

The **thalie** package*

A package to typeset drama plays

Louis Paternault
spalax(at)gresille(dot)org

September 26, 2021

Abstract

This package is meant to typeset drama plays using \LaTeX . It provides commands to introduce characters' lines, to render stage direction, to divide a play into acts and scenes, to automatically build the dramatis personæ, etc.

Contents

1	Introduction	2
1.1	Other classes and packages	2
1.2	License	3
1.3	Acknowledgements	3
1.4	Examples	3
1.5	Overview	3
2	Download and Install	4
2.1	GNU/Linux Distribution	4
2.2	\LaTeX distribution	4
2.3	Manual installation	4
3	Usage	4
3.1	Localization	4
3.2	Package options	4
3.3	Sectioning	6
3.4	Dramatis personæ	8
3.5	Characters	13
3.6	Stage directions	15
3.7	Splitting verses	16

*This document corresponds to **thalie** v0.10c, dated 2021/09/26. Home page, bug requests, etc. at <http://framagit.org/spalax/thalie>

4	Localization and Internationalization	19
4.1	Localization	19
4.2	Non-latin alphabets	20
4.3	Language specific comments	20
5	Examples	21
5.1	Characters' lines	21
5.2	Headers and footers	21
6	Implementation	22
6.1	Required packages	22
6.2	Localization	22
6.3	Obsolete localization commands	22
6.4	Sectioning	23
6.5	Characters	27
6.6	Stage directions	33
6.7	Splitting verses	34
6.8	Package options	34
	Change History	36
	References	37
	Index	37

1 Introduction

This document introduces the `thalie` package, used to typeset drama plays.

1.1 Other classes and packages

This package is far from being the only one that can be used to render drama plays. If you do not like my work, you can use one of the following ones (and I guess there exists others): `drama` [6], `dramatist` [7], `play` [8], `screenplay` [9], `sides` [10].

I began to write this package in 2010, and now, at the end of 2012, I must admit that I wonder why I started this... There are already several such packages in CTAN¹, and the `dramatist` package seems really nice (I borrowed some ideas and copied some code from it). There are actually a few improvements in my package compared to `dramatist`: in my package, plays, acts and scenes appear in the table of contents; it is possible to include several plays in a single document; there are more options when building the dramatis personæ; headers and footers are taken into account, etc. But these improvements are small; it might have been smarter to contribute to `dramatist` instead of starting my own package. The

¹<http://www.ctan.org/topic/drama-script>

good part is that I have a package that fits my needs, and I learned how to write a L^AT_EX package.

Oh, yes! I know why I started this: I am a geek. . .

1.2 License

This work may be distributed and/or modified under the conditions of the L^AT_EXProject Public License, either version 1.3 of this license or (at your option) any later version.

Further information can be found in the `.dtx` file used to build this document.

In short (but this paragraph has no legal value), you can use this package freely to render your drama plays, and modify it almost freely. Nevertheless, if you like my work, you can invite me to the performance of the play you typeset using my package².

1.3 Acknowledgements

I borrowed ideas and lines of code from the L^AT_EX packages `drama`[6] and `dramatist`[7].

I used the following guides to package my package: *How to Package Your L^AT_EX Package* [4], and *L^AT_EX 2_ε for class and package writers* [5].

1.4 Examples

Most of the examples are taken from Edmond Rostand's *Cyrano de Bergerac* [2] or from William Shakespeare's *A Midsummer Night's Dream* [3]. The snippets of *Cyrano de Bergerac* are from the original (French) version of this play, roughly translated by myself (which means that they must be of poor quality).

1.5 Overview

Installation instruction are given in section 2. Documentation about how to use this package is given in section 3. In particular, section 3.3 explains how to use acts and scenes, sections 3.4 and 3.5 explains how to define characters, and use these definitions to introduce characters' lines, and section 3.6 describes commands used to render stage directions. At last, specific instruction about how to use this package with a language other than English (or a non-latin alphabet) are given in section 4. Examples are given in section 5.

Implementation is given in section 6.

²You will not take a big risk, since there is little chance I travel half the world to see a play. But if by chance I can attend it, it would make me really happy.

2 Download and Install

2.1 Gnu/Linux Distribution

If applicable, the easiest way to get `thalie` working is by installing it by your distribution package. With Debian (and Ubuntu, and surely other distributions that inherit from Debian), simply run:

```
sudo apt-get install texlive-humanities
```

2.2 L^AT_EX distribution

This package is included both in T_EXLive and MiK_TE_X. It can be installed by their respective package managers.

2.3 Manual installation

- Download the latest archive :

Stable version <http://mirrors.ctan.org/install/macros/latex/contrib/thalie.tds.zip>

Development version <https://framagit.org/spalax/thalie/repository/archive.zip?ref=main>

- Unzip the archive.
- If you got the archive from CTAN (stable version), move file `tex/latex/thalie/thalie.sty` in a L^AT_EX path.
- If you got the development version, `cd` to the main archive directory, and run `latex thalie.ins` to build `thalie.sty`. Move this file into a L^AT_EX path.

3 Usage

3.1 Localization

By default, commands `\act`, `\scene`, etc. use English words, but they are translated to the current language used by `babel` or `polyglossia`³. More information is given in section 4.1.

3.2 Package options

Package options are summed up in table 1.

Note that any package option can be redefined anywhere in the document. This makes it possible to have several plays collected in a single document, while applying different styles or options to each play. See section 3.2.4 for more detail.

³Given that someone provided the translation. Right now, the only available languages are English, French, and German.

Name	Available options	Default
<code>characterstyle</code>	<code>bold margin center simple arden imprimerie-verse imprimerie-prose</code>	<code>simple</code>
<code>playstyle</code> <code>actstyle</code> <code>scenestyle</code>	<code>center bigcenter box custom</code>	<code>box bigcenter center</code>
<code>playlevel</code> <code>actlevel</code> <code>scenelevel</code>	<code>part chapter section ...subparagraph</code>	<code>chapter section subsection</code>
<code>interludelevel</code>	<code>play act scene</code>	<code>act</code>
<code>xspace</code>	<code>true false</code>	<code>true</code>

Table 1: Package options summary

3.2.1 Space following character commands

As commands introducing characters' lines and displaying characters' names may be frequently used, it might be tempting to omit the following `{}`. For instance, one might prefer to write:

```
1 \cyrano Indeed, \cyranoname is my name!
```

instead of:

```
1 \cyrano{} Indeed, \cyranoname{} is my name!
```

If package option `xspace` is set (e.g. `xspace=true`), space is automatically added after those commands if necessary⁴; otherwise, it is not.

For historical reasons, the option `xspace` default is `true`, but this might change in some later non-backward compatible version.

3.2.2 Style

The way characters' lines are displayed, as well as play, act and scene titles, can be set when loading the options. To set character style, use option `characterstyle=<style>`. Available styles, and indication to use a custom one, are described in section 3.5.3.

Several play, act and scene title styles are defined. Choose it using options `playstyle=<style>`, `actstyle=<style>` and `scenestyle=<style>`. Description of available styles, and how to define your own one, are described in section 3.3.2.

3.2.3 Sectioning levels

If you use a table of contents, or if you also use “usual” sectioning commands (`\chapter`, `\section` and so on), the relative importance of plays, acts and scenes is important. You can set this using options `playlevel=<level>`, `actlevel=<level>`

⁴This is done with the `\xspace` command (from the `xspace` package), hence the name.

and `scenelevel=<level>`. The argument is the name of the corresponding sectioning level, i.e. one of `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph` and `subparagraph`.

Setting the interlude level is slightly different. While setting it using `interludelevel=<level>`, instead of choosing one of L^AT_EX vanilla sectioning levels as the level, you may choose `play`, `act` or `scene`. It defines if an interlude is at the same level as a play, an act or a scene.

3.2.4 Redefining options

`\setthalieoptions`

Packages options can be reset in the document body using command `\setthalieoptions{<package options>}`. Its only argument has the same syntax as the package options.

For instance, if you are writing a collections of plays, some in prose, some in verses, you might want to write something like the following.

```

1 \documentclass{book}
2 ...
3 \usepackage{thalie}
4 ...
5 \begin{document}
6 ...
7 \play{First play}
8 \setthalieoptions{characterstyle=imprimerie-verse}
9 ...
10 \play{Second play}
11 \setthalieoptions{characterstyle=impriemie-prose}
12 ...
13 \end{document}

```

3.3 Sectioning

Here begin the parts explicitly relating to drama.

3.3.1 Levels

`\play` To introduce a new play, act or scene, use commands `\play`, `\act` and `\scene`. Their
`\play*` behaviour is as close as the “usual” sectioning commands (`\chapter`, `\section` and
`\act` so on) as possible, i.e.:

- Their signature is `\play[<short title>]{<longtitle>}` (the optional short title is the one used in the table of content, and in headers and footers).
- A starred version (`\play*`, `\act*` and `\scene*`) is provided, which inserts a play (or act, or scene) which is not numbered, and does not insert any line in the table of content.
- Headers and footers are changed (more information in section 3.3.4).

Both commands `\act` and `\scene` (and their starred versions) are designed to deal with empty titles. Indeed, it is common not to give any name to acts and scenes.

By default, a play is as deep (regarding to the table of contents) as a chapter, an act as a section, and a scene as a subsection. But this can be set using package options `playlevel`, `actlevel` and `scenelevel` (see packages options, page 4). That way, you can use in your document plays, acts and scenes as well as chapters, sections and so on. It can be useful if you want a foreword, and appendix, etc.

It is not compulsory to use all three commands `\play`, `\act` and `\scene`. The rule of thumb is: if only one element exists, skip the corresponding command: if your document has a single play, you should ignore `\play`; if your document has several single act plays, set `playlevel=section,scenelevel=subsection` and ignore `\act`; etc.

`\interlude` You may want to use interludes, which are acts or scenes which are not numbered, but which should appear in the table of content. Command `\interlude[⟨short title⟩]{⟨long title⟩}` has this purpose.

You may choose the sectioning level an interlude is equivalent to in the package options. If your interludes are acts, use `interludelevel=act`; if they are scenes, use `interludelevel=scene`.

`\curtain` At last, to mark the end of an act or of the play, you can use command `\curtain`, which prints the word `\GetTranslation{Curtain}` in the middle of its own line.

3.3.2 Title styles

Several styles are available to render play, act and scene titles. Choose them using package options `playstyle`, `actstyle` and `scenestyle`. Default is `playstyle=box`, `actstyle=bigcenter`, `scenestyle=center`.

Custom titles Unfortunately, as play, act and scene titles are not considered (by L^AT_EX) as usual sections, package `titlesec`⁵ cannot be used to use alternative section titles. Here is the way to set your own one.

`\customplay` When loading the package, use `custom` as the style of the title you want to customize (e.g. `actstyle=custom`). Then, commands `\customact{⟨counter⟩}{⟨title⟩}` and `\customact*{⟨title⟩}` will be called by this package to render titles. You *must* define them. Figure 1 gives the example of the definition of the `center` style.

The first argument of `\customact` is the label of the act being printed (that is, `\theact` for an act, `\theplay` for a play, etc.), its second argument is its title. Command `\customact*` only has one argument, which is the act title.

3.3.3 Labels and counters

`\theplay` Using the same tools as `\chapter`, `\section` and so on, it is possible to define the way counters of plays, acts and scenes are displayed. You can do this by redefining `\theplay`, `\theact` and `\thescene`. For example, to have acts numbered using letters, use `\renewcommand{\theact}{\Alph{act}}`.

⁵<http://www.ctan.org/pkg/titlesec>

3.3.4 Headers and footers

`\playmark` `\actmark` `\scenemark` Once again, similar tools as those used by `\section` are provided to deal with headers and footers. When introducing, a new play, act or scene, respectively, commands `\playmark{<label>}`, `\actmark{<label>}` and `\scenemark{<label>}` are called, so that titles can be used in headers and footers. If the default behaviour does not suit you (which should be the case if you did not choose the default option for `playlevel`, `actlevel` or `scenelevel`), you can redefine them.

Examples are given in section 5.2.

3.4 Dramatis personæ

This part explains how to build and display the dramatis personæ.

Definition of characters is done in document body. As it is possible to have several plays in a single documents (for a collection of plays or sketches), it is possible to define several dramatis personæ. A new one disables the character commands defined by the previous one.

`dramatis` Definition of characters is done inside the `dramatis` environment. If the `hidden` option is given, the dramatis personæ is not printed (its only purpose is then to define the character commands).

```
\begin{dramatis} [hidden]
```

Then, several commands are available to define characters, and organize character definitions.

`\characterspace` Command `\characterspace` put some vertical space into the dramatis personæ.

3.4.1 Character definition

`\character` **Basic definition** To define a character, use command `\character`.

```
\character [cmd=command,drama=dramatis,desc=description] {<name>}
```

The mandatory argument is the name of the character, as it will appear in each of this character's line. It is later possible to redefine it using command `\setcharactername` (see part 3.5.2). Optional arguments are:

desc is a description of your character, appearing in the dramatis personæ;

cmd is the name of the command that will be used to introduce this character's lines in the remaining part of your document;

drama is the name of your character, as it will appear in the dramatis personæ. The name of the character (mandatory argument) is used as a default value.

If *cmd* is defined, this command creates two new commands: `\<cmd>` and `\<cmd>name`. The first one is used to introduce a character's line. The second one prints the character's name. An error is raised if a command with any of these two names already exists.

An example of the use of this command is given in figure 2.


```

1 \newcommand\customact [2]{
2   \begin{center}
3     \textsc{\GetTranslation{Act} #1}
4
5     #2
6   \end{center}
7 }
8 \WithSuffix\newcommand\customact*[1]{
9   \begin{center}
10    #1
11   \end{center}
12 }

```

Figure 1: Example of custom act definition

```

1 \begin{dramatis}
2   \character[cmd={cyrano}, drama={Cyrano de Bergerac}]{Cyrano}
3   \character[cmd={lebret}]{Le Bret}
4   \character[cmd={bellerose}]{Bellerose}
5 \end{dramatis}
6
7 \bigskip
8
9 \lebret[to \cyranoname, holding his arm]
10 Let's talk !
11
12 \cyrano
13 Wait for the crowd to leave. \did{To \bellerose} Can I stay?

```

Cyrano de Bergerac
Le Bret
Bellerose

LE BRET, *to Cyrano, holding his arm*: Let's talk !
CYRANO: Wait for the crowd to leave. *(To Bellerose)* Can I stay?

Figure 2: Example of character definition

		empty $\langle name \rangle$		$\langle name \rangle$	
		no $\langle cmd \rangle$	$\langle cmd \rangle$	no $\langle cmd \rangle$	$\langle cmd \rangle$
no $\langle drama \rangle$	no $\langle desc \rangle$			silent ^{iv}	default ⁱ hidden ⁱⁱⁱ
	$\langle desc \rangle$	description only ⁱⁱ		silent ^{iv}	default ⁱ
$\langle drama \rangle$	no $\langle desc \rangle$	silent ^{iv}			default ⁱ
	$\langle desc \rangle$	silent ^{iv}			default ⁱ

An empty cell means that the corresponding combination is forbidden. The superscript number refers to the list of special character definitions (page 10).

Table 2: (Not) defining arguments in character definition

Special character definition Although optional arguments are not mandatory, not defining them, or leaving the mandatory argument blank, have special meaning. The combination are summed up in table 2.

- (i) **Default definition ($\langle name \rangle$ and $\langle cmd \rangle$ are given; $\langle desc \rangle$ and $\langle drama \rangle$ may be omitted):** The character is defined as described in 3.4.1. If $\langle description \rangle$ is omitted, no description appear in the dramatis personæ; if no $\langle drama \rangle$ is given, character in the dramatis personæ has the same name as it will have in the document. Example:

```

1 \character [
2   drama={A ghost},
3   desc={the king's ghost},
4   cmd={ghost},
5   ]{The ghost}

```

- (ii) **Description only (everything omitted but $\langle desc \rangle$):** A description is inserted in the dramatis personæ. Useful to add characters such as *The kings' armies*. Example:

```

1 \character[desc={The kings' armies}]{}

```

- (iii) **Hidden character ($\langle drama \rangle$ is empty (defined, but empty), $\langle cmd \rangle$ and $\langle name \rangle$ are defined, $\langle desc \rangle$ is omitted)** Definition of a character that does not appear in the dramatis personæ. Example:

```

1 \character[drama={}, cmd={postman}]{The postman}

```

- (iv) **Silent character (one of $\langle name \rangle$ or $\langle drama \rangle$ is defined; $\langle desc \rangle$ may be omitted; $\langle cmd \rangle$ is omitted):** The character only appear in the dramatis personæ. It will not be used elsewhere in the document. An optional description may also appear in the dramatis personæ. Example:

```

1 \character{A priest}

```

`charactergroup` **Group of characters** It is possible to group several characters' definition if they have the same description. This is done with environment `charactergroup` [*width*] {*description*}. The optional argument sets the group width (on the left of the brace). See an example on figure 3.

The effect of this code will be, in the `dramatis personæ`, to have a nice brace mapping the three characters to their common description.

The behaviour of nested `charactergroup` environments is undefined⁶.

3.4.2 Customize style of `dramatis personæ`

`dramatisenv`
`\dramatischaracter`
`\dramatischaractername`
`\dramatischaracterdescription`

A default style is used to display the `dramatis personæ`, but is it made to be customized. The following commands can be redefined to make `dramatis personæ` look different. They do nothing apart from displaying text (no internal definition, etc.).

- `dramatisenv`: This environment wraps the `dramatis personæ`.
- `\dramatischaracter`{*name*}{*description*} Renders the character name and description. It should take care of cases where name or description (but not both) is missing.
- `\dramatischaractername`{*name*} Renders the character name.
- `\dramatischaracterdescription`{*description*} Renders the character description.
- Environment `charactergroup` [*width*] {*description*} Renders a group of characters.
- `\characterspace` Adds a vertical character space.

Figure 4 illustrates how those commands are used.

For example, to render character names in small caps, one would define :

```
\renewcommand{\dramatischaractername}[1]{\textsc{#1}}
```

Or to make `dramatis personæ` include a title, one can define :

```
\renewenvironment{dramatisenv}{%
  \list{}{\rightmargin 1cm\leftmargin 2cm}\item []
  \act*{Dramatis person\ae}%
}{%
  \endlist%
}
```

⁶It might work; it might not. It might work now, but stop working in the future; it might not work now, but work by accident in the future. Etc.

```

1 \begin{dramatis}
2 \begin{charactergroup}[2.5cm]{Officers}
3   \character[desc={a soldier}]{Marcellus}
4   \character[desc={a soldier}]{Bernardo}
5 \end{charactergroup}
6 \end{dramatis}

```



Figure 3: Defining a group of characters

```

1 \begin{dramatis}
2   \character{The queen}
3   \character[desc={the queen's husband.}]{The king}
4   \begin{charactergroup}{Lords and ladies}
5     \character{Lord Foo}
6     \character[desc={Lord Foo's sister.}]{Lady Bar}
7     \character[desc={Lord and Lady Baz.}]{ }
8   \end{charactergroup}
9   \characterspace
10  \character[desc={Several servants}]{ }
11 \end{dramatis}

```

To display the dramatis personæ above, the commands below are called.

```

1 \begin{dramatisenv}
2   \dramatischaracter{The queen}{}
3   \dramatischaracter{The king}{the queen's husband.}
4   \begin{charactergroup}{Lords and ladies}
5     \dramatischaracter{Lord Foo}{}
6     \dramatischaracter{Lady Bar}{Lord Foo's sister.}
7     \dramatischaracter{}{Lord and Lady Baz.}
8   \end{charactergroup}
9   \characterspace
10  \dramatischaracter{}{Several servants}
11 \end{dramatisenv}

```

And command `\dramatischaracter{<Name>}{<Description>}` is defined as something like the following code (excepted that parts taking care of cases where there is no character name or description is omitted here).

```

1 \newcommand{\dramatischaracter}[2]{ %
2   \dramatischaractername{#1}, \dramatischaracterdescription{#2}\
3 }

```

Figure 4: Commands used to display the dramatis personæ.

3.5 Characters

This part explains how to define characters, and introduce character's lines.

It has been explained that definition of a character in the *dramatis personæ* (previous subsection) also creates commands used to introduce lines of characters. For instance, defining a character using `\character[cmd=cyrano]{Cyrano}` means that *Cyrano's* lines can now be introduced using the following code.

```
1 \cyrano
2 No! It's a bit short, young man\ldots
```

This would display the name *Cyrano* using the current character style, then his line.

3.5.1 Custom characters

`\disposablecharacter` **Disposable character** One can need to define characters that are used only once (or a few times). Command `\disposablecharacter` is here to help.

```
\disposablecharacter[directions]{name}
```

This command, used in the body, introduces a line for character `name`, with optional stage directions. It is rendered the same way regular characters are rendered, but nothing is added to the *dramatis personæ*. It is used in figure 5.

This command is also useful to introduce lines of characters in non standard ways, for instance if several characters speak at once. See examples in section 5.1.

3.5.2 Changing character's name

`\setcharactername`

It is possible to change the name that appears to introduce character's lines within the text. This can be used, for example, when a character is first referred to as *A voice*, until we learn his real identity, which is *Cyrano*. An example is given in figure 5.

```
\setcharactername{command}{name}
```

This command takes two mandatory arguments. The first one, `command`, is the command used to introduce this character's line. The second one is the name to display for this character.

3.5.3 Lines

To introduce characters' lines, use the commands defined in the character definition (see the `\character` command, or the example in figure 2).

On the first scenes of *Cyrano de Bergerac*, the main character (Cyrano) is somewhere in a crowd, and only appears as *a voice*. He appears as Cyrano in the following verses.

```

1 \begin{dramatis}[hidden]
2   \character[desc={Cyrano de Bergerac}, cmd=cyrano]{The voice}
3   \character[cmd=montfleury]{Montfleury}
4 \end{dramatis}
5
6 \cyrano Leave!
7
8 \disposablecharacter{The crowd} Oh!
9
10 \montfleury[losing his voice] ‘‘Happy he who far away from court\ldots’’
11
12 \setcharactername{cyrano}{Cyrano}
13
14 \cyrano[emerging from the crowd, standing up on a chair {[}\ldots{]} with
15   a terrible nose] Ah! I am getting angry!\ldots

```

<p>THE VOICE: Leave! THE CROWD: Oh! MONTFLEURY, <i>losing his voice</i>: “Happy he who far away from court...” CYRANO, <i>emerging from the crowd, standing up on a chair [...] with a terrible nose</i>: Ah! I am getting angry!...</p>

Figure 5: Example of changing the name of a character

Choose style Several styles are available, to typeset character’s name and lines in different ways. Change the style by loading package using option `characterstyle={\style}` (available styles are `bold`, `center`, `margin`, `simple`, `imprimerie-verse`, `imprimerie-prose`). A few notes:

- Default style is `simple`.
- `imprimerie-verse` and `imprimerie-prose` styles are French *de facto* standards for typesetting drama plays, respectively in verse and in prose, as defined by the *Imprimerie nationale* [1].
- `arden` tries to mimick typesetting of the Arden Shakespeare series.
- `bold` needs a sans serif font to be defined (it should be done by default with latin alphabets, but must be done by hand with languages using non latin characters (greek, hindi, etc.)).

`\speakswithoutdirection` **Customize style** If available styles does not fit your need, you can also define your own one. To do so, you can redefine the following commands.

```
\speakswithoutdirection{\name}
\speakswithdirection{\name}{\direction}
```

The first one (`\speakswithoutdirection`) is invoked to display a character’s name to introduce its line. It takes one argument, which is the character’s line. The second one (`\speakswithdirection`) is also invoked to display a character’s name to introduce its line, but it takes a second argument, which is stage directions to be printed together with character’s name.

See also section 3.6 to see other ways to print stage directions.

3.6 Stage directions

Let us begin with a warning: since we could not figure out a nice command name made from “stage directions”, we used the French word (*didascalie*) to build up command names.

`\did` We define two ways to render stage directions: a short one, that is printed inline, and a long one, printed in its own paragraph. They act a bit like `\formula` and `\[formula\]` for formulas. The first way is the command `\did{\directions}`. The second way is the environment `dida`.

`\onstage` Some stage directions are displayed centered under the scene or act definition (I often see it in classical drama, where the list of characters appearing in each scene is displayed that way). Command `\onstage{\directions}` can be used to display such information.

`\pause` As we often need to mark pauses in theater, we created the command `\pause`, which is a shortcut for `\did{\GetTranslation{Pause}}`.

3.7 Splitting verses

When writing verses, it is sometimes useful to split a verse between two characters: the first character starts it in its line, and the second one finishes it in a second line. Visually, the start of the second line is vertically aligned to the end of the first line.



Warning: this is still experimental. I present several recipes to achieve this, but they all have their pros and cons.

`\pauseverse`
`\resumeverse`

The recipes presented below all implement macros `\pauseverse` and `\resumeverse`. Command `\pauseverse` is used at the end of the verse to be continued, while command `\resumeverse` is used at the beginning of the continuing verse. See figure 6 (page 17) for an example.

When writing a play in verse, one can use a lot of those commands, which can be tedious because of their long names. So, defining “shortcuts” of those commands can be handy, as in the following example.

```
1 \newcommand{\pv}{\pauseverse}
2 \newcommand{\rv}{\resumeverse}
```

This is not done by default, because short command names are a scarce resource in L^AT_EX, so, defining these commands by default might conflict with other packages, while most of `thalie` users will not use them.

3.7.1 Which recipe should I use?

As I have already stated, there are several ways to achieve this, but none of them is perfect.

- Do try the first recipe (section 3.7.2). If it works, stick to it. Otherwise...
- If you do not mind loading the `TikZ` package, use it (section 3.7.4). As far as I know, from a user point of view, it is the easiest recipe to use, with the less bugs. Otherwise...
- You can try building you own commands (section 3.7.3). Otherwise...
- The last recipe (section 3.7.5) always works. It is far less user-friendly, but it does the job.

3.7.2 Recipe 1 : Pure L^AT_EX



Warning: I do not understand the code behind this feature. I copied-pasted code by David Carlisle⁷ without understanding it. There are ~~probably~~ bugs⁸, but I won't be able to solve them.

The implemented macros `\pauseverse` and `\resumeverse` are written in pure L^AT_EX (without any external package). For backward compatibility reasons, this will remain the default implementation for some time...

⁷Originally written as an answer to a StackExchange question: see <https://tex.stackexchange.com/a/107727>.

⁸One on them lead to the next recipe.


```

1 \hermia
2 So is Lysander.\pauseverse
3
4 \theseus
5 \resumeverse In himself he is.

```

	HERMIA
So is Lysander.	
	THESEUS
	In himself he is.

Figure 6: Example of commands to split verse

```

1 \adjustverse{-7em}
2
3 \hermia
4 So is Lysander.\pauseverse
5
6 \theseus
7 \resumeverse In himself he is.

```

HERMIA: So is Lysander.	
THESEUS:	In himself he is.

Figure 7: Example of commands to adjust splitted verse

```

1 \hermia
2 So is Lysander.
3
4 \theseus
5 \phantom{So is Lysander.} In himself he is.

```

HERMIA: So is Lysander.	
THESEUS:	In himself he is.

Figure 8: Example of commands to split verse (using the `\phantom` command)

`\adjustverse` If the character names are displayed on the same line as the verse, the lines are not aligned the way one would expect them to. To correct this, command `\adjustverse{<length>}` can be used to add an extra (possibly negative) space when calling `\resumeverse`, to get a correct alignment. See figure 7, page 17, for an example of those three commands.

The young inexperienced fool⁹ who originally wrote those commands had not noticed that `\adjustverse` is needed because `\resumeverse` does not take into account the length of the character names, in styles where it appears on the same line as the verse. This (probably) won't be fixed, because some people might be using this, and fixing this might break their document.

If you encounter the same issue, you might prefer one of the other recipes.

3.7.3 Recipe 2 : With package hanging



Warning: This is a do-it-yourself recipe.

The bug announced at the end of the previous recipe, has been spotted and partially corrected by egreg on StackExchange¹⁰. He proposes the following code to fix this.

```

1 \usepackage{hanging}
2
3 \newlength{\brokenverse}
4 \renewcommand*{\speakswithoutdirection}[1]{%
5   \settowidth{\brokenverse}{\textsc{#1}.}%
6   \hangpara{2\parindent}{1}\noindent\textsc{#1}.%
7 }
8 \makeatletter
9 \renewcommand{\resumeverse}{%
10  \hspace{\@verseadjust}\hspace{\dimexpr\dimen\@ne-\brokenverse-2em}
11 }
12 \makeatother

```

This code ignores the chosen style (it actually redesigns it), and it only works with lines without character directions. If needed, you should also define a `\speakswithdirection` command to implement the same feature.

3.7.4 Recipe 3 : With package TikZ

As far as I know, this recipe is the best one, from a user point of view. But it loads the TikZ package, which is a huge package to implement such a tiny feature¹¹.

```

1 \usepackage{tikz}
2 \usetikzlibrary{tikzmark}
3 \usetikzlibrary{math}
4 \newcounter{thalieverse}

```

⁹Me...

¹⁰<https://tex.stackexchange.com/a/545326>

¹¹In French, we would say : “Un marteau-pilon pour écraser une mouche” (a far too heavy solution for a tiny problem).

```

5 \tikzset{tikzmark suffix=-\thethalieverse}%
6 \renewcommand{\pauseverse}{%
7   \stepcounter{thalieverse}
8   \tikzmark{thaliepauseverse}
9 }
10 \renewcommand{\resumeverse}{%
11   \tikzmark{thalieresumeverse}%
12   \iftikzmark{thaliepauseverse}{%
13     \begin{tikzpicture}[remember picture]%
14       % Uncomment to see "pause" and "resume" tikz marks.
15       %\draw[overlay,green, ultra thick] (pic cs:thaliepauseverse) -- (pic cs:tha
16       \tikzmath{
17         coordinate \c;
18         \c1 = (pic cs:thaliepauseverse);
19         \c2 = (pic cs:thalieresumeverse);
20         \width = max(0, \cx1 - \cx2);
21       }
22       % Set opacity to 1 to see the space
23       \draw[opacity=0, black, line width=1ex] (0, 0) -- (\width pt, 0);
24     \end{tikzpicture}%
25   }{}%
26 }

```

This recipe is implemented in the `examples/henry.tex` file in the repository of this project.

3.7.5 Recipe 4 : With ``

At last, if nothing else workd, this feature can also be implemented using the `\phantom` command (see figure 8 for an example, page 17). It is far less user-friendly, but it is also less error prone. Choose wisely...

4 Localization and Internationalization

4.1 Localization

Language cannot be defined directly in this package. Instead, the language currently used by `babel` or `polyglossia` is used to translate the following words (with or without capital letter, singular or plural): *play*, *act*, *scene*, *interlude*, *pause*, *curtain*.

If the option for your language does not exist¹², or if you want to change the default words used here, you can set your own translation using command `\DeclareTranslation` (see the `translations` package for more information). For instance use `\DeclareTranslation{Spanish}{Curtain}{Tel\‘on}` for Spanish¹³.

See section 4.3 for the list of supported languages.

¹²You can also send me the translation for your language, to improve this package.

¹³I may be wrong: I cannot speak Spanish.

4.1.1 Common translations

`\playname`
`\actname` Commands defined here are shorter versions of strings commonly translated.
`\scenename` For instance, `\playname` is a shortcut for `\GetTranslation{Play}`, `\actname` is a
`\interludename` shortcut for `\GetTranslation{Act}`, and so on.
`\pausename`
`\curtainname`

4.2 Non-latin alphabets

This package should work with languages using non-latin alphabets (greek, hindi, etc.). However, here are a few remarks.

- The `bold` character style (see section 3.5.3) can only be used if a sans serif font has been defined. For instance, for hindi, you might write in the preamble: `\newfontfamily\hindifontsf{Noto Sans Devanagari}`.

4.3 Language specific comments

The following languages are available:

- English,
- French,
- German.

4.3.1 English

- The `arden` character style tries to mimick the typesetting of the Arden Shakespeare series.

4.3.2 French

- The `imprimerie-verse` and `imprimerie-prose` character styles implements the French *de facto* standards for typesetting drama plays, respectively in verse and in prose, as defined by the *Imprimerie nationale* [1].
- The same book ([1]) states that act numbers should be typeset in capital roman numerals (which is done by default with `thalie`), and that scene numbers should be typeset in small capital roman numerals, which is *not* the default, but can be achieved by adding the following line in the preamble:
`\renewcommand{\thescene}{\textsc{\roman{scene}}}`.

4.3.3 Other languages

I do not know anything related to other languages. If you have interesting advice for those languages that you would like to share (preferably with a reference to a reliable source), I would be happy to add them to this list!

5 Examples

5.1 Characters' lines

5.1.1 Introduce lines of several characters at the same time.

Since `\disposablecharacter` can display arbitrary text using the style of character names, it can be used to render a custom text when two characters are speaking together.

```
1 \bob
2 Let's open it.
3
4 \disposablecharacter{\alicename and \evename}
5 No!
```

BOB: Let's open it. ALICE AND EVE: No!

5.1.2 Add a footnote on a character name

Adding a footnote on the character name introducing its line cannot be done on the character command defined by the `dramatis personæ`. Instead, the `\disposablecharacter` command can be used instead.

```
1 \disposablecharacter{\alicename\footnote{An interesting note.}}
2 Hello, world!
```

5.1.3 Add a footnote in a stage direction

Footnote must be protected to be included in stage directions.

```
1 \cyrano[clapping his hands\protect\footnote{An interesting footnote.}]
2 One!
```

5.2 Headers and footers

By default, with the `article` class, the header of the left page contains the play name, and the header of the right page contains the act number.

5.2.1 Act number on left, scene number on right

```
1 \renewcommand{\actmark}[1]{%
2   {\markboth{\MakeUppercase{\GetTranslation{Act}}\ theact\ #1}}}%
3 }
4 \renewcommand{\scenemark}[1]{%
5   {\markright{\MakeUppercase{\GetTranslation{Scene}}\ thescene\ #1}}%
6 }
```

5.2.2 Play name of left, act and scene name on right

```
1 \usepackage{fancyhdr}
2 \pagestyle{fancy}
3 \fancyhf{}
4 \fancyhead[R0]{%
5   \textit{\MakeUppercase{%
6     \GetTranslation{Act} \theact{} , %
7     \GetTranslation{Scene} \thescene%
8   }}%
9 }
10 \fancyhead[LE]{\leftmark}
```

6 Implementation

6.1 Required packages

Loading some packages.

```
1 % Options
2 \RequirePackage{pgfkeys}
3 % Chapters, sections, etc
4 \RequirePackage{etoolbox}
5 \RequirePackage{suffix}
6 % Dramatis person\ae{} is written using a table.
7 \RequirePackage{tabularx}
8 % Handling spaces after having introduced characters' lines.
9 \RequirePackage{xspace}
```

6.2 Localization

Use translations. Translations are defined in the dicts/*thalie*-<LANGUAGE>.trs1 files.

```
10 \RequirePackage{translations}
11 \LoadDictionary{thalie}
12 \LoadDictionaryFor{fallback}{thalie}
```

6.3 Obsolete localization commands

<code>\playname</code>	Shorter versions of commonly translated strings.
<code>\actname</code>	13 <code>\newcommand{\playname}{\GetTranslation{Play}}</code>
<code>\scenename</code>	14 <code>\newcommand{\actname}{\GetTranslation{Act}}</code>
<code>\interludename</code>	15 <code>\newcommand{\scenename}{\GetTranslation{Scene}}</code>
<code>\curtainname</code>	16 <code>\newcommand{\interludename}{\GetTranslation{Interlude}}</code>
<code>\pausename</code>	17 <code>\newcommand{\curtainname}{\GetTranslation{Curtain}}</code>
	18 <code>\newcommand{\pausename}{\GetTranslation{Pause}}</code>

6.4 Sectioning

6.4.1 Headears, footers, counters, etc.

`playmark` Define commands `\playmark`, `\actmark` and `\scenemark`, which are involved in headers and footers definition.

`actmark`

`scenemark`

```
19 \newcommand{\playmark}[1]{%
20   \markboth{\MakeUppercase{#1}}{}}%
21 }
22 \newcommand{\actmark}[1]{%
23   \markright{\MakeUppercase{%
24     \GetTranslation{Act}\ theact%
25     \ifdefempty{#1}{}{: #1}%
26   }}%
27 }
28 \newcommand{\scenemark}[1]{%
29 }
```

`theplay` Defines counters for plays, acts and scenes, and the associated labels (`\theplay`, `\theact`, `\thescene`).

`theact`

`thescene`

```
30 \newcounter{play}
31 \renewcommand{\theplay}{\arabic{play}}
32 \newcounter{act}[play]
33 \renewcommand{\theact}{\Roman{act}}
34 \newcounter{scene}[act]
35 \renewcommand{\thescene}{\arabic{scene}}
```

6.4.2 Styles

Style definition. Command `\@displaytitle` is later used by commands `\play`, `\act` and `\scene` (and their starred version) to typeset the title.

```
36 \newcommand{\@displaytitle}[3]{
37   % Arguments:
38   % - Style
39   % - Label (none = not in toc)
40   % - Title
41   \ifdefstring{#1}{center}{
42     \begin{center}
43       \textsc{#2}
44
45       #3
46     \end{center}
47 }{\ifdefstring{#1}{bigcenter}{
48   \begin{center}
49     \Large
50     \textsc{#2}
51
52     #3
53   \end{center}}
```

```

54 }{\ifdefstring{#1}{box}{
55   \begin{center}
56     \framebox{\begin{minipage}{0.7\textwidth}
57       \begin{center}
58         \Large \bfseries
59         \vspace{0.5em}
60
61         #2
62         \ifboolexpr{test{\ifstrempy{#3}} or test{\ifstrempy{#2}}}{}{---}
63         #3
64
65         \vspace{0.5em}
66       \end{center}
67     \end{minipage}}
68   \end{center}
69   \vspace{1em}
70 }{}}
71 }

```

6.4.3 Sectioning commands

Some general commands to handle clearing pages, and table of contents.

```

72 \newcommand\@clearpage[1]{%
73   % Clear page if necessary
74   \ifboolexpr{test{\ifdefstring{#1}{part}} or test{\ifdefstring{#1}{chapter}}}{
75     \cleardoublepage
76     \thispagestyle{empty}
77   }{ }
78 }
79

```

`\play` Define sectioning commands to introduce plays. As for `\section` (and other) sectioning command, the starred version does the same, excepted that nothing is written in the table of content.

```

80 \newcommand{\play}[2] []{%
81   \refstepcounter{play}
82   \ifstrempy{#1}{
83     \def\@short{#2}
84   }{
85     \def\@short{#1}
86   }
87   \@clearpage{\@playlevel}
88   \playmark{\@short}
89   \addcontentsline{toc}{\@playlevel}{\@short}
90   \ifdefstring{\@playstyle}{custom}{
91     \customplay{\theplay}{#2}
92   }{
93     \@displaytitle{\@playstyle}{#2}
94   }

```



```

95 }
96 \WithSuffix\newcommand\play*[1]{%
97   \clearpage{\@playlevel}
98   \ifdefstring{\@playstyle}{custom}{
99     \customplay*{#1}
100  }{
101    \@displaytitle{\@playstyle}{}{#1}
102  }
103 }

\act Define commands to introduce acts.
\act* 104 \newcommand{\act}[2] [] {%
105   \refstepcounter{act}
106   \ifstrempy{#1}{
107     \def\@short{#2}
108   }{
109     \def\@short{#1}
110   }
111   \ifdefempty{\@short}{
112     \def\@label{\GetTranslation{Act} \theact{}}
113   }{
114     \def\@label{\GetTranslation{Act} \theact{}}\xspace: }
115   }
116   \clearpage{\@actlevel}
117   \actmark{\@short}
118   \addcontentsline{toc}{\@actlevel}{\@label\@short}
119   \ifdefstring{\@actstyle}{custom}{
120     \customact{\theact}{#2}
121   }{
122     \@displaytitle{\@actstyle}{\GetTranslation{Act} \theact}{#2}
123   }
124 }
125 \WithSuffix\newcommand\act*[1]{%
126   \clearpage{\@actlevel}
127   \ifdefstring{\@actstyle}{custom}{
128     \customact*{#1}
129   }{
130     \@displaytitle{\@actstyle}{}{#1}
131   }
132 }

\scene Define commands to introduce scenes.
\scene* 133 \newcommand{\scene}[2] [] {%
134   \refstepcounter{scene}
135   \ifstrempy{#1}{
136     \def\@short{#2}
137   }{
138     \def\@short{#1}
139   }
140   \ifdefempty{\@short}{

```

```

141   \def\@label{\GetTranslation{Scene} \thescene{}}
142   }{
143   \def\@label{\GetTranslation{Scene} \thescene{}}\xspace: }
144   }
145   \@clearpage{\@scenelevel}
146   \scenemark{\@short}
147   \addcontentsline{toc}{\@scenelevel}{\@label\@short}
148   \ifdefstring{\@scenestyle}{custom}{
149     \customscene{\thescene}{#2}
150   }{
151     \displaytitle{\@scenestyle}{\GetTranslation{Scene} \thescene}{#2}
152   }
153 }
154 \WithSuffix\newcommand\scene*[1]{%
155   \@clearpage{\@scenelevel}
156   \ifdefstring{\@scenestyle}{custom}{
157     \customscene*{#1}
158   }{
159     \displaytitle{\@scenestyle}{}{#1}
160   }
161 }

```

`\interlude` Define commands to introduce interludes.

```

\interlude* 162 \newcommand{\interlude}[2] [] {%
163   \ifstrempy{#1}{
164     \def\@short{#2}
165   }{
166     \def\@short{#1}
167   }
168   \ifdefempty{\@short}{
169     \def\@label{\GetTranslation{Interlude}}
170   }{
171     \def\@label{\GetTranslation{Interlude}}\xspace: }
172   }
173   \ifdefstring{\@interludelevel}{play}{
174     \@clearpage{\@playlevel}
175     \playmark{\@short}
176     \addcontentsline{toc}{\@playlevel}{\@label\@short}
177     \displaytitle{\@playstyle}{\GetTranslation{Interlude}}{#2}
178   }{\ifdefstring{\@interludelevel}{act}{
179     \@clearpage{\@actlevel}
180     \actmark{\@short}
181     \addcontentsline{toc}{\@actlevel}{\@label\@short}
182     \displaytitle{\@actstyle}{\GetTranslation{Interlude}}{#2}
183   }{\% \@interludelevel is scene
184     \@clearpage{\@scenelevel}
185     \scenemark{\@short}
186     \addcontentsline{toc}{\@scenelevel}{\@label\@short}
187     \displaytitle{\@scenestyle}{\GetTranslation{Interlude}}{#2}
188   }}

```

```

189 }
190 \WithSuffix\newcommand\interlude*[1]{%
191   \ifdefstring{\@interludelevel}{play}{
192     \clearpage{\@playlevel}
193     \displaytitle{\@playstyle}{\GetTranslation{Interlude}}{#1}
194   }{\ifdefstring{\@interludelevel}{act}{
195     \clearpage{\@actlevel}
196     \displaytitle{\@actstyle}{\GetTranslation{Interlude}}{#1}
197   }{% \@interludelevel is scene
198     \clearpage{\@scenelevel}
199     \displaytitle{\@scenestyle}{\GetTranslation{Interlude}}{#1}
200   }}
201 }

```

6.4.4 Curtain

`\curtain` Used to mark the end of an act. Prints “curtain” centered in its own line.

```

202 % Curtain
203 \newcommand\curtain{
204   \begin{center}
205     \Large\textsc{\GetTranslation{Curtain}}
206   \end{center}
207 }

```

6.5 Characters

6.5.1 Spacing

Add (or not) an `\backslash$xspace` command at the end of character names, depending on the `xspace` package option.

```

208 \newcommand{\@maybexspace}{%
209   \ifxspace%
210     \xspace%
211   \fi%
212 }

```

6.5.2 Characters’ line

Define the command `\@speaks`, which introduces a character’s line. It is not meant to be used by user, but will be used by further commands. It takes two arguments: the character’s name, and optionally, a stage direction.

```

213 \newcommand{\@speaks}[2] []{%
214   \ifstrempy{#1}{%
215     \speakswithoutdirection{#2}%
216   }{%
217     \speakswithdirection{#2}{#1}%
218   } \@maybexspace%
219 }

```

`\speakswithdirection` Definition of styles for introducing characters' lines. These commands can be
`\speakswithoutdirection` overloaded by user.

```

220 \newenvironment{@smallcenter}
221   {\par\smallskip\centering}
222   {\par\nopagebreak\ignorespacesafterend}
223 \providecommand\speakswithdirection{}
224 \providecommand\speakswithoutdirection{}
225 \newcommand{\@setcharacterstyle}[1]{
226   \ifstrequal{#1}{bold}{%
227     % Bold style
228     \renewcommand\speakswithdirection[2]{%
229       \noindent%
230       {\bfseries\sffamily ##1} \emph{##2}\xspace:%
231     }
232     \renewcommand\speakswithoutdirection[1]{%
233       \noindent%
234       {\bfseries\sffamily ##1\xspace:%
235     }%
236   }{}%
237   \ifstrequal{#1}{center}{%
238     % Center style
239     \renewcommand\speakswithdirection[2]{%
240       \begin{center}%
241         \textsc{##1},\emph{##2}%
242       \end{center}%
243       \par\ignorespacesafterend%
244     }%
245     \renewcommand\speakswithoutdirection[1]{%
246       \begin{center}%
247         \textsc{##1}%
248       \end{center}%
249       \par\ignorespacesafterend%
250     }%
251   }{}%
252   \ifstrequal{#1}{imprimerie-verse}{%
253     % Style for verse plays defined by the French Imprimerie nationale
254     \renewcommand\speakswithdirection[2]{%
255       \begin{@smallcenter}%
256         \textsc{##1}, \emph{##2}%
257       \end{@smallcenter}%
258     }%
259     \renewcommand\speakswithoutdirection[1]{%
260       \begin{@smallcenter}%
261         \textsc{##1}%
262       \end{@smallcenter}%
263     }%
264   }{}%
265   \ifstrequal{#1}{imprimerie-prose}{%
266     % Style for prose plays defined by the French Imprimerie nationale

```

```

267 \renewcommand\speakswithdirection[2]{%
268 \noindent\hspace*{-\parindent}\textsc{##1}, \emph{##2}\xspace:%
269 }%
270 \renewcommand\speakswithoutdirection[1]{%
271 \noindent\hspace*{-\parindent}\textsc{##1}\xspace:%
272 }%
273 }{}%
274 \ifstrequal{#1}{arden}{%
275 \renewcommand\speakswithdirection[2]{%
276 \noindent\hspace*{-\parindent}\textsc{\MakeLowercase{##1}} [\emph{##2}]\quad%
277 }%
278 \renewcommand\speakswithoutdirection[1]{%
279 \noindent\hspace*{-\parindent}\textsc{\MakeLowercase{##1}}\quad%
280 }%
281 }{}%
282 \ifstrequal{#1}{simple}{%
283 % Simple style
284 \renewcommand\speakswithdirection[2]{%
285 \indent\textsc{##1}, \emph{##2}\xspace:%
286 }%
287 \renewcommand\speakswithoutdirection[1]{%
288 \indent\textsc{##1}\xspace:%
289 }%
290 }{}%
291 \ifstrequal{#1}{margin}{%
292 % Margin style
293 \setlength{\leftskip}{3cm}
294 \renewcommand\speakswithdirection[2]{%
295 \hspace{-3cm} ##1 ##2
296 }
297 \renewcommand\speakswithoutdirection[1]{%
298 \hspace{-3cm} ##1
299 }%
300 }{}%
301 }

```

6.5.3 Dramatis personæ

`dramatis` In this environment are defined characters.

```

302 \newcommand{\@dramatis@clear}{}
303 \newcommand{\@empty@}{}
304 \providebool{@dramatis@hidden}
305 \newenvironment{dramatis}[1][]{
306 \@dramatis@clear{}
307 \undef{\@dramatis@clear}
308 \ifstrequal{#1}{hidden}{
309 \booltrue{@dramatis@hidden}
310 }{\ifstreempty{#1}{
311 \boolfalse{@dramatis@hidden}

```

```

312   \dramatisenv
313   }{
314   \ClassError{thalie}{%
315     Environment dramatis does not accept option #1.%
316   }{
317   }
318   }
319   }
320 }{%
321   \notbool{@dramatis@hidden}{%
322     \enddramatisenv
323   }{%
324   }

```

dramatisenv Customizable commands to build dramatis personæ.

\dramatischaracter 325 % Environment used to *display* dramatis personæ.

\dramatischaractername 326 \newenvironment{dramatisenv}{%

\dramatischaracterdescription 327 \list{}{\rightmargin1cm\leftmargin2cm}\item[]

\characterspace 328 }{%

charactergroup 329 \endlist%

330 }

331

332 % Display character name

333 \newcommand{\dramatischaractername}[1]{\textbf{#1}}

334

335 % Display character description

336 \newcommand{\dramatischaracterdescription}[1]{#1}

337

338 % Adds a vertical space

339 \newcommand{\characterspace}{ %

340 \notbool{@dramatis@hidden}{%

341 \smallskip\newline %

342 }{ %

343 }

344

345 % Display dramatis personæ name and description

346 \newcommand{\dramatischaracter}[2]{ %

347 \hspace*{-1cm} %

348 \ifboolexpr{(not test {\ifdefempty{#1}}) and test {\ifdefempty{#2}}}{%

349 \dramatischaractername{#1} %

350 }{%

351 \ifboolexpr{ test{\ifdefempty{#1}} and not test{\ifdefempty{#2}}}{%

352 \dramatischaracterdescription{#2} %

353 }{%

354 \ifboolexpr{(not test{\ifdefempty{#1}}) and (not test{\ifdefempty{#2}})}{%

355 \dramatischaractername{#1}, \dramatischaracterdescription{#2} %

356 }{%

357 \newline %

358 }

359

```

360 % Environment to manage groups of characters
361 \newlength{\@spaceaftergroup}
362 \newenvironment{charactergroup}[2][5cm]{%
363 \notbool{@dramatis@hidden}{%
364 \gdef\@groupname{#2} %
365 \gdef\@grouplength{#1} %
366 \hspace*{-1.3pt}\math\left. %
367 \minipage[c]{#1} %
368 \vspace*{2pt} %
369 }{}%
370 }{}%
371 \notbool{@dramatis@hidden}{%
372 \vspace*{-8pt} %
373 \endminipage %
374 \right\} \endmath %
375 %
376 \setlength{\@spaceaftergroup}{\linewidth}
377 \addtolength\@spaceaftergroup{-\@grouplength}
378 \addtolength\@spaceaftergroup{-20pt}
379 \begin{minipage}[c]{\@spaceaftergroup}
380 \@groupname %
381 \end{minipage}
382 \newline %
383 }{}%
384 }

```

6.5.4 Character definitions

`\setcharactername` Set (or change) the name used to introduce the lines of a character.

```

385 % The tough part of this code is to deal with optional argument.
386 \newcommand{\setcharactername}[2]{%
387 \expandafter\gdef\csname#1name\endcsname{%
388 #2\@maybexspace%
389 }%
390 \expandafter\gdef\csname#1\endcsname{%
391 \@ifnextchar[{%
392 \defcharcommand@with{#2}%
393 }{%
394 \defcharcommand@without{#2}%
395 }%
396 }%
397 \xappto{\@dramatis@clear}{%
398 \global\noexpand\csundef{#1}%
399 \global\noexpand\csundef{#1name}%
400 }%
401 }

```

Define the command used to introduce a character's line. It takes two arguments: the first one is the name of the command to define, and the second one is

the name of the character corresponding to this command.

```
402 \newcommand{\@definecharactercommand}[2]{%
403   \ifcsdef{#1}{%
404     \ClassError{thalie}{%
405       A command named \@backslashchar#1 already exists. We cannot define a new
406       one.%
407     }{%
408       Choose another command name to introduce character #2's lines.%
409     }%
410   }{%
411   }%
412   \ifcsdef{#1name}{%
413     \ClassError{thalie}{%
414       A command named \@backslashchar#1name already exists. We cannot define a
415       new one.%
416     }{%
417       Choose another command name to introduce character #2's lines, such that
418       when a new command is defined by adding "name" to it, it does not
419       conflict with an existing one.
420     }%
421   }{%
422   }%
423   \setcharactername{#1}{#2}%
424 }
425 \def\defcharcommand@with#1[#2]{\@speaks[#2]{#1}}
426 \def\defcharcommand@without#1{\@speaks{#1}}
```

`\character` Define a character: put it in the dramatis personæ, and define corresponding commands.

```
427 \pgfkeys{
428   % Character definition
429   /THALIE/CHARACTER/.is family,
430   /THALIE/CHARACTER,
431   cmd/.value required,
432   cmd/.store in=\@cmd,
433   drama/.value required,
434   drama/.store in=\@drama,
435   desc/.value required,
436   desc/.store in=\@desc,
437 }
438 \newcommand{\character}[2][]{
439   \undef{\@drama}
440   \undef{\@cmd}
441   \undef{\@desc}
442   \pgfkeys{/THALIE/CHARACTER, #1}%
443   % Forbidden combinations
444   \ifboolexpr{
445     ( test{\ifdef{\@cmd}} and test{\ifstrempy{#2}} ) or
446     ( test{\ifstrempy{#2}} and test{\ifundef{\@cmd}} and test{\ifundef{\@desc}} and test{\ifun
```



```

447   ( test{\ifundef{\@cmd}} and (not test{\ifstrempy{#2}}) and test{\ifdef{\@drama}} )
448   ){
449   \ClassError{thalie}{Invalid character definition.}{All combination of omitted arguments are
450   }{ }
451   % Defining character command
452   \ifboolexpr{ test{\ifdef{\@cmd}} and (not test{\ifstrempy{#2}}) }{%
453     \@definecharactercommand{\@cmd}{#2}
454   }{%
455     \notbool{@dramatis@hidden}{%
456       \ifboolexpr{ (not test{\ifstrempy{#2}}) and test{\ifdef{\@cmd}} and test{\ifdef{\@drama}}
457         \ifdefempty{\@drama}{%
458           % Hidden character. Nothing added to dramatis personae
459         }{%
460           % Populating dramatis personae
461           \dramatischaracter{\@drama}{\@empty@}
462         }%
463       }{%
464         % Populating dramatis personae
465         \ifcsundef{@desc}{\gdef\@desc{}}{ }%
466         \ifcsundef{@drama}{\gdef\@drama{#2}}{ }%
467         \dramatischaracter{\@drama}{\@desc}
468       }%
469     }{ }%
470 }

```

`\disposablecharacter` Disposable character (character used only a few times, defined on-the-fly, that does not appear in the dramatis personæ).

```

471 \newcommand{\disposablecharacter}[2] [] {}%
472 \@speaks[#1]{#2}%
473 }

```

6.6 Stage directions

`\onstage` Centered stage direction.

```

474 \newcommand{\onstage}[1]{{\centering \emph{#1}\par\medskip}}

```

`\did` Inline stage directions.

```

475 \newcommand{\did}[1]{\emph{( #1 ) } }

```

`dida` Bigger stage directions, in its own paragraph.

```

476 \newenvironment{dida}{%
477   \begin{quote}
478   \begin{em}
479   }{%
480   \end{em}
481   \end{quote}
482 }

```

`\pause` Shortcut for `\did{\GetTranslation{Pause}}`.

```

483 \newcommand\pause{\did{\GetTranslation{Pause}}}

```

6.7 Splitting verses

```
\pauseverse  Commands to split a verse between several characters.
\resumeverse 484 % Thanks to Timothy Li for his question, and David Carlisle for his answer:
\adjustverse 485 % https://tex.stackexchange.com/questions/107726/107727#107727
486
487 \newlength{\@verseadjust}
488 \setlength{\@verseadjust}{0pt}
489
490 \newcommand{\adjustverse}[1]{\setlength{\@verseadjust}{#1}}
491
492 \newcommand{\pauseverse}{\aboveplayskip\z@\aboveplayskip\z@
493   \belowplayskip\z@\belowplayskip\z@
494   $$$\global\dimen\@ne\preplaysize
495   \xdef\tmp{%
496     \preplaysize\the\preplaysize
497     \prevgraf\the\prevgraf\relax}%
498   $$$\vskip\dimexpr-\parskip-\baselineskip\relax}\tmp
499 }
500
501 \newcommand{\resumeverse}{%
502   \hspace{\@verseadjust}\hspace{\the\dimen\@ne}
503 }
```

6.8 Package options

Manage package options.

```
504 \RequirePackage{pgfopts}
505
506 \pgfkeys{
507   % Character style
508   /THALIE/.cd,
509   characterstyle/.value required,
510   characterstyle/.default=simple,
511   characterstyle/.is choice,
512   characterstyle/bold/.code=\@setcharacterstyle{bold},
513   characterstyle/center/.code=\@setcharacterstyle{center},
514   characterstyle/margin/.code=\@setcharacterstyle{margin},
515   characterstyle/simple/.code=\@setcharacterstyle{simple},
516   characterstyle/arden/.code=\@setcharacterstyle{arden},
517   characterstyle/imprimerie-verse/.code=\@setcharacterstyle{imprimerie-verse},
518   characterstyle/imprimerie-prose/.code=\@setcharacterstyle{imprimerie-prose},
519   characterstyle,
520 }
521
522 \pgfkeys{
523   % play style
524   /THALIE/.cd,
525   playstyle/.value required,
```

```

526 playstyle/.default=box,
527 playstyle/.is choice,
528 playstyle/center/.code=\def\@playstyle{center},
529 playstyle/bigcenter/.code=\def\@playstyle{bigcenter},
530 playstyle/box/.code=\def\@playstyle{box},
531 playstyle/custom/.code=\def\@playstyle{custom},
532 playstyle,
533 }
534 \pgfkeys{
535   % act style
536   /THALIE/.cd,
537   actstyle/.value required,
538   actstyle/.default=bigcenter,
539   actstyle/.is choice,
540   actstyle/center/.code=\def\@actstyle{center},
541   actstyle/bigcenter/.code=\def\@actstyle{bigcenter},
542   actstyle/box/.code=\def\@actstyle{box},
543   actstyle/custom/.code=\def\@actstyle{custom},
544   actstyle,
545 }
546 \pgfkeys{
547   % scene style
548   /THALIE/.cd,
549   scenestyle/.value required,
550   scenestyle/.default=center,
551   scenestyle/.is choice,
552   scenestyle/center/.code=\def\@scenestyle{center},
553   scenestyle/bigcenter/.code=\def\@scenestyle{bigcenter},
554   scenestyle/box/.code=\def\@scenestyle{box},
555   scenestyle/custom/.code=\def\@scenestyle{custom},
556   scenestyle,
557 }
558
559 \pgfkeys{
560   % play level
561   /THALIE/.cd,
562   playlevel/.value required,
563   playlevel/.default=chapter,
564   playlevel/.store in=\@playlevel,
565   playlevel,
566 }
567 \pgfkeys{
568   % act level
569   /THALIE/.cd,
570   actlevel/.value required,
571   actlevel/.default=section,
572   actlevel/.store in=\@actlevel,
573   actlevel,
574 }
575 \pgfkeys{

```

```

576 % scene level
577 /THALIE/.cd,
578 scenelevel/.value required,
579 scenelevel/.default=subsection,
580 scenelevel/.store in=\@scenelevel,
581 scenelevel,
582 }
583 \pgfkeys{
584 % interlude level
585 /THALIE/.cd,
586 interludelevel/.value required,
587 interludelevel/.default=act,
588 interludelevel/.is choice,
589 interludelevel/play/.code=\def\@interludelevel{play},
590 interludelevel/act/.code=\def\@interludelevel{act},
591 interludelevel/scene/.code=\def\@interludelevel{scene},
592 interludelevel,
593 }
594 \newif\if@xspace
595 \pgfkeys{
596 % xspace option
597 /THALIE/.cd,
598 xspace/.value required,
599 xspace/.is if=@xspace,
600 xspace/.default=true,
601 xspace,
602 }
603
604 \ProcessPgfPackageOptions{/THALIE}

```

```

\setthalieoptions Redefine package options.
605 \newcommand{\setthalieoptions}[1]{%
606 \pgfkeys{/THALIE/.cd, #1}%
607 }

```

Change History

v0.10a	General: New translation: German (thanks Ekkart Kleinod).	20	v0.6	General: New character styles <code>imprimerie-verse</code> , <code>imprimerie-prose</code> and <code>arden</code>	15
v0.10c	<code>\pauseverse</code> : Document other implementations of <code>\pauseverse</code> and <code>\resumeverse</code>	34	v0.7	<code>charactergroup</code> : Groups of characters are now aligned with other characters.	30
v0.5	General: First published version.	36		<code>\pauseverse</code> : New commands <code>\pauseverse</code> , <code>\resumeverse</code> ,	

<code>\adjustverse</code>	34	General: Add section <i>Examples</i> . . .	21
General: New package option		Add section <i>Localization</i>	4
<code>xspace</code>	5	<code>dramatisenv</code> : It is now possible to	
v0.9a		customize style of <code>dramatis</code>	
<code>\setthalieloptions</code> : Add		<code>personæ</code>	30
command <code>\setthalieloptions</code>	36		

References

- [1] Imprimerie nationale, *Lexique des règles typographiques en usage à l’Imprimerie nationale*, 2002, ISBN : 978-2-7433-0482-9
- [2] Edmond Rostand, *Cyrano de Bergerac*, 1897
- [3] William Shakespeare, *A Midsummer Night’s Dream*, 1600
- [4] Scott Pakin, *How to Package Your L^AT_EX Package — Tutorial on writing .dtx and .ins files*, 2004, <http://www.ctan.org/pkg/dtxut/>
- [5] The L^AT_EX Team, *L^AT_EX 2_ε for class and package writers*, 2006, <http://www.ctan.org/pkg/clsguide>
- [6] Matt Swift, *drama — Production-style stage script in LaTeX*, 2001, <http://www.ctan.org/pkg/drama>
- [7] Massimiliano Dominici, *dramatist — Typeset dramas, both in verse and in prose*, 2005, <http://www.ctan.org/pkg/dramatist>
- [8] James Kilfiger, *play — Typeset drama using L^AT_EX*, 2001, <http://www.ctan.org/pkg/play>
- [9] John Pate, *screenplay — A class file to typeset screenplays*, 2012, <http://www.ctan.org/pkg/screenplay>
- [10] Wing L Mui, *sides — A LaTeX class for typesetting stage plays*, 2005, <http://www.ctan.org/pkg/sides>

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.

Symbols		<code>\act*</code>	<u>104</u>
<code>\@setcharacterstyle</code>	<code>_</code>	<code>\actmark</code> <u>19</u> , 22, 117, 180	
. 225, 512–518	A	<code>\actname</code>	<u>13</u>
<code>\}</code> 374	<code>\act</code>	<code>\adjustverse</code>	<u>484</u>

C		E	<code>\playname</code> 13
<code>\character</code> 427	<code>\enddramatisenv</code> ... 322	environments:	<code>\providecommand</code> 223 , 224
<code>charactergroup</code> (environment) ... 325	<code>charactergroup</code> . 325	R	
<code>\characterspace</code> ... 325	<code>dida</code> 476	<code>\resumeverse</code> 484	
<code>\curtain</code> 202	<code>dramatis</code> 302	S	
<code>\curtainname</code> 13	<code>dramatisenv</code> 325	<code>\scene</code> 133	
<code>\customact</code> ... 120 , 128	I		<code>\scene*</code> 133
<code>\customplay</code> 91 , 99	<code>\interlude</code> 162	<code>\scenemark</code> 19 , 28 , 146 , 185
<code>\customscene</code> .. 149 , 157	<code>\interlude*</code> 162	<code>\scenename</code> 13	
D		<code>\interludename</code> 13	<code>\setcharactername</code> .
<code>\did</code> 475 , 483	N	 385 , 423
<code>dida</code> (environment) .. 476	<code>\notbool</code> 321 ,	<code>\setthalieloptions</code> . 605	
<code>\disposablecharacter</code> 471	340 , 363 , 371 , 455	<code>\speakswithdirection</code>	217 , 220
<code>dramatis</code> (environment) 302	O		<code>\speakswithoutdirection</code>
<code>\dramatischaracter</code> .	<code>\onstage</code> 474	215 , 220	
..... 325 , 461 , 467	P		
<code>\dramatischaracterdescription</code>	<code>\pause</code> 483	T	
..... 325	<code>\pausename</code> 13	<code>\theact</code> .. 24 , 30 , 33 ,	
<code>\dramatischaractername</code>	<code>\pauseverse</code> 484	112 , 114 , 120 , 122	
..... 325	<code>\play</code> 80	<code>\theplay</code> 30 , 31 , 91	
<code>\dramatisenv</code> 312	<code>\play*</code> 80	<code>\thescene</code> ... 30 , 35 ,	
<code>dramatisenv</code> (environment)	<code>\playmark</code> 19 , 19 , 88 , 175	141 , 143 , 149 , 151	
..... 325			