@gmail.com>`

v1.13 Support for title alignment, suggested by Lars Kasper
Support for footnotes, suggested by Alain Coletta
Let rubrics honor the current `\linewidth`
Changed default value of `\subrubricbeforespace` to `20pt`
Fix some overfull boxes, reported by Nico Schlölmer
FAQ and documentation update

v1.12 Support for Swedish thanks to Konrad Skeri Persson
`<konrad@skeri.com>`
New customizable length `\rubricafterspace` defining the space between each rubric
Fix incompatibilities with the `multibbl` package.
Honor bibliography titles (if) provided by `multibib` or `multibbl`

New command `\today`
FAQ update
Fix implementation of `skipsamekey` option

v1.11 New FAQ section in the documentation
New command `\noentry` to manually enlarge too narrow rubrics
Make `\pagebreak`, `\nopagebreak` and `\newpage` work in rubrics, suggested by Alexandre Duret-Lutz
Fix spurious right margin spaces
Fix usage of the bib counter, disable `skipsamekey` and the prefix in bibliographic entries

v1.10 Support automatic skipping of identical keys, suggested by Akim Demaille
Fix alignment problem with empty prefix, reported by Jonas Haulin

v1.9 Fix incompatibilities with the `bibentry` package, reported by Joris Desmet
Fix standard bibliography support (broken in v1.8)

v1.8 Prevent page breaks after subrubric headings

v1.7 Support for key horizontal alignment
`\raggedleft` and `\raggedright` can now be used within individual entries
Fix typo in Danish version of `\continuedname`

v1.6 Support for rubric and subrubric titles horizontal alignment
Support for standard LaTeX page style mechanism
Support for `oneside` and `twoside` options
Support for Portuguese thanks to Adiel Mittmann `<adiel@inf.ufsc.br>`
Fix bug in `\bibliography`: protect against non existant files, reported by Andrew Comport
Fix conflict with `hyperref` in some bibliography definitions

v1.5 Support for Dutch thanks to Thomas Delaet
`<Thomas.Delaet@student.kuleuven.ac.be>`
Fix typo in rubric environment, reported by Torsten Liesk

v1.4 Support for photo inclusion
Support for headers horizontal scaling
Optional argument to `\makeheaders` for vertical alignment, suggested by Dan Luecking

v1.3 Support for Danish thanks to Kim Rud Bille `<krbi01@control.auc.dk>`

v1.2 Support for standard bibliography mechanism(s)
New macro `\entry*`
Improvements in AUC-TeX support
Support for German thanks to Harald Harders `<h.harders@tu-bs.de>`
Support for Spanish thanks to Agustín Martín `<agusmba@terra.es>`

v1.1 Support for Italian thanks to Riccardo Murri `<murri@phc.unipi.it>`

A.3 Implementation

First, the class announcement and the initial requirements:

```
⟨curve⟩\NeedsTeXFormat{LaTeX2e}
⟨*header⟩
\ProvidesClass{curve}[2010/12/14 v1.16 Curriculum Vitae class for LaTeX2e]

⟨/header⟩
⟨*curve⟩
\RequirePackage{ltxtable}
\RequirePackage{ifthen}
\RequirePackage{calc}
\RequirePackage{filehook}

```

A.3.1 Rubric Files

We don't want to output an extra `subrubricbeforespace` if no entry is present before the subrubric. This is done by using an `\@beforespace` command which is set to `0pt` at the beginning of each rubric, and switched to the proper value when an entry is added.

`\@nextentry` The `@nextentry` command is used to implement `\entry*` while maintaining backward compatibility with `\entry` and `\subrubric`. A new entry or a subrubric might have to close the preceding entry if it was opened using the starred form.

```
\gdef\@nextentry{}

```

A.3.1.1 Entries

`\@keyfont`
`\keyfont` `{`⟨*cmds*⟩`}`

```
\def\@keyfont{}
\newcommand\keyfont[1]{\gdef\@keyfont{#1}}

```

`\keyalignment` `{`⟨*hpos*⟩`}`

```
\newcolumntype{k}{>{\@keyfont}l}
\newcommand\keyalignment[1]{%
  \ifthenelse{\equal{#1}{l}}{}{%
    \ifthenelse{\equal{#1}{r}}{}{%
      \ifthenelse{\equal{#1}{c}}{}{%
        \ClassError{curve}{Invalid key alignment}{%
          You have called \protect\keyalignment\space with an invalid value.%
          \MessageBreak
          Valid options include l, c, and r.\MessageBreak
          Type X <return> to quit, fix the typo, and rerun LaTeX.}}}%
  \newcolumntype{k}{>{\@keyfont}#1}}

```

`\@prefix`

`\prefix` `{`⟨*cmds*⟩`}`

```
\def\@prefix{\textbullet}
\newcommand\prefix[1]{\gdef\@prefix{#1}}
```

`\@maybekey`
`\@alwayskey` `{`⟨*key*⟩`}`
`skipsamekey` As of version 1.10, CurVe can skip keys identical to the previous one, if the option `skipsamekey` is used.

```
\def\@maybekey#1{%
  \def\@newkey{#1}%
  \ifx\@previouskey\@newkey\gdef\@@key{}\else%
    \gdef\@@key{#1}\gdef\@previouskey{#1}%
  \fi}

\def\@alwayskey#1{%
  \gdef\@@key{#1}}

\let\@key\@alwayskey
\DeclareOption{skipsamekey}{\let\@key\@maybekey}
```

`\@entry` `[`⟨*key*⟩`]{`⟨*contents*⟩`}`
`\@sentry` `[`⟨*key*⟩`]`
`\entry`

```
\newcommand\@entry[2][]{%
  \gdef\@nextentry{}\@key{#1}%
  \egroup% end of \noalign opened in \entry.
  \@@key&\@prefix&#2\\\par}

\newcommand\@sentry[1][]{%
  \gdef\@nextentry{\\\par}\@key{#1}%
  \egroup% end of \noalign opened in \entry.
  \@@key&\@prefix&}

\newcommand\entry{%
  \@nextentry
  \noalign\bgroup\gdef\@beforespace{\subrubricbeforespace}%
  \@ifstar{\@sentry}{\@entry}}
```

`\@almosttextwidth`
`\@text` `{`⟨*contents*⟩`}`
`\@stext` This macro wraps around `\multicol` to let people input plain text within a rubric.
`\text` Currently, plain text doesn't interfere with the `skipsamekey` mechanism, but this choice is arguable. Note that the use of `\text*` is not documented yet because it doesn't work quite right. It produces an undesired space before the text and I don't understand why.

```
\newlength{\@almosttextwidth}
\AtBeginDocument{\setlength\@almosttextwidth{\textwidth-\hfuzz}}

\newcommand\@text[1]{%
  \gdef\@nextentry{}%
```

```
  \egroup% end of \noalign opened in \text.
  \multicolumn{3}{@{}p{\@almosttextwidth}@{}}{#1}\\}

\newcommand\@stext{%
  \gdef\@nextentry{\egroup\\\par}%
  \egroup% end of \noalign opened in \text.
  \multicolumn{3}{@{}p{\@almosttextwidth}@{}}\bgroup}

\newcommand\text{%
  \@nextentry
  \noalign\bgroup\gdef\@beforespace{\subrubricbeforespace}%
  \@ifstar{\@stext}{\@text}}

```

\noentry {⟨*key*⟩}

This macro is a wrapper around \kill to manually adjust too narrow rubrics.

```
\newcommand\noentry[1]{\@nextentry
  \noalign{\gdef\@nextentry{}}#1&&\kill}

```

A.3.1.2 Subrubrics

\@subrubricfont
\subrubricfont {⟨*cmds*⟩}

```
\def\@subrubricfont{\Large\itshape}
\newcommand\subrubricfont[1]{\gdef\@subrubricfont{#1}}

```

\subrubricbeforespace
\subrubricspace

```
\newlength\subrubricbeforespace
\setlength\subrubricbeforespace{20pt}

\newlength\subrubricspace
\setlength\subrubricspace{5pt}

```

\@subrubricmark
\@@subrubric {⟨*title*⟩}

Note that \@subrubricmark is called outside the raisebox. That's because otherwise, the mark would not go to the toplevel page vertical box, and TEX would not notice it.

```
\let\@subrubricmark\@gobble

\def\@@subrubric#1{%
  \rule{0bp}{\@beforespace}
  {\@subrubricfont#1}
  \@subrubricmark{#1}}

```

\@clcccolsep The normal intercolumn space between the prefix and the entry's content is replaced with an unbreakable space. This causes a problem (fixed in version 1.10) with the [sub]rubric alignments cl and cc when the prefix is empty, because the unbreakable space in question slightly shifts the entry's content to the right. If

we want a proper alignment, we then have to take this offset into account in the cl and cc multicolumns, for both rubrics and subrubrics. The following macro implements this:

```
\def\@clcccolsep{\hspace{\tabcolsep}\ifx\@prefix\@empty~\fi}

\@subrubric@*  {⟨title⟩}
  \@subrubric

\def\@subrubric@l#1{\multicolumn{3}{@{}l@{}}{\@@subrubric{#1}}}
\def\@subrubric@c#1{\multicolumn{3}{@{}c@{}}{\@@subrubric{#1}}}
\def\@subrubric@r#1{\multicolumn{3}{@{}r@{}}{\@@subrubric{#1}}}
\def\@subrubric@cl#1{&\multicolumn{2}{@{\@clcccolsep}l@{}}{\@@subrubric{#1}}}
\def\@subrubric@cc#1{&\multicolumn{2}{@{\@clcccolsep}c@{}}{\@@subrubric{#1}}}

\let\@subrubric\@subrubric@cl

\subrubricalignment  {⟨hpos⟩}
\newcommand\subrubricalignment[1]{%
  \def\@curve@temp@a{\let\@subrubric}
  \expandafter\@curve@temp@a\csname @subrubric@#1\endcsname
  \@ifundefined{@subrubric}{%
    \ClassError{curve}{Invalid subrubric alignment}{%
      You have called \protect\subrubricalignment\space with an invalid value.%
      \MessageBreak
      Valid options include l, c, r, cl and cc.\MessageBreak
      Type X <return> to quit, fix the typo, and rerun LaTeX.}}
}

\subrubric  {⟨title⟩}
\newcommand\subrubric[1]{%
  \@nextentry
  \noalign{\gdef\@nextentry{}}%
  \@subrubric{#1}\\*[\subrubricspace]\par}
```

A.3.1.3 Rubrics

```
\@rubricfont
 \rubricfont  {⟨cmds⟩}
\def\@rubricfont{\Large\bfseries}
\newcommand\rubricfont[1]{\gdef\@rubricfont{#1}}

\rubricspace
\newlength\rubricspace
\setlength\rubricspace{10pt}

\@@rubrichead  {⟨title⟩}
\@rubrichead@*  ...
```

\@rubrichead

```
\def\@@rubrichead#1{\@rubricfont#1}

\def\@rubrichead@l#1{\multicolumn{3}{@{}l@{}}{\@@rubrichead{#1}}}
\def\@rubrichead@c#1{\multicolumn{3}{@{}c@{}}{\@@rubrichead{#1}}}
\def\@rubrichead@r#1{\multicolumn{3}{@{}r@{}}{\@@rubrichead{#1}}}
\def\@rubrichead@cl#1{&\multicolumn{2}{@{\@clcccolsep}l@{}}{\@@rubrichead{#1}}}
\def\@rubrichead@cc#1{&\multicolumn{2}{@{\@clcccolsep}c@{}}{\@@rubrichead{#1}}}

\let\@rubrichead\@rubrichead@c

```

\rubricalignment {⟨*hpos*⟩}

```
\newcommand\rubricalignment[1]{%
  \def\@curve@temp@a{\let\@rubrichead} \expandafter\@curve@temp@a\csname
  @rubrichead@#1\endcsname \@ifundefined{@rubrichead}{%
    \ClassError{curve}{Invalid rubric alignment}{%
      You have called \protect\rubricalignment\space with an invalid value.%
      \MessageBreak Valid options include l, c, r, cl and cc.\MessageBreak
      Type X <return> to quit, fix the typo, and rerun LaTeX.}} }

```

\rubricafterspace

```
\newlength\rubricafterspace
\setlength\rubricafterspace{0pt}

```

\@rubricmark

rubric {⟨*title*⟩}

Marking commands don't seem to work in longtable headings. So the rubric mark is issued just after it.

As of version 1.7, `\raggedleft` and `\raggedright` are redefined in order to work within individual entries. This redefinition simply consists in removing the `\\` definition since it's not available anyway, and also to remove the `\parskip` setting since it's `0pt` in the whole class.

```
\let\@rubricmark\@gobble

\newenvironment{rubric}[1]{%
  %% \begin{rubric}
  \def\raggedright{%
    \@rightskip\@flushglue\rightskip\@rightskip\leftskip\z@skip}%
  \def\raggedleft{%
    \rightskip\z@skip\leftskip\@flushglue\parfillskip\z@skip}%
  \gdef\@beforespace{0pt}%
  \gdef\@nextentry{}%
  \gdef\@previouskey{}%
  \global\let\old@newpage\newpage%
  \global\let\old@pagebreak\pagebreak%
  \global\let\old@nopagebreak\nopagebreak
  \begin{longtable}{@{}kl@{~}X@{}}
    \@rubrichead{#1}\\*[\rubricspace]
    \endfirsthead
    \@rubrichead{#1\@continuedname}\\*[\rubricspace]
```

```
    \endhead
    \noalign{\@rubricmark{#1}%
      \global\let\in@newpage\newpage%
      \global\let\in@pagebreak\pagebreak%
      \global\let\in@nopagebreak\nopagebreak%
      \gdef\newpage{\@nextentry\noalign{\gdef\@nextentry{}}\in@newpage}
      \gdef\pagebreak{\@nextentry\noalign{\gdef\@nextentry{}}\in@pagebreak}
      \gdef\nopagebreak{\@nextentry\noalign{\gdef\@nextentry{}}\in@nopagebreak}}}{%
    %% \end{rubric}
    \@nextentry
  \end{longtable}\par\vspace\rubricafterspace
  \global\let\newpage\old@newpage%
  \global\let\pagebreak\old@pagebreak%
  \global\let\nopagebreak\old@nopagebreak}

```

\continuedname {⟨*continuation text*⟩}

```
\newcommand\continuedname[1]{\gdef\@continuedname{#1}}

```

A.3.2 Skeleton File

A.3.2.1 Utilities

\today

```
\def\today{\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or December\fi
  \space\number\day, \number\year}

```

A.3.2.2 Headers

\header@scale
\headerscale {⟨*scale*⟩}

```
\def\header@scale{.5}
\newcommand\headerscale[1]{\gdef\header@scale{#1}}
\@onlypreamble\headerscale

```

\headerspace

```
\newlength\headerspace
\setlength\headerspace{10pt}

```

\@leftheader
\leftheader {⟨*contents*⟩}
\@rightheader
\rightheader {⟨*contents*⟩}

If the user calls `\makeheaders` without specifying headers first, an error will be generated. The same applies for the title (not the subtitle), but this is already managed by LaTeX itself.

```
\def\@leftheader{%
```

```
  \ClassError{curve}{No \protect\leftheader\space given}{%
    You have called \protect\makeheaders, %
    but you didn't provide a left header.\MessageBreak
    Type X <return> to quit, add a call to \protect\lefheader\space %
    in the preamble of your CV,\MessageBreak
    and rerun LaTeX.}}
\newcommand\leftheader[1]{\gdef\@leftheader{#1}}
\@onlypreamble\leftheader

\def\@rightheader{%
  \ClassError{curve}{No \protect\rightheader\space given}{%
    You have called \protect\makeheaders, %
    but you didn't provide a right header.\MessageBreak
    Type X <return> to quit, add a call to \protect\rightheader\space %
    in the preamble of your CV,\MessageBreak
    and rerun LaTeX.}}
\newcommand\rightheader[1]{\gdef\@rightheader{#1}}
\@onlypreamble\rightheader

```

\photoscale {⟨*scale*⟩}
\photosep {⟨*length*⟩}
\photo [⟨*hpos*⟩]{⟨*file*⟩}

```
\def\photo@scale{.1}
\newcommand\photoscale[1]{\gdef\photo@scale{#1}}
\@onlypreamble\photoscale

\newlength\photosep
\setlength\photosep{10pt}

\newcommand\photo[2][l]{%
  \RequirePackage{graphicx}
  \ifthenelse{\equal{#1}{l}}{}{%
    \ifthenelse{\equal{#1}{r}}{}{%
      \ifthenelse{\equal{#1}{c}}{}{%
        \ClassError{curve}{Invalid argument to \protect\photo}{%
          Argument 2 of \protect\photo must be `l', `r' or `c'.}}}%
  \def\tmp@cmd{\global\let\makeheaders@}
  \expandafter\tmp@cmd\csname makeheaders@#1\endcsname
  \gdef\photo@file{#2}}
\@onlypreamble\photo

```

\photo@width
\includephoto@*

These different versions of the photo inclusion command exist for proper alignment of the picture itself with the left and right headers.

```
\newlength\photo@width

\def\includephoto@t{%
  \raisebox{.7\baselineskip-\height}{%
    \includegraphics[width=\photo@width]{\photo@file}}}

\def\includephoto@c{%
  \raisebox{-.5\height}{%
    \includegraphics[width=\photo@width]{\photo@file}}}
```

```
244
245 \def\includephoto@b{\includegraphics[width=\photo@width]{\photo@file}}
246
```

\leftheader@width
\rightheader@width

```
247 \newlength\leftheader@width
248 \newlength\rightheader@width
249
```

And here are the different versions of the \makeheaders command:

\makeheaders@l {⟨*vpos*⟩}

```
250 \def\makeheaders@l#1{%
251   \setlength\photo@width{\photo@scale\textwidth} \setlength\leftheader@width{%
252     (\textwidth - \photo@width - \photosep) * \real{\header@scale}}
253   \setlength\rightheader@width{%
254     \textwidth - \photo@width - \photosep - \leftheader@width}
255   \parbox[#1]{\photo@width + \photosep}{\includephoto@\hspace\photosep}%
256   \parbox[#1]{\leftheader@width}{\@leftheader}%
257   \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}}
258
```

\makeheaders@c {⟨*vpos*⟩}

```
259 \def\makeheaders@c#1{%
260   \setlength\photo@width{\photo@scale\textwidth}
261   \setlength\leftheader@width{(\textwidth - \photo@width) * \real{.5}}
262   \setlength\rightheader@width{\leftheader@width}
263   \parbox[#1]{\leftheader@width}{\@leftheader}%
264   \parbox[#1]{\photo@width}{\includephoto@}%
265   \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}}
266
```

\makeheaders@r {⟨*vpos*⟩}

```
267 \def\makeheaders@r#1{%
268   \setlength\photo@width{\photo@scale\textwidth}
269   \setlength\leftheader@width{%
270     (\textwidth - \photo@width - \photosep) * \real{\header@scale}}
271   \setlength\rightheader@width{%
272     \textwidth - \photo@width - \photosep - \leftheader@width}
273   \parbox[#1]{\leftheader@width}{\@leftheader}%
274   \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}%
275   \parbox[#1]{\photo@width + \photosep}{\hspace\photosep\includephoto@}}
276
```

\makeheaders@ {⟨*vpos*⟩}

```
277 \def\makeheaders@#1{%
278   \setlength\leftheader@width{\header@scale\textwidth}%
279   \setlength\rightheader@width{\textwidth - \leftheader@width}%
280   \parbox[#1]{\leftheader@width}{\@leftheader}%
281   \parbox[#1]{\rightheader@width}{\raggedleft\@rightheader}}
282
```

\makeheaders [⟨*vpos*⟩]

```
283 \newcommand\makeheaders[1][c]{%
```

```
284   \ifthenelse{\equal{#1}{t}}{}{%
285     \ifthenelse{\equal{#1}{b}}{}{%
286       \ifthenelse{\equal{#1}{c}}{}{%
287         \ClassError{curve}{Invalid argument to \protect\makeheaders}{%
288           Argument of \protect\makeheaders must be 't', 'b' or 'c'.}}}}%
289   \def\tmp@cmd{\global\let\includephoto@}
290   \expandafter\tmp@cmd\csname includephoto@#1\endcsname
291   \makeheaders@{#1}%
292   \par\vspace\headerspace}
293
```

A.3.2.3 Titles

\@titlefont
\titlefont {⟨*cmds*⟩}

```
294 \@onlypreamble\title
295
296 \def\@titlefont{\Huge\bfseries}
297 \newcommand\titlefont[1]{\gdef\@titlefont{#1}}
298 \@onlypreamble\titlefont
299
```

\titlespace

```
300 \newlength\titlespace
301 \setlength\titlespace{0pt}
302
```

\@subtitle
\subtitle {⟨*title*⟩}

```
303 \let\@subtitle\@undefined
304 \newcommand\subtitle[1]{\gdef\@subtitle{#1}}
305 \@onlypreamble\subtitle
306
```

\@subtitlefont
\subtitlefont {⟨*cmds*⟩}

```
307 \def\@subtitlefont{\huge\itshape}
308 \newcommand\subtitlefont[1]{\gdef\@subtitlefont{#1}}
309 \@onlypreamble\subtitlefont
310
```

\@titlealignment@*
\@title@lignment

```
311 \def\@titlealignment@l{\raggedright}
312 \def\@titlealignment@c{\centering}
313 \def\@titlealignment@r{\raggedleft}
314
315 \let\@title@lignment\@titlealignment@c
316
```

\@titlealignment {⟨*src*⟩}{⟨*hpos*⟩}

```
317 \def\@titlealignment#1#2{%
318   \def\@curve@temp@a{\let\@title@lignment}
319   \expandafter\@curve@temp@a\csname @titlealignment@#2\endcsname
```

```
320   \@ifundefined{@title@lignment}{%
321     \ClassError{curve}{Invalid title alignment}{%
322       You have called \expandafter\string\csname#1\endcsname\space%
323       with an invalid value.%
324       \MessageBreak
325       Valid options include l, c and r.\MessageBreak
326       Type X <return> to quit, fix the typo, and rerun LaTeX.}}
327 }
328
```

`\titlealignment` `{`⟨*hpos*⟩`}`

```
329 \newcommand\titlealignment[1]{\@titlealignment{titlealignment}{#1}}
330
```

`\@@maketitle` `\@maketitle` `\maketitle` `[`⟨*hpos*⟩`]`

```
331 \def\@@maketitle{%
332   \bgroup\trivlist\@title@lignment\item\relax {\@titlefont\@title}
333   \ifx\@subtitle\@undefined\else\\\@subtitlefont\@subtitle\fi
334   \endtrivlist\egroup
335   \vspace\titlespace}
336
337 \def\@maketitle[#1]{\@titlealignment{maketitle}{#1}\@@maketitle}
338
339 \newcommand\maketitle{\@ifnextchar[%]
340   \@maketitle\@@maketitle}
341
```

A.3.2.4 Rubric Inclusion

`\@flavor` `ask`

```
342 \let\@flavor\empty \newcommand\flavor[1]{\gdef\@flavor{#1}
343   \ifx\@flavor\empty\else\edef\@flavor{.\@flavor}\fi}
344
345 \DeclareOption{ask}{%
346   \typein[\@flavor]{Please specify a CV flavor (none by default):}
347   \ifx\@flavor\empty\else\edef\@flavor{.\@flavor}\fi}
348
```

`\@curveinput` `{`⟨*file*⟩`}`
`\input` `\input` is redefined in order to deal with flavors and the `ltx` extension.

```
349 \def\@curveinput#1{%
350   \IfFileExists{#1\@flavor.ltx}{\@iinput{#1\@flavor.ltx}}{%
351     \IfFileExists{#1\@flavor.tex}{\@iinput{#1\@flavor.tex}}{%
352       \IfFileExists{#1.ltx}{\@iinput{#1.ltx}}{%
353 \IfFileExists{#1.tex}{\@iinput{#1.tex}}{%
354   \@iinput{#1}}}}}}
355 \renewcommand\input{\@ifnextchar\bgroup\@curveinput\@@input}
356
```

`\makerubric` `{`⟨*name*⟩`}`

```
357 \newcommand\makerubric[1]{\LTXtable{\linewidth}{#1}}
358
```

A.3.2.5 Bibliography

In case `splitbib` is loaded, we require it to export to the sbb file. It is simpler to make CurVe compatible with it that way.

```
\PassOptionsToPackage{export}{splitbib}
```

`openbib`

```
\let\newblock\empty
\DeclareOption{openbib}{\let\newblock\par}
```

`revbib`
`bibcount`
`bibtotal`
`curve@revbib`

```
\newcounter{bibcount}
\newcounter{bibtotal}
\newif\ifcurve@revbib\curve@revbibfalse
\AtEndDocument{
  \if@filesw
    \immediate\write\@auxout{\string\setcounter{bibtotal}{\thebibcount}}%
  \fi}
\DeclareOption{revbib}{\curve@revbibtrue}
```

`\@listpubname`
`\listpubname` {⟨*title*⟩}

```
\newcommand\listpubname[1]{\gdef\@listpubname{#1}}
```

`\@curve@input@` {⟨*file*⟩}

Emulation of `\@input@` for bibliographic inclusion:

```
\def\@curve@input@#1{%
  \IfFileExists{#1}{\makerubric{#1}}{\typeout{No file #1.}}}
```

Redefine `\bibitem` and its internal implementation to behave like `\entry`.

`\@curveltx@lbibitem` [⟨*label*⟩]{⟨*ref*⟩}
`\@curve@lbibitem` ...
`\@lbibitem`

```
\let\@curveltx@lbibitem\@lbibitem
\def\@curve@lbibitem[#1]#2{\@sentry[\@biblabel{#1}]%
  \if@filesw
    {\let\protect\noexpand%
     \immediate\write\@auxout{\string\bibcite{#2}{#1}}}%
  \fi%
  \ignorespaces}
\let\@lbibitem\@curve@lbibitem
```

`\@curveltx@bibitem` {⟨*ref*⟩}
`\@curve@bibitem` ...

\@bibitem

```
\let\@curveltx@bibitem\@bibitem
\def\@curve@bibitem#1{\stepcounter{bibcount}%
  \@sentry[\@biblabel{\ifcurve@revbib\thebibtotal\else\thebibcount\fi}]%
  \ifcurve@revbib\addtocounter{bibtotal}{-1}\fi%
  \if@filesw
    \immediate\write\@auxout{\string\bibcite{#1}{\thebibcount}}%
  \fi%
  \ignorespaces}
\let\@bibitem\@curve@bibitem
```

\@curveltxbibitem
\@curvebibitem
\bibitem

```
\let\@curveltxbibitem\bibitem
\def\@curvebibitem{%
  %% #### FIXME: Dirty code duplication from \entry.
  \@nextentry
  \noalign\bgroup\gdef\@beforespace{\subrubricbeforespace}%
  \@ifnextchar[%]
  \@lbibitem\@bibitem}
\let\bibitem\@curvebibitem
```

Special cases:

- `bibentry` (version 1.9): restore standard definitions because bibentry just inlines bibliographic contents.

```
  \AtBeginOfFile{bibentry.sty}{
    \let\@lbibitem\@curveltx@lbibitem
    \let\@bibitem\@curveltx@bibitem
    \let\bibitem\@curveltxbibitem}
```

- `splitbib` (version 1.16): restore standard definitions because ours are used only as part of the `\endthebibliography` redefinition.

```
  \AtBeginOfFile{splitbib.sty}{
    \let\@lbibitem\@curveltx@lbibitem
    \let\@bibitem\@curveltx@bibitem
    \let\bibitem\@curveltxbibitem}
```

- `hyperref`: restore whatever previous behavior we had. It is arguable whether we should preserve `hyperref`'s functionality, but this answer is no, at least for the time being. Note that we assume that `hyperref` is loaded late in the process. Also, `hyperref` does not affect `\bibitem`.

```
  \AtBeginOfFile{hyperref.sty}{
    \let\@curveprevious@lbibitem\@lbibitem
    \let\@curveprevious@bibitem\@bibitem}
  \AtEndOfFile{hyperref.sty}{
    \let\@lbibitem\@curveprevious@lbibitem
    \let\@bibitem\@curveprevious@bibitem}
```

`thebibliography` {⟨*key*⟩}

`bibentry` overrides this definition in its own group in `\BR@nobib`, but this is in fact okay because it's not supposed to print anything. The same goes for `splitbib` which saves this definition and restores it when it actually matters.

```
\newenvironment{thebibliography}[1]{%
```

Disable the `skipsamekey` mechanism and the prefix which would get in the way.

```
  \let\@key\@alwayskey
  \def\@prefix{}
```

For compatibility with bibliographic packages such as `multibib`, the bibliography title is set (by order of priority) to either `\bibname`, `\refname` or `\@listpubname`.

```
  \begin{rubric}{\@ifundefined{bibname}{%
\@ifundefined{refname}{\@listpubname}{\refname}}{%
\bibname}}
  }{%
  \end{rubric}
}
```

`\@curveltxbibliography` {⟨*bibs*⟩}

`\bibliography` ...

Define `\bibliography` to issue a `\makerubric` call on the `bbl` file.

```
\let\@curveltxbibliography\bibliography
\def\bibliography#1{%
  \if@filesw
    \immediate\write\@auxout{\string\bibdata{#1}}%
  \fi%
  \@curve@input@{\jobname.bbl}}
```

Special cases:

- `multibbl` (version 1.12): merge definitions.

```
    \AtEndOfFile{multibbl.sty}{
      \def\bibliography##1##2##3{%
        \@ifundefined{##1@auxfile}{}{%
          \expandafter\immediate\write\csname ##1@auxfile\endcsname%
          {\string\bibdata{##2}}}%
        \def\bibname{##3}%
        \def\refname{##3}%
        \@curve@input@{##1.bbl}}}
```

- `splitbib`: restore original version because ours are used only as part of the `\endthebibliography` redefinition.

```
    \AtBeginOfFile{splitbib.sty}{
      \let\bibliography\@curveltxbibliography}
```

Apart from that, completing `splitbib` support is a pain in the ass. We need to do two things. First, redefine the abominable `\NMSB@writeentry` in order to output a correctly typeset rubric (titles are converted to subrubrics for instance).

Next we need to redefine \endthebibliography to restore our own definitions for \bibitem, \@bibitem and \@lbibitem, and make it use \@curve@input@ instead of \@input@.

```
\AtEndOfFile{splitbib.sty}{
  \def\NMSB@writeentry##1##2##3##4##5,{\ifx\relax ##5\relax
    \else
      \def\NMSB@currcat{##1##2##3##4}%
      \def\NMSB@currcatlevelone{##1##2}%
      \ifx\NMSB@currcatlevelone\NMSB@prevcatlevelone\else
        \expandafter\ifx\csname NMSBtitle@\NMSB@currcatlevelone
         \endcsname\relax
        \else
          \if@filesw
            \expandafter\let\expandafter\NMSB@tempentry
              \csname NMSBtitle@\NMSB@currcatlevelone\endcsname
            \edef\NMSB@tempentry{\@backslashchar subrubric{\NMSB@tempentry}}
            \expandafter\NMSB@writecatbib\expandafter{\NMSB@tempentry}
            \expandafter\ifx\csname
             NMSBcomment@\NMSB@currcatlevelone\endcsname\relax
            \else
              \immediate\write\NMSB@catbib{\string\vskip2ex^^J%
                \string\hspace{-\leftmargin}\string\relax^^J%
                \string\begin{minipage}{\textwidth}^^J%
                \string\addtolength\string\parindent{20pt}^^J%
                \string\noindent}
              \expandafter\let\expandafter\NMSB@tempentry
                \csname NMSBcomment@\NMSB@currcatlevelone\endcsname
              \expandafter\NMSB@writecatbib\expandafter{%
                \NMSB@tempentry^^J}%
              \immediate\write\NMSB@catbib{\string\end{minipage}^^J%
                \string\par\string\addpenalty{\NMSB@penalty}
                \string\vskip2ex}
            \fi
          \fi
        \fi
        \xdef\NMSB@prevcatlevelone{##1##2}%
        \ifnum\theSBresetdepth>0\relax
          \setcounter{\@listctr}{0}%
        \fi
        \expandafter\ifx\csname NMSBprefix@\NMSB@prevcatlevelone
         \endcsname\relax
          \NMSB@currprefixlevelonetok{\relax}%
        \else
          \expandafter\expandafter\expandafter\NMSB@currprefixlevelonetok
          \expandafter\expandafter\expandafter{%
             \csname NMSBprefix@\NMSB@prevcatlevelone\endcsname}%
        \fi
      \fi
      \ifx\NMSB@currcat\NMSB@prevcat\else
        \ifnum\NMSB@currcat=9999\else
          \expandafter\ifx\csname NMSBtitle@\NMSB@currcat\endcsname
           \relax
          \else
            \if@filesw
```

```
            \immediate\write\NMSB@catbib{%
              \string\par\string\addpenalty{-\NMSB@halfpenalty}%
              \string\relax^^J%
              \string\item[]%
              \string\SBsubtitle}%
            \expandafter\let\expandafter\NMSB@tempentrya
              \csname NMSBtitle@\NMSB@currcatlevelone\endcsname
            \expandafter\let\expandafter\NMSB@tempentryb
              \csname NMSBtitle@\NMSB@currcat\endcsname
            \expandafter\NMSB@writecatbib\expandafter{%
              \expandafter{\NMSB@tempentrya}}
            \expandafter\NMSB@writecatbib\expandafter{%
              \expandafter{\NMSB@tempentryb}}
            \immediate\write\NMSB@catbib{\string\relax^^J%
              \string\par\string\addpenalty{\NMSB@penalty}%
              \string\relax}%
            \expandafter\ifx\csname
             NMSBcomment@\NMSB@currcat\endcsname\relax
            \else
              \immediate\write\NMSB@catbib{\string\vskip2ex^^J%
                \string\hspace{-\leftmargin}\string\relax^^J%
                \string\begin{minipage}{\textwidth}^^J%
                \string\addtolength\string\parindent{20pt}^^J%
                \string\noindent}
              \expandafter\let\expandafter\NMSB@tempentry
                \csname NMSBcomment@\NMSB@currcat\endcsname
              \expandafter\NMSB@writecatbib\expandafter{%
                \NMSB@tempentry^^J}%
              \immediate\write\NMSB@catbib{\string\end{minipage}^^J%
                \string\par\string\addpenalty{\NMSB@penalty}%
                \string\vskip2ex}
            \fi
          \fi
        \fi
      \fi
      \xdef\NMSB@prevcat{##1##2##3##4}%
      \ifnum\theSBresetdepth>1\relax
        \setcounter{\@listctr}{0}%
      \fi
      \expandafter\ifx\csname NMSBprefix@\NMSB@currcat\endcsname\relax
        \expandafter\NMSB@currprefixtok\expandafter{%
           \the\NMSB@currprefixlevelonetok}%
      \else
        \expandafter\expandafter\expandafter\NMSB@currprefixtok
        \expandafter\expandafter\expandafter{%
          \csname NMSBprefix@\NMSB@currcat\endcsname}
      \fi
    \fi
    \expandafter\ifx\csname NMSBlabel@##5\endcsname\relax
      \if@filesw
        \stepcounter\@listctr
        \expandafter\ifx\expandafter\relax\the\NMSB@currprefixtok
          \def\NMSB@tempentry{}%
        \else
```

```
            \expandafter\expandafter\expandafter\def
            \expandafter\expandafter\expandafter\NMSB@tempentry
            \expandafter\expandafter\expandafter{%
              \expandafter\the\expandafter\NMSB@currprefixtok
                      \the\value{\@listctr}}%
            \edef\NMSB@tempentry{[\NMSB@tempentry]}
          \fi
          \edef\NMSB@tempentry{%
            \@backslashchar bibitem%
            \NMSB@tempentry{\csname NMSBkey@##5\endcsname}}%
          \expandafter\NMSB@writecatbib\expandafter{%
            \NMSB@tempentry}
          \expandafter\let\expandafter\NMSB@tempentry
            \csname NMSBentry@##5\endcsname%
          \expandafter\NMSB@writecatbib\expandafter{%
            \NMSB@tempentry}
        \fi
        \setbox\@tempboxa=\hbox{\the\NMSB@currprefixtok\the\value{\@listctr}}%
        \ifdim\NMSB@reallylongest<\wd\@tempboxa
          \setlength{\NMSB@reallylongest}{\wd\@tempboxa}%
          \xdef\NMSB@reallylongestlabel{%
            \expandafter\ifx\expandafter\relax\the\NMSB@currprefixtok
    \else\the\NMSB@currprefixtok\fi\the\value{\@listctr}}%
        \fi
      \else
        \if@filesw
          \immediate\write\NMSB@catbib{%
            \string\bibitem}
          \expandafter\let\expandafter\NMSB@tempentry
            \csname NMSBlabel@##5\endcsname
          \expandafter\NMSB@writecatbib\expandafter{%
            \expandafter[\NMSB@tempentry]}%
          \immediate\write\NMSB@catbib{%
            {\csname NMSBkey@##5\endcsname}}%
          \expandafter\let\expandafter\NMSB@tempentry
            \csname NMSBentry@##5\endcsname%
          \expandafter\NMSB@writecatbib\expandafter{%
            \NMSB@tempentry^^J^^J}
        \fi
        \setbox\@tempboxa=\hbox{\csname NMSBlabel@##5\endcsname}%
        \ifdim\NMSB@reallylongest<\wd\@tempboxa
          \setlength{\NMSB@reallylongest}{\wd\@tempboxa}%
          \expandafter\let\expandafter\NMSB@reallylongestlabel\csname
    NMSBlabel@##5\endcsname
        \fi
      \fi
    \fi
  }
  \def\endthebibliography{%
    \usecounter{enumiv}%
    \edef\NMSB@sortedvaluelist{%
      \expandafter\NMSBsortlt\expandafter{\NMSB@valuelist}}%
    \expandafter\NMSB@writelist\expandafter{\NMSB@sortedvaluelist}%
    \if@filesw
```

```
611       \immediate\write\NMSB@catbib{%
612         \string\end{thebibliography}}
613       \immediate\closeout\NMSB@catbib
614     \fi
615     \begingroup
616       \ifNMSB@newchap\else
617         \@ifundefined{chapter}{\def\section####1####2{}}{%
618           \def\chapter####1####2{}}%
619       \fi
620       \let\thebibliography\@NMSBorig@thebibliography
621       \let\endthebibliography\@NMSBorig@endthebibliography
622       \let\bibitem\@curvebibitem
623       \let\@lbibitem\@curve@lbibitem
624       \let\@bibitem\@curve@bibitem
625       \@curve@input@{\jobname.sbb}
626     \endgroup
627     \ifx\NMSB@missingcat\NMSB@empty\else
628       \expandafter\NMSB@warnnocateg\NMSB@missingcat\end
629     \fi
630     \ifNMSB@strict
631       \ifx\NMSB@doublecat\NMSB@empty\else
632         \expandafter\NMSB@warndblcateg\NMSB@doublecat\end
633       \fi
634     \fi
635     \ifdim\NMSB@reallylongest=\NMSB@longest\relax\else
636       \NMSB@warnwronglongest
637     \fi}}
638
```

A.3.3 Language Processing

english
french
francais
spanish
portuges
portugese
brazil
brazilian
italian
german
ngerman
dutch
danish
swedish
polish

```
639 \DeclareOption{english}{%
640   \continuedname{~(continued)}
641   \listpubname{List of Publications}}
642 \DeclareOption{french}{%
643   \continuedname{~(suite)}
644   \listpubname{Liste des Publications}}
645 \DeclareOption{francais}{%
646   \ExecuteOptions{french}}
647 \DeclareOption{spanish}{%
648   \continuedname{~(contin\'ua)}
649   \listpubname{Lista de Publicaciones}}
650 \DeclareOption{portuges}{%
651   \continuedname{~(continua\c c\~ao)}
652   \listpubname{Publica\c c\~oes}}
653 \DeclareOption{portuguese}{%
654   \ExecuteOptions{portuges}}
655 \DeclareOption{brazil}{%
656   \ExecuteOptions{portuges}}
657 \DeclareOption{brazilian}{%
658   \ExecuteOptions{portuges}}
659 \DeclareOption{italian}{%
660   \continuedname{~(continua)}
```

```
  \listpubname{Pubblicazioni}}
\DeclareOption{german}{%
  \continuedname{~(fortgesetzt)}
  \listpubname{Verzeichnis der Ver\"offentlichungen}}
\DeclareOption{ngerman}{%
  \ExecuteOptions{german}}
\DeclareOption{dutch}{%
  \continuedname{~(vervolg)}
  \listpubname{Publicaties}}
\DeclareOption{danish}{%
  \continuedname{~(fortsat)}
  \listpubname{Udgivelser}}
\DeclareOption{swedish}{%
 \continuedname{~(forts.)}
 \listpubname{Publikationer}}
\DeclareOption{polish}{
  \continuedname{~(kontynuacja)}
  \listpubname{Publikacje}}

```

A.3.4 Standard Class Processing

```
*paper
landscape \DeclareOption{a4paper}{
  \setlength\paperheight{297mm}
  \setlength\paperwidth{210mm}}
\DeclareOption{a5paper}{
  \setlength\paperheight{210mm}
  \setlength\paperwidth{148mm}}
\DeclareOption{b5paper}{
  \setlength\paperheight{250mm}
  \setlength\paperwidth{176mm}}
\DeclareOption{letterpaper}{
  \setlength\paperheight{11in}
  \setlength\paperwidth{8.5in}}
\DeclareOption{legalpaper}{
  \setlength\paperheight{14in}
  \setlength\paperwidth{8.5in}}
\DeclareOption{executivepaper}{
  \setlength\paperheight{10.5in}
  \setlength\paperwidth{7.25in}}
\DeclareOption{landscape}{
  \setlength\@tempdima{\paperheight}
  \setlength\paperheight{\paperwidth}
  \setlength\paperwidth{\@tempdima}}

   10pt
   11pt \DeclareOption{10pt}{\def\@ptsize{0}}
   12pt \DeclareOption{11pt}{\def\@ptsize{1}}
        \DeclareOption{12pt}{\def\@ptsize{2}}

 oneside
 twoside
```

```
\DeclareOption{oneside}{\@twosidefalse\@mparswitchfalse}
\DeclareOption{twoside}{\@twosidetrue\@mparswitchtrue}

final
draft \DeclareOption{draft}{\setlength\overfullrule{5pt}}
\DeclareOption{final}{%
  \setlength\overfullrule{0pt}
  \setlongtables}

\ExecuteOptions{english,letterpaper,10pt,oneside,final}
\ProcessOptions

\input{size1\@ptsize.clo}
\setlength\parindent{0pt}
\setlength\parskip{0pt}
\setlength\tabcolsep{10pt}
\setlength\arrayrulewidth{.4\p@}
\setlength\leftmargini{2.5em}
\leftmargin\leftmargini
\setlength\leftmarginii{2.2em}
\setlength\leftmarginiii{1.87em}
\setlength\leftmarginiv{1.7em}
\setlength\leftmarginv{1em}
\setlength\leftmarginvi{1em}
\setlength\labelsep{.5em}
\setlength\labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
\@beginparpenalty -\@lowpenalty
\@endparpenalty   -\@lowpenalty
\@itempenalty     -\@lowpenalty
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{(\theenumii)}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
\newcommand\labelitemi{\textbullet}
\newcommand\labelitemii{\normalfont\bfseries \textendash}
\newcommand\labelitemiii{\textasteriskcentered}
\newcommand\labelitemiv{\textperiodcentered}
\raggedbottom
\onecolumn
\pagestyle{empty}
\pagenumbering{arabic}

\newcommand\@makefntext[1]{\noindent\hb@xt@1em{\hss\@makefnmark}#1}
```

```
\if@twoside
  \def\ps@headings{%
    \let\@oddfoot\@empty\let\@evenfoot\@empty
    \def\@evenhead{\thepage\hfil\slshape\leftmark}%
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
    \def\@rubricmark##1{\markboth{\MakeUppercase{##1}}{}}%
    \def\@subrubricmark##1{\markright{##1}}%
  }
\else
  \def\ps@headings{%
    \let\@oddfoot\@empty
    \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
    \let\@mkboth\markboth
    \def\@rubricmark##1{\markright{\MakeUppercase{##1}}}%
  }
\fi
\def\ps@myheadings{%
  \let\@oddfoot\@empty\let\@evenfoot\@empty
  \def\@evenhead{\thepage\hfil\slshape\leftmark}%
  \def\@oddhead{{\slshape\rightmark}\hfil\thepage}%
  \let\@mkboth\@gobbletwo
  \let\@rubricmark\@gobble
  \let\@subrubricmark\@gobble
}

\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}

⟨/curve⟩
```

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.

Well, I think that's it. Enjoy using CurVe!