The **thalie** package*

A package to typeset drama plays

Louis Paternault
spalax(at)gresille(dot)org

June 16, 2022

**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 2
1.3 Acknowledgements 3
1.4 Examples 3
1.5 Overview 3

2 Download and Install 3

2.1 GNU/Linux Distribution 3
2.2 \LaTeX{} distribution 3
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 7
3.5 Characters 12
3.6 Stage directions 14
3.7 Splitting verses 15

4 Localization and Internationalization 18

4.1 Localization 18
4.2 Non-latin alphabets 18
4.3 Language specific comments 19

---

*This document corresponds to thalie v0.11b, dated 2022/06/16. Home page, bug requests, etc. at [http://framagit.org/spalax/thalie](http://framagit.org/spalax/thalie)
1 Introduction

This document introduces the \texttt{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): \texttt{drama} \cite{drama}, \texttt{dramatist} \cite{dramatist}, \texttt{play} \cite{play}, \texttt{screenplay} \cite{screenplay}, \texttt{sides} \cite{sides}.

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 \textsc{ctan}\footnote{http://www.ctan.org/topic/drama-script} and the \texttt{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 \texttt{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 \texttt{dramatist} instead of starting my own package. The good part is that I have a package that fits my needs, and I learned how to write a \LaTeX{} 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 \LaTeX{}Project 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 package.\footnote{You 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.}

1.3 Acknowledgements

I borrowed ideas and lines of code from the \LaTeX\ packages drama\footnote{\textit{drama}} and dramatist\footnote{\textit{dramatist}}. I used the following guides to package my package: \textit{How to Package Your \LaTeX\ Package} \footnote{\textit{How to Package Your \LaTeX\ Package}}, and \textit{\LaTeX\ 2ε for class and package writers} \footnote{\textit{\LaTeX\ 2ε for class and package writers}}.

1.4 Examples

Most of the examples are taken from Edmond Rostand’s \textit{Cyrano de Bergerac} \footnote{\textit{Cyrano de Bergerac}} or from William Shakespeare’s \textit{A Midsummer Night’s Dream} \footnote{\textit{A Midsummer Night’s Dream}}. The snippets of \textit{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 \ref{sec:download}. Documentation about how to use this package is given in section \ref{sec:documentation}. In particular, section \ref{sec:definition} explains how to use acts and scenes, sections \ref{sec:characters} and \ref{sec:lines} explains how to define characters, and use these definitions to introduce characters’ lines, and section \ref{sec:stagings} 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 \ref{sec:multilingual}. Examples are given in section \ref{sec:examples}.

Implementation is given in section \ref{sec:implementation}.

2 Download and Install

2.1 Gnu/Linux Distribution

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

\begin{verbatim}
sudo apt-get install texlive-humanities
\end{verbatim}

2.2 \LaTeX\ distribution

This package is included both in \TeX\Live and MiK\TeX. 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](http://mirrors.ctan.org/install/macros/latex/contrib/thalie.tds.zip)
  - **Development version** [https://framagit.org/spalax/thalie/repository/archive.zip?ref=main](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 LaTeX 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 LaTeX 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`. 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.

---

Given that someone provided the translation. See 4.3 for the list of available languages.
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:

\cyrano Indeed, \cyranoname is my name!

instead of:

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

If package option xspace is set (e.g. xspace=true), space is automatically added after those commands if necessary\footnote{This is done with the \texttt{xspace} command (from the \texttt{xspace} package), hence the name.} 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=\langle\textit{style}\rangle. 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=\langle\textit{style}\rangle, actstyle=\langle\textit{style}\rangle and scenestyle=\langle\textit{style}\rangle. 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 (\texttt{chapter}, \texttt{section} and so on), the relative importance of plays, acts and scenes is important. You can set this using options playlevel=\langle\textit{level}\rangle, actlevel=\langle\textit{level}\rangle and scenelevel=\langle\textit{level}\rangle. 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=\langle\textit{level}\rangle, instead of choosing one of \LaTeX\ 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

\sethalieoptions

Packages options can be reset in the document body using command \sethalieoptions{\langlepackage\textit{options}\rangle}. 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.

\begin{verbatim}
\documentclass{book}
...
\usepackage{thalie}
...
\begin{document}
\end{verbatim}
3.3 Sectioning

Here begin the parts explicitly relating to drama.

3.3.1 Levels

To introduce a new play, act or scene, use commands \play, \act and \scene. Their behaviour is as close as the “usual” sectioning commands (\chapter, \section and 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.

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.

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 \LaTeX) as usual sections, package titlesec cannot be used to use alternative section titles. Here is the way to set your own one.

When loading the package, use custom as the style of the title you want to customize (e.g. actstyle=custom). Then, commands customact\{\langle counter\rangle\}\{\langle title\rangle\} and customact*\{\langle title\rangle\} 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}{\text{Alph}(act)}.

3.3.4 Headers and footers

\playmark 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\{\langle label\rangle\}, \actmark\{\langle label\rangle\} and \scenemark\{\langle label\rangle\} 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.

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}\{\langle hidden\rangle\}**

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

**\begin{dramatis}{\langle hidden\rangle}**

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

\textit{http://www.ctan.org/pkg/titlesec}
3.4.1 Character definition

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: \(\langle \text{cmd} \rangle\) and \(\langle \text{cmd} \rangle\text{name}\). 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.

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.

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

\begin{verbatim}
\character[
  drama={A ghost},
  desc={the king’s ghost},
  cmd={ghost},
]{The ghost}
\end{verbatim}

Table 2: (Not) defining arguments in character definition
newcommand{customact}[2]{
  \begin{center}
  textsc{\text{GetTranslation\{Act\} \#1}}
  #2
  \end{center}
}

WithSuffix newcommand{customact*}[1]{
  \begin{center}
  #1
  \end{center}
}

Figure 1: Example of custom act definition

\begin{dramatis}
\character[cmd={cyrano}, drama={Cyrano de Bergerac}]{Cyrano}
\character[cmd={lebret}]{Le Bret}
\character[cmd={bellerose}]{Bellerose}
\end{dramatis}

\baselineskip
\lebret[to \cyranoname, holding his arm]
Let's talk !

\cyrano
Wait for the crowd to leave. \did{To \bellerosename} Can I stay?

<table>
<thead>
<tr>
<th>Cyrano de Bergerac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Le Bret</td>
</tr>
<tr>
<td>Bellerose</td>
</tr>
</tbody>
</table>

\textit{Le Bret, to Cyrano, holding his arm:} Let’s talk !
\textit{Cyrano:} Wait for the crowd to leave. (To Bellerose) Can I stay?

Figure 2: Example of character definition
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:

```
\character{desc={The kings' armies}}{}
```

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:

```
\character{drama={}, cmd={postman}}{The postman}
```

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:

```
\character{A priest}
```

Group of characters It is possible to group several characters' definition if they have the same description. This is done with environment `charactergroup`\[\langle$width$\rangle$\]{$\langle$description$\rangle$}. 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 undefined.

3.4.2 Customize style of dramatis personæ

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{\langle$name$\rangle$}{\langle$description$\rangle$}` Renders the character name and description. It should take care of cases where name or description (but not both) is missing.
- `\dramatischaractername{\langle$name$\rangle$}` Renders the character name.
- `\dramatischaracterdescription{\langle$description$\rangle$}` Renders the character description.
- Environment `charactergroup`\[\langle$width$\rangle$\]{$\langle$description$\rangle$} Renders a group of characters.
- `\characterspace` Adds a vertical character space.
\begin{dramatis}
\begin{charactergroup}[2.5cm]{Officers}
\character[desc={a soldier}]{Marcellus}
\character[desc={a soldier}]{Bernardo}
\end{charactergroup}
\end{dramatis}

\begin{figure}[h]
\centering
\begin{tabular}{|c|c|}
\hline
Marcellus, a soldier & Officers \\
Bernardo, a soldier & \\
\hline
\end{tabular}
\caption{Defining a group of characters}
\end{figure}

\begin{dramatis}
\begin{character}{The queen}
\begin{charactergroup}{Lords and ladies}
\character{Lord Foo}
\character[desc={Lord Foo’s sister.}]{Lady Bar}
\character[desc={Lord and Lady Baz.}]{}
\end{charactergroup}
\begin{characterspace}
\character[desc={Several servants}]{}
\end{characterspace}
\end{character}
\end{dramatis}

Figure 3: Defining a group of characters

\begin{dramatisenv}
\dramatischaracter{The queen}{}
\dramatischaracter{The king}{{the queen’s husband.}}
\begin{charactergroup}{Lords and ladies}
\dramatischaracter{Lord Foo}{}
\dramatischaracter{Lady Bar}{Lord Foo’s sister.}
\dramatischaracter{}{Lord and Lady Baz.}
\end{charactergroup}
\begin{characterspace}
\dramatischaracter{}{Several servants}
\end{characterspace}
\end{dramatisenv}

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

\begin{dramatisenv}
\dramatischaracter{\langle Name \rangle}{\langle Description \rangle}
\end{dramatisenv}

And command \texttt{\dramatischaracter\{\langle Name\}\{\langle 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).

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

Figure 4: Commands used to display the dramatis personæ.
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{}{\rightmargin1cm\leftmargin2cm}\item[]\
\act*{Dramatis personæ}}{\endlist%}

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 than Cyrano’s lines can now be introduced using the following code.

\cyrano
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

\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{\langle\textsc{command}\rangle}{\langle\textsc{name}\rangle}

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.

\setcharactername{\langle\textsc{command}\rangle}{\langle\textsc{name}\rangle}

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.
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.

\begin{dramatis}[hidden]
\character[desc={Cyrano de Bergerac}, cmd=cyrano]{The voice}
\character[cmd=montfleury]{Montfleury}
\end{dramatis}

\cyrano Leave!
\disposablecharacter{The crowd} Oh!
\montfleury[losing his voice] ‘‘Happy he who far away from court\dots’’
\setcharactername{cyrano}{Cyrano}
\cyrano[emerging from the crowd, standing up on a chair {[}\ldots{]} with a terrible nose] Ah! I am getting angry!\dots

\begin{verbatim}
THE VOICE: Leave!
THE CROWD: Oh!
MONTFLEURY, losing his voice: “Happy he who far away from court…”
CYRANO, emerging from the crowd, standing up on a chair [...] with a terrible nose: Ah! I am getting angry!...
\end{verbatim}

Figure 5: Example of changing the name of a 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 \[2\]).

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.)).

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\speakswithdirection

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\dida

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.

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 16) 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.

\newcommand{\pv}{\pauseverse}
\newcommand{\rv}{\resumeverse}

This is not done by default, because short command names are a scarce resource in \LaTeX, 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 \LaTeX

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

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

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

---

\footnote{Originally written as an answer to a StackExchange question: see \url{https://tex.stackexchange.com/a/107727}}

\footnote{One on them lead to the next recipe.}
Hermia
So is Lysander.

Theseus
In himself he is.

Figure 6: Example of commands to split verse

\adjustverse{-7em}

Hermia
So is Lysander.

Theseus
\phantom{So is Lysander.} In himself he is.

Figure 8: Example of commands to split verse (using the \phantom command)
when calling \resumeverse, to get a correct alignment. See figure\[7\] page\[16\] for an example of those three commands.

The young inexperienced fool\[9\] 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 StackExchange\[10\]. He proposes the following code to fix this.

\begin{verbatim}
\usepackage{hanging}
\newlength{\brokenverse}
\ renewcommand*{\speakswithoutdirection}{% 
  \settowidth{\brokenverse}{\textsc{#1}.}%
  \hangpara{2\ parindent}{1\ noindent\textsc{#1}.}%
}\makeatletter
\ renewcommand{\resumeverse}{% 
  \hspace{\@verseadjust}\hspace{\dimexpr \dimen@ - \brokenverse -2em}
}\makeatother
\end{verbatim}

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 feature\[11\].

\begin{verbatim}
\usepackage{tikz}
\usetikzlibrary{tikzmark}
\usetikzlibrary{math}
\newcounter{thalieverse}
\tikzset{tikzmark suffix=\theralieverse}\
\ renewcommand{\pauseverse}{% 
  \stepcounter{thalieverse}\
  \tikzmark{thaliepauseverse}\
}\ renewcommand{\resumeverse}{% 
  \tikzmark{thalieresumeverse}\
  \iftikzmark{thaliepauseverse}{% 
  Me...\[9\]

\[10\]https://tex.stackexchange.com/a/545326
\[11\]In French, we would say: “Un marteau-pilon pour écraser une mouche” (a far too heavy solution for a tiny problem).
This recipe is implemented in the `examples/henry.tex` file in the repository of this project.

3.7.5 Recipe 4: With \texttt{\phantom{}}

At last, if nothing else worked, this feature can also be implemented using the \texttt{\phantom{}} command (see figure 8 for an example, page 16). 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 \texttt{babel} or \texttt{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 exist\footnote{You can also send me the translation for your language, to improve this package.} or if you want to change the default words used here, you can set your own translation using command \texttt{\DeclareTranslation} (see the \texttt{translations} package for more information). For instance use \texttt{\DeclareTranslation{Spanish}{Curtain}{Telón}} for Spanish\footnote{I may be wrong: I cannot speak Spanish.}.

See section 4.3 for the list of supported languages.

4.1.1 Common translations

\begin{verbatim}
\playname \actname \scenename \interludename \pausename \curtainname
\end{verbatim}

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

4.2 Non-latin alphabets

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

\begin{verbatim}
\begin{tikzpicture}[remember picture]
% Uncomment to see "pause" and "resume" tikz marks.
%\draw[overlay,green, ultra thick] (pic cs:thaliepauseverse) -- (pic cs:thalieresumeverse);
\tikzmath{
  coordinate \c;
  \c1 = (pic cs:thaliepauseverse);
  \c2 = (pic cs:thalieresumeverse);
  \width = max(0, \cx1 - \cx2);
}
% Set opacity to 1 to see the space
\draw[opacity=0, black, line width=1ex] (0, 0) -- (\width pt, 0);
\end{tikzpicture}%
\end{verbatim}
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: \texttt{\newfontfamily{Noto Sans Devanagari}}.

4.3 Language specific comments

The following languages are available:

- English,
- French,
- German,
- Italian.

4.3.1 English

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

4.3.2 French

- The imprimerie-verse and imprimerie-prose character styles implement 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: \texttt{\renewcommand{\thescene}{\textsc{\texttt{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 \texttt{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.
Let’s open it.

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.

```
\disposablecharacter{alice \footnote{An interesting note.}}
```

5.1.3 Add a footnote in a stage direction

Footnote must be protected to be included in stage directions.

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

5.2 Headers and footers

By default, with the \texttt{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

```
\renewcommand{\markboth}{\MakeUppercase{\GetTranslation{Act}}\ theact \ #1}{%}
\renewcommand{\markright}{\MakeUppercase{\GetTranslation{Scene}}\ thescene \ #1}{%}
```

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

```
usepackage{fancyhdr}
pagestyle{fancy}
fancyhf{}
fancyhead[RO]{% 
\textit{\MakeUppercase{\GetTranslation{Act}} theact}}, % 
\GetTranslation{Scene} thescene%
}
fancyhead[LE]{\leftmark}
```
6 Implementation

6.1 Required packages

Loading some packages.
\%
\RequirePackage{pgfkeys}
\%
\RequirePackage{etoolbox}
\%
\RequirePackage{suffix}
\%
\RequirePackage{tabularx}
\%
\RequirePackage{xspace}

6.2 Localization

Use \texttt{translations}. Translations are defined in the dicts/thalie-\texttt{<LANGUAGE>}.trsl files.
\%
\RequirePackage{translations}
\%
\LoadDictionary{thalie}
\%
\LoadDictionaryFor{fallback}{thalie}

6.3 Obsolete localization commands

\newcommand{\playname}{\GetTranslation{Play}}
\newcommand{\actname}{\GetTranslation{Act}}
\newcommand{\scenename}{\GetTranslation{Scene}}
\newcommand{\interludename}{\GetTranslation{Interlude}}
\newcommand{\curtainname}{\GetTranslation{Curtain}}
\newcommand{\pausename}{\GetTranslation{Pause}}

6.4 Sectioning

6.4.1 Headears, footers, counters, etc.

\newcommand{\playmark}[1]{\markboth{\MakeUppercase{#1}}{}}
\newcommand{\actmark}[1]{\markright{\MakeUppercase{\GetTranslation{Act} \theact\ifdefempty{#1}{:: #1}}}{}
\newcommand{\scenemark}[1]{\GetTranslation{Scene}}

\newcommand{\theplay}{\MakeUppercase{\theplay}}
\newcommand{\theact}{\MakeUppercase{\theact}}
\newcommand{\thescene}{\MakeUppercase{\thescene}}
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.

\newcommand{\displaytitle}[3]{
  % Arguments:
  %   - Style
  %   - Label (none = not in toc)
  %   - Title
  \ifdefstring{#1}{center}{
    \begin{center}
    \textsc{#2}
    #3
    \end{center}
  }{
    \ifdefstring{#1}{bigcenter}{
      \begin{center}
      \Large
      \textsc{#2}
      #3
      \end{center}
    }{
      \ifdefstring{#1}{box}{
        \begin{center}
        \framebox{\begin{minipage}{0.7\textwidth}
        \begin{center}
        \Large \bfseries
        \vspace{0.5em}
        #2
        \ifboolexpr{test{\ifstrempty{#3}} or test{\ifstrempty{#2}}}{}{---}
        #3
        \vspace{0.5em}
        \end{center}
        \end{minipage}}
        \end{center}
      }{
      }}}}

6.4.3 Sectioning commands

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

\newcommand{\clearpage}{% Clear page if necessary
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.

\newcommand{\play}[2][\{}{%
  \refstepcounter{play}%
  \ifstrempty{#1}{% #1 is optional
    \def\@short{#2}%
  }{% #1 is mandatory
    \def\@short{#1}%
  }
  \@clearpage{\@playlevel}%
  \playmark{\@short}%
  \addcontentsline{toc}{\@playlevel}{\@short}%
  \ifdefstring{\@playstyle}{custom}{%
    \customplay{\theplay}{#2}%;
  }{%
  @displaytitle{\@playstyle}{}{#2}%;
}
\WithSuffix\newcommand{\play*}[1][\{}{%
  \@clearpage{\@playlevel}%
  \ifdefstring{\@playstyle}{custom}{%
    \customplay*{#1}%;
  }{%
  @displaytitle{\@playstyle}{}{#1}%;
}

Define commands to introduce acts.

\newcommand{\act}[2][\{}{%
  \refstepcounter{act}%
  \ifstrempty{#1}{% #1 is optional
    \def\@short{#2}%
  }{% #1 is mandatory
    \def\@short{#1}%
  }
  \ifdefempty{\@short}{% #1 is iteration
    \def\@label{\GetTranslation{Act} \theact{}%}
  }{% #1 is iteration
    \def\@label{\GetTranslation{Act} \theact{}xspace: }%;
  }
  \@clearpage{\@actlevel}%
  \actmark{\@short}%
  \addcontentsline{toc}{\@actlevel}{\@label\@short}%
  \ifdefstring{\@actstyle}{custom}{% #2 is iteration
    \customact{\theact}{#2}%;
  }{% #2 is iteration
  @displaytitle{\@actstyle}{}{#1}%;
}
\WithSuffix\newcommand{\act*}[1][\{}{%
  \@clearpage{\@actlevel}%
  \ifdefstring{\@actstyle}{custom}{% #1 is iteration
    \customact*{#1}%;
  }{% #1 is iteration
  @displaytitle{\@actstyle}{}{#1}%;
}
\displaytitle{\actstyle}{\GetTranslation{Act} \theact} (#2)

WithSuffix\newcommand\act*[1] {% 
@clearpage{\actlevel}
@ifdefstring{\actstyle}{custom}{
  \customact*[#1]
} {
  \displaytitle{\actstyle}{}{#1}
}

newcommand\act* [1] {% 
  @clearpage{\actlevel}
  ifdefstring{\actstyle}{custom}{
    \customact*{#1}
  }{
    \displaytitle{\actstyle}{}{#1}
  }
}

\scene Define commands to introduce scenes.
\newcommand{\scene}[2] [] {%
  \refstepcounter{scene}
  \ifstrempty{#1}{
    \def\@short{#2}
  }{
    \def\@short{#1}
  }
  \ifdefempty{\@short}{
    \def\@label{\GetTranslation{Scene} \thescene{}}
  }{
    \def\@label{\GetTranslation{Scene} \thescene{} xspace: }
  }
  \@clearpage{\@scenelevel}
  \scenemark{\@short}
  \addcontentsline{toc}{\@scenelevel}{\@label\@short}
  \ifdefstring{\@scenestyle}{custom}{
    \customscene{\thescene}{#2}
  }{
    \displaytitle{\@scenestyle}{\GetTranslation{Scene} \thescene}{#2}
  }
}

\scene* Define commands to introduce scenes.
\newcommand{\scene*}[1] [] {%
  \refstepcounter{scene}
  \ifstrempty{#1}{
    \def\@short{#2}
  }{
    \def\@short{#1}
  }
  \ifdefempty{\@short}{
    \def\@label{\GetTranslation{Scene}}
  }{
    \def\@label{\GetTranslation{Scene} xspace: }
  }
  \@clearpage{\@scenelevel}
  \scenemark{\@short}
  \ifdefstring{\@scenestyle}{custom}{
    \customscene*{#1}
  }{
    \displaytitle{\@scenestyle}{}{#1}
  }
}

\interlude Define commands to introduce interludes.
\newcommand{\interlude}[2] [] {%
  \ifstrempty{#1}{
    \def\@short{#2}
  }{
    \def\@short{#1}
  }
  \ifdefempty{\@short}{
    \def\@label{\GetTranslation{Interlude}}
  }{
    \def\@label{\GetTranslation{Interlude} xspace: }
  }
  \@clearpage{\@scenestyle}
  \addcontentsline{toc}{\@scenestyle}{\@label\@short}
  \ifdefstring{\@scenestyle}{custom}{
    \customscene{\@scenestyle}{#2}
  }{
    \displaytitle{\@scenestyle}{\GetTranslation{Scene} \theinterlude}{#2}
  }
}
\interlude* Define commands to introduce interludes.
\newcommand{\interlude*}[1] [] {%
  \ifstrempty{#1}{
    \def\@short{#2}
  }{
    \def\@short{#1}
  }
  \ifdefempty{\@short}{
    \def\@label{\GetTranslation{Interlude}}
  }{
    \def\@label{\GetTranslation{Interlude} xspace: }
  }
  \@clearpage{\@scenestyle}
  \addcontentsline{toc}{\@scenestyle}{\@label\@short}
  \ifdefstring{\@scenestyle}{custom}{
    \customscene*{#1}
  }{
    \displaytitle{\@scenestyle}{}{#1}
  }
}
6.4.4 Curtain
\curtain Used to mark the end of an act. Prints “curtain” centered in its own line.

6.5 Characters
6.5.1 Spacing
Add (or not) an $\backslash\text{xspace}$ command at the end of character names, depending on the xspace package option.
6.5.2 Characters’ line

Define the command \@speaks, which introduce 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.

\newcommand\@speaks[2][]{
  \ifstrempty{#1}{
    \speakswithoutdirection{#2}\%
  }{
    \speakswithdirection{#2}{#1}\%
  }\@maybexspace%
}

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

\newenvironment{@smallcenter}{\par\smallskip\centering}{\par\nopagebreak\ignorespacesafterend}
\providecommand\speakswithdirection{}
\providecommand\speakswithoutdirection{}
\newcommand\@setcharacterstyle[1]{
  \ifstrequal{#1}{bold}{%)
    \% Bold style
    \renewcommand\speakswithdirection[2][]{%)
      \noindent%}
      \textbf{\sffamily ##1} \textit{##2} xspace:%
    }%)
  \renewcommand\speakswithoutdirection[1][]{%)
    \noindent%}
    \textbf{\sffamily ##1} xspace:%
  }%)
  \ifstrequal{#1}{center}{%)
    \% Center style
    \renewcommand\speakswithdirection[2][]{%)
      \begin{center}%
      \textsc{##1}, \textit{##2}%
    \end{center}%
    \par\ignorespacesafterend%
  }%)
  \renewcommand\speakswithoutdirection[1][]{%)
    \begin{center}%
    \textsc{##1}%
  \end{center}%
  \par\ignorespacesafterend%
  }%)
  \ifstrequal{#1}{imprimerie-verse}{%)
    \% Style for verse plays defined by the French Imprimerie nationale
    \renewcommand\speakswithdirection[2][]{%)
      \begin{imprimerie-verse}%
      \textsc{##1}, \textit{##2}%
    \end{imprimerie-verse}%
  }%)
  }%)
\ifsstrequal{#1}{imprimerie-verse}{}%)

6.5.3 Dramatis personæ

In this environment are defined characters.

\newcommand{\dramatis@clear}{}
\newcommand{\@empty}{}
\providebool{@dramatis@hidden}
\newenvironment{dramatis}[1][]{}
\@dramatis@clear{}
\edef{\@dramatis@clear}{}
\ifstrequal{#1}{hidden}{
\noindent\hspace*{3cm}\textsc{##1}, \emph{##2}\xspace:%
}\noindent\hspace*{3cm}\textsc{##1}\xspace:%
}\ifstrequal{#1}{arden}{
\noindent\hspace*{-3cm}##1##2
}\ifstrequal{#1}{simple}{
\indent\textsc{##1}, \emph{##2}\xspace:%
}\indent\textsc{##1}\xspace:%
}\ifstrequal{#1}{margin}{
\setlength{\leftskip}{3cm}
\noindent\hspace*{(-3cm)}##1##2
}\hspace*{(-3cm)}##1
\}
\iffalse
\booltrue{@dramatis@hidden}
{\ifstrempty{#1}\{
\boolfalse{@dramatis@hidden}
{\dramatisenv
{\ClassError{thalie}{%
Environment dramatis does not accept option #1.%
}{}
}{\notbool{@dramatis@hidden}{%}
\enddramatisenv
}{%}
}
{\notbool{@dramatis@hidden}{%}
\enddramatisenv
}{%}
}
% Environment used to *display* dramatis personae.
\newenvironment{dramatisenv}{%}
\list{}{\rightmargin1cm\leftmargin2cm}\item[]
}{%}
% Display character name
\newcommand{\dramatischaractername}[1]{\textbf{#1}}
% Display character description
\newcommand{\dramatischaracterdescription}[1]{#1}
% Adds a vertical space
\newcommand{\characterspace}{%}
% Display dramatis personae name and description
\newcommand{\dramatischaracter}[2]{%}
\hspace*{-1cm} %
{\ifboolexpr{(not test \ifdefempty{#1}) and test \ifdefempty{#2})}{%}
\dramatischaractername{#1} %
}{%}
{\ifboolexpr{ test\ifdefempty{#1} and not test\ifdefempty{#2})}{%}
\dramatischaracterdescription{#2} %
}{%}
{\ifboolexpr{ (not test\ifdefempty{#1}) and (not test\ifdefempty{#2})}{%}
\dramatischaractername{#1}, \dramatischaracterdescription{#2} %
}{%}
\newline %
}{%}
% Environment to manage groups of characters
\newlength{@spaceaftergroup}
\newenvironment{charactergroup}[2][5cm]{%
  \notbool{@dramatis@hidden}{%
    \gdef@groupname{#2} %
    \gdef@grouplength{#1} %
    \hspace*{-1.3pt}\math\left. %
    \minipage[c]{#1} %
    \vspace*{2pt} %
  }{%
    \notbool{@dramatis@hidden}{%
    \addtolength\@spaceaftergroup{-20pt}
    \addtolength\@spaceaftergroup{-1.3pt}
    \minipage[c]{\@spaceaftergroup}
    \@groupname %
    \end{minipage}
  }%}
}

6.5.4 Character definitions
\setcharactername Set (or change) the name used to introduce the lines of a character.
\newcommand{\setcharactername}[2]{%
\expandafter\gdef\csname#1name\endcsname{#2\@maybexspace%}
\expandafter\gdef\csname#1\endcsname{%
  \@ifnextchar[{
    \defcharcommand@with{#2}%
  }{%
    \defcharcommand@without{#2}%
  }%
}
\xappto{\@dramatis@clear}{%
\global\noexpand\csundef{#1}%
\global\noexpand\csundef{#1name}%
}%
}

\definecharactercommand 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.
\newcommand{\definecharactercommand}[2]{%
\ifcsdef{#1}{%
\ClassError{thaliel}{}%
A command named \backslash#1 already exists. We cannot define a new one.}%
}
Choose another command name to introduce character #2’s lines.{}

\ifcsdef{#1}{%
\ClassError{thalie}{% A command named \backslash#1 already exists. We cannot define a new one. %}
}\}%

Choose another command name to introduce character #2’s lines, such that when a new command is defined by adding "name" to it, it does not conflict with an existing one.

}%

\setcharactername{#1}{#2}{}

\def\defcharcommand@with#1[#2]{\@speaks[#2]{#1}}
\def\defcharcommand@without#1{\@speaks{#1}}

\character

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

\pgfkeys{
% Character definition
/THALIE/CHARACTER/.is family,
/THALIE/CHARACTER,
 cmd/.value required,
 cmd/.store in={@cmd},
 drama/.value required,
 drama/.store in={@drama},
 desc/.value required,
 desc/.store in={@desc},
}

\newcommand{\character}[2][]{
\undef{@drama}
\undef{@cmd}
\undef{@desc}
\pgfkeys{/THALIE/CHARACTER, #1}%
% Forbidden combinations
\ifboolexpr{
( test{\ifdef{@cmd}} and test{\ifstrempty{#2}} ) or
( test{\ifstrempty{#2}} and test{\ifundef{@cmd}} and test{\ifundef{@desc}} and test{\ifundef{@drama}} ) or
( test{\ifundef{@cmd}} and (not test{\ifstrempty{#2}}) and test{\ifdef{@drama}} )
}{%
\ClassError{thalie}{Invalid character definition.}{All combination of omitted arguments are not allowed. See the documentation for more information}
}{%}
% Defining character command
\ifboolexpr{ test{\ifdef{@cmd}} and (not test{\ifstrempty{#2}}) }{%}
\ifdefempty{@drama}{%
\ifdefempty{@cmd}{}%
% Hidden character. Nothing added to dramatis personæ
)%
% Populating dramatis personæ
\dramatischaracter{@drama}{@empty}%
)%
% Populating dramatis personæ
\ifsundef{@desc}{\gdef{@desc{}}}{}%
\ifsundef{@drama}{\gdef{@drama{#2}}}{}%
\dramatischaracter{@drama}{@desc}%
}%
}%
}%
% Populating dramatis personæ
\ifsundef{@drama}{\gdef{@drama{}}}{}%
\ifsundef{@desc}{\gdef{@desc{}}}{}%
\dramatischaracter{@drama}{@desc}%
}%
}%
}%
}%
}%
}%
}%
disposablecharacter Disposable character (character used only a few times, defined on-the-fly, that
does not appear in the dramatis personæ).
\newcommand{\disposablecharacter}[2][]{%
@speaks[#1]{#2}%
}

6.6 Stage directions
\onstage Centered stage direction.
\newcommand{\onstage}[1]{{\centering \emph{#1}\par\medskip}}
\did Inline stage directions.
\newcommand{\did}[1]{{\emph{(#1)}}}
\did Bigger stage directions, in its own paragraph.
\newenvironment{\did}{\begin{quote}\begin{em}}{\end{em}\end{quote}}
\pause Shortcut for \did{\GetTranslation{Pause}}.
\newcommand{\pause}{\did{\GetTranslation{Pause}}}

6.7 Splitting verses
\pauseverse \resumeverse \adjustverse Commands to split a verse between several characters.
% Thanks to Timothy Li for his question, and David Carlisle for his answer:
% https://tex.stackexchange.com/questions/107726/107727#107727
\newlength{\overseadj}
\setlength{\overseadj}{\Opt}
\newcommand{\adjustverse}[1]{{\setlength{\overseadj}{#1}}}
\newcommand{\pauseverse}{\above\displayshortskip\z\above\displayskip\z\z}
\newcommand{\resumeverse}{\above\displayshortskip\z\above\displayskip\z\z}
6.8 Package options

Manage package options.

\RequirePackage{pgfplots}
\pgfkeys{
  % Character style
  \THALIE/.cd,
  characterstyle/.value required,
  characterstyle/.default=simple,
  characterstyle/.is choice,
  characterstyle/bold/.code=\@setcharacterstyle{bold},
  characterstyle/center/.code=\@setcharacterstyle{center},
  characterstyle/margin/.code=\@setcharacterstyle{margin},
  characterstyle/simple/.code=\@setcharacterstyle{simple},
  characterstyle/arden/.code=\@setcharacterstyle{arden},
  characterstyle/imprimerie-verse/.code=\@setcharacterstyle{imprimerie-verse},
  characterstyle/imprimerie-prose/.code=\@setcharacterstyle{imprimerie-prose},
  characterstyle,
}
\pgfkeys{
  % play style
  \THALIE/.cd,
  playstyle/.value required,
  playstyle/.default=box,
  playstyle/.is choice,
  playstyle/center/.code=\def\@playstyle{center},
  playstyle/bigcenter/.code=\def\@playstyle{bigcenter},
  playstyle/box/.code=\def\@playstyle{box},
  playstyle/custom/.code=\def\@playstyle{custom},
  playstyle,
}
\pgfkeys{
  % act style
  \THALIE/.cd,
  actstyle/.value required,
  actstyle/.default=bigcenter,
  actstyle/.is choice,
  actstyle/center/.code=\def\@actstyle{center},
  actstyle/bigcenter/.code=\def\@actstyle{bigcenter},
  actstyle/box/.code=\def\@actstyle{box},
}
actstyle/custom/.code=\def\@actstyle{custom},
actstyle,
\pgfkeys{
\pgfkeys{% scene style
/THALIE/.cd,
scenestyle/.value required,
scenestyle/.default=center,
scenestyle/.is choice,
scenestyle/center/.code=\def\@scenestyle{center},
scenestyle/bigcenter/.code=\def\@scenestyle{bigcenter},
scenestyle/box/.code=\def\@scenestyle{box},
scenestyle/custom/.code=\def\@scenestyle{custom},
scenestyle,
}
\pgfkeys{
\pgfkeys{% play level
/THALIE/.cd,
playlevel/.value required,
playlevel/.default=chapter,
playlevel/.store in=@playlevel,
playlevel,
}
\pgfkeys{
\pgfkeys{% act level
/THALIE/.cd,
actlevel/.value required,
actlevel/.default=section,
actlevel/.store in=@actlevel,
actlevel,
}
\pgfkeys{
\pgfkeys{% scene level
/THALIE/.cd,
scenelevel/.value required,
scenelevel/.default=subsection,
scenelevel/.store in=@scenelevel,
scenelevel,
}
\pgfkeys{
\pgfkeys{% interlude level
/THALIE/.cd,
interludelevel/.value required,
interludelevel/.default=act,
interludelevel/.is choice,
interludelevel/play/.code=\def\@interludelevel{play},
interludelevel/act/.code=\def\@interludelevel{act},
interludelevel/scene/.code=\def\@interludelevel{scene},
interludelevel,
}
\newif\if@xspace
\pgfkeys{
\pgfkeys{% xspace option

\setthalieoptions Redefine package options.
\newcommand{\setthalieoptions}{\pgfkeys{\THALIE/.cd, #1}}%
\pgfkeys{\THALIE/.cd, xspace/.value required, xspace/.is if=@xspace, xspace/.default=true, xspace,}
\ProcessPgfPackageOptions{\THALIE}

Change History

v0.10a
  General: New translation: German
  (thanks Eckart Kleinod) . . . . . 19
v0.10c
  \pauseverse: Document other implementations of \pauseverse and \resumeverse . . . . . . . . . . . . 31
v0.11a
  General: New translation: Italian
  (thanks Giuseppe Palma) . . . . . 19
v0.5
  General: First published version. 34
v0.6
  General: New character styles \imprimerie-verse, \imprimerie-prose and arden. 14
v0.7
  \sethalieoptions: Add command \sethalieoptions. 34
v0.9a
  \sethalieoptions: Add \sethalieoptions. 34
v0.9a
  \sethalieoptions: Add \sethalieoptions. 34

References

[2] Edmond Rostand, Cyrano de Bergerac, 1897
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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>\dramatischaracter</th>
<th>\pausename</th>
</tr>
</thead>
<tbody>
<tr>
<td>@setcharacterstyle</td>
<td>. . . . 225, 512–518</td>
<td>. . . . 13</td>
</tr>
<tr>
<td>}</td>
<td>. . . . . . . . . . . . . . 374</td>
<td>. . . . . . . . . . . . . . 24</td>
</tr>
<tr>
<td>\u</td>
<td>\dramatischaracter</td>
<td>\pausename</td>
</tr>
<tr>
<td>A</td>
<td>. . . . . . . . . . . . 104</td>
<td>. . . . . . . . . . . . . . 484</td>
</tr>
<tr>
<td>\act</td>
<td>. . . . . . . . . . . . 104</td>
<td>. . . . . . . . . . . . . . 484</td>
</tr>
<tr>
<td>\act*</td>
<td>. . . . . . . . . . . . 104</td>
<td>. . . . . . . . . . . . . . 484</td>
</tr>
<tr>
<td>\actmark</td>
<td>19, 22, 117, 180</td>
<td>13</td>
</tr>
<tr>
<td>\actname</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>\adjustverse</td>
<td>. . . . . . . . . . . . 104</td>
<td></td>
</tr>
<tr>
<td>\charactergroup</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\characterspace</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\curtain</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\curtainname</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\customact</td>
<td>120, 128</td>
<td></td>
</tr>
<tr>
<td>\customplay</td>
<td>91, 99</td>
<td></td>
</tr>
<tr>
<td>\customscene</td>
<td>149, 157</td>
<td></td>
</tr>
<tr>
<td>\didi</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\disposablecharacter</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatis</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatisenv</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatisenv (environment)</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatischaracter</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatischaracterdescription</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatischaractername</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatisenv</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatisenv (environment)</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\enddramatisenv</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\environments:</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\charactergroup</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dida</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatis</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\dramatisenv</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\interlude</td>
<td>. . . . . . . . . . . . 162</td>
<td></td>
</tr>
<tr>
<td>\interlude*</td>
<td>. . . . . . . . . . . . 162</td>
<td></td>
</tr>
<tr>
<td>\interludename</td>
<td>. . . . . . . . . . . . 13</td>
<td></td>
</tr>
<tr>
<td>\notbool</td>
<td>. . . . . . . . . . . . 325</td>
<td></td>
</tr>
<tr>
<td>\onstage</td>
<td>. . . . . . . . . . . . 474</td>
<td></td>
</tr>
<tr>
<td>\pause</td>
<td>. . . . . . . . . . . . 483</td>
<td></td>
</tr>
<tr>
<td>\providecommand</td>
<td>. . . . . . . . . . . . 223, 224</td>
<td></td>
</tr>
<tr>
<td>\resumeverse</td>
<td>. . . . . . . . . . . . 484</td>
<td></td>
</tr>
<tr>
<td>\scene</td>
<td>. . . . . . . . . . . . 133</td>
<td></td>
</tr>
<tr>
<td>\scene*</td>
<td>. . . . . . . . . . . . 133</td>
<td></td>
</tr>
<tr>
<td>\scenemark</td>
<td>. . . . . . . . . . . . 13</td>
<td></td>
</tr>
<tr>
<td>\scenename</td>
<td>. . . . . . . . . . . . 13</td>
<td></td>
</tr>
<tr>
<td>\setcharactername</td>
<td>. . . . . . . . . . . . 385, 423</td>
<td></td>
</tr>
<tr>
<td>\sethalieoptions</td>
<td>. . . . . . . . . . . . 423</td>
<td></td>
</tr>
<tr>
<td>\settheact</td>
<td>. . . . . . . . . . . . 217, 220</td>
<td></td>
</tr>
<tr>
<td>\settheplay</td>
<td>. . . . . . . . . . . . 215, 220</td>
<td></td>
</tr>
<tr>
<td>\settheact</td>
<td>. . . . . . . . . . . . 217, 220</td>
<td></td>
</tr>
<tr>
<td>\settheplay</td>
<td>. . . . . . . . . . . . 215, 220</td>
<td></td>
</tr>
<tr>
<td>\theact</td>
<td>. . . . . . . . . . . . 24, 30, 33, 112, 114, 120, 122</td>
<td></td>
</tr>
<tr>
<td>\theplay</td>
<td>. . . . . . . . . . . . 30, 31, 91</td>
<td></td>
</tr>
<tr>
<td>\thescene</td>
<td>. . . . . . . . . . . . 30, 35, 141, 143, 149, 151</td>
<td></td>
</tr>
</tbody>
</table>