The phfextendedabstract package[1]

Philippe Faist *philippe.faist@bluewin.ch*

September 08, 2021

[1] *This document corresponds to phfextendedabstract v1.0, dated 2021/09/08. It is part of the phfqitltx package suite, see https://github.com/phfaist/phfqitltx.*

phfextendedabstract—Typeset extended abstracts for conferences, such as often encountered in quantum information theory.

1 Introduction **1**

2 Main features **2**

3 Example Usage **3**

4 Class options **4**

5 Macros and environments **5**
5.1 Sections and paragraphs 5
5.2 Vertical spacing 7
5.3 List environments 8
5.4 Customization of the main title and general headings style 9

6 Implementation **9**

Change History **18**

Index **19**

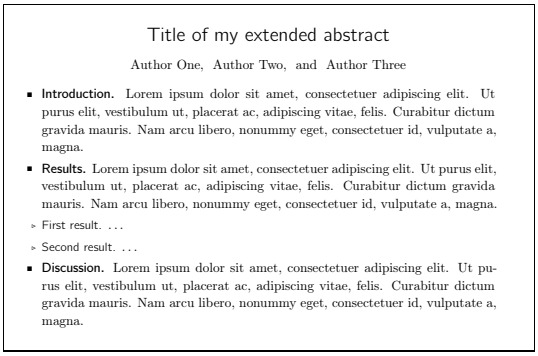
■ 1 Introduction

Several conferences in Quantum Information Theory (and presumably in other fields, too) require the submission of *extended abstracts*. An extended abstract is a summary of a scientific result, presented at a high level, and consisting of at most a small handful of pages.

I found myself repeatedly copying my LaTeX preamble from one submission to another with definitions for a format that I elaborated and liked. So at some point I collected the main definitions into this class file.

The phfextendedabstract class sets up the document for an extended abstract submission as a single-column document, typeset by default at 11 point size, with at most two sectioning levels (`\section`s and `\paragraph`s). It is based on the RevTeX class.

The extended abstract style looks approximately like this:


Title of my extended abstract

Author One, Author Two, and Author Three

- **Introduction.** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
- **Results.** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.
  - First result. ...
  - Second result. ...
- **Discussion.** Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna.

2 Main features

Here are the main features of the phfextendedabstract class:

- The class phfextendedabstract loads the revtex4-2 class, so you can directly use REVTEX's features such as author affiliations, etc.
- Only two sectioning commands are enabled: `\section` and `\paragraph`. Both have run-in headings. If you find yourself needing additional sectioning levels, it might be that your extended abstract is too detailed and you might want to think about how to keep it at a higher level. It might also be that phfextendedabstract is too limited for your needs.

  By default, section and paragraphs have "decorations" (by default a symbol in the margin) to guide the reader through the overall high-level structure of the document. (Disable them with the `noheadingdecorations` class option.)
- You can easily scale all vertical spacing dimensions (section and paragraph spacing, vertical space around theorems and list environments, etc.) by a common factor with a class option. Do you have those extra two lines that make you exceed your 3-page limit? Try squeezing everything together with a class option like `compressverticalspacing=0.7`.
- The phfnote package is loaded in order to provide a set of default LATEX packages and set up hyperlinks. A generous default set of standard LaTeX packages are loaded, including caption and enumitem; see phfnote's documentation for the option `pkgset=extended`.

  You are expected to include `\usepackage{hyperref}` somewhere in your preamble. We deliberately don't include hyperref when loading the class in order to give you greater control of package loading order (most packages you might want to use must be loaded before hyperref).

- The page margins are tweaked with the geometry package. (Simply call `\geometry{...}` if you'd like to further change them. Refer to the geometry package's documentation.)
- Lists, i.e. the `itemize` and `enumerate` environments, are customized using the enumitem package so that they take up less space. You also get a `enumerate*` environment that typesets its items in-line, in a single paragraph.

  (Note that this customization won't work if you choose to load a package set via a `pkgset=` option that doesn't include enumitem.)
- By default we load the phfthm package and set a customized theorem style so that it stands out but also so that it contrasts well with the section and paragraph headings.

Check out the class option documentation in section 4 to get a better idea of how to customize phfextendedabstract's behavior!

■ 3 Example Usage

Here's a simple template:

```
\documentclass[papertype=a4paper]{phfextendedabstract}

\usepackage{hyperref}

\begin{document}
\title{Title goes here}
\author{Author 1}
\author{Author 2}
\maketitle

\section{Introduction.}
Lorem ipsum ...

\section{Results.}
We had some cool results about
\begin{enumerate*} % inline enumeration
\item a first result,
\item a second result, % and
\item a final result.
\end{enumerate*}

\paragraph{First result.}
Here's a first result ...

\paragraph{Second result.}
Here's another result ...
```

```
...

\end{document}
```

(Try it out!)

4 Class options

Here is a summary of class options:

`papertype=a4paper`, `letterpaper`, ⟨*paper type*⟩, ⟨*empty*⟩

Specify the paper type to use (A4 or letter). The argument given to this option is directly specified as an option to the underlying RevTeX class.

`ptsize=10pt`, `11pt`, `12pt`

Default font size to use. Again the argument given to this option is directly specified as an option to the underlying RevTeX class.

`sectiondecorations=`⟨*true or false*⟩

Whether or not to "decorate" section headings, by default with a small black square typeset in the margin of the section heading.

`paragraphdecorations=`⟨*true or false*⟩

Whether or not to "decorate" paragraph headings, by default with a small right-pointing outlined triangle typeset in the margin of the paragraph heading.

`noheadingdecorations`

Shorthand for `sectiondecorations=false,paragraphdecorations=false`.

`compressverticalspacing=`⟨*factor*⟩

Real factor by which to multiply the vertical spacing between sections, paragraphs, theorems, paragraphs, list environments such as `itemize` and `enumerate`, etc. If you need to compress the sections a bit to fit more content on a fixed number of pages (e.g. because of page number constraints), you can set this option to a factor that's less than one. A ⟨*factor*⟩ that's less than one compresses sections together, a ⟨*factor*⟩ equal to 1 does nothing, and a ⟨*factor*⟩ greater than one expands the sections further apart.

`loadtheorems=`⟨*true or false*⟩

If `loadtheorems=true` (the default), then the phfthm package is loaded with some suitable default options and a custom theorem style. (The theorem style `phfextendedabstractthm` is defined regardless of this package option.)

`sansstyle=`⟨*true or false*⟩

Use sans serif style by default for the main title as well as for section and paragraph headings. For greater degree of control, see subsection 5.4.

`usehyperref=`⟨*true or false*⟩

Should we set up the document for use with the hyperref package or not? This option influences how we load the phfnote package. If this option is set to `true` (the default), then the phfnote package is loaded with `hyperrefdefs={defer,noemail}`. This means that the document is set up for use with hyperref, although you will still need to say `\usepackage{hyperref}` somewhere in your preamble. If `usehyperref=false`, then the phfnote package is loaded with hyperref options disabled.

(The `noemail` option is specified to phfnote to avoid interference with REVTEX's own `\email` command used to specify email addresses for individual authors.)

`pkgset=`⟨*phfnote package set name*⟩

The phfnote package (which we load internally) loads a bunch of standard packages for your convenience, such as enumitem or microtype. You can influence this behavior by specifying a "package set" to load. By default, the `pkgset=extended` package set is loaded. If you don't want to load any additional packages beyond those that are strictly necessary, use `pkgset=none` or `pkgset=minimal`. See phfnote's documentation for the package option `pkgset=...` for more information and a list of possible package set names.

5 Macros and environments

5.1 Sections and paragraphs

`\section` `\paragraph`

This class only provides two sectioning levels: `\section` and `\paragraph`. Both produce run-in headings.

By default, sections and paragraphs produce a small "decoration" in the left margin, to guide the reader's eye through the document's high-level structure.

You can disable/enable section and paragraph decorations with the `sectiondecorations`, `paragraphdecorations` and `noheadingdecorations` class options.

The `\section` and `\paragraph` macros also accept a special syntax: If they are immediately followed by an exclamation mark (`!`), then they do not produce the corresponding decoration. E.g.:

```
\section!{A section heading without its decoration}
```

Also, section and paragraphs will not produce any horizontal space if the title is empty. You can use this feature to produce the spacing and decoration of a section, but not the heading:

```
\section{} This line has the vertical spacing and decoration
associated with a section, but with no run-in heading.

\section!{} This line has the vertical spacing associated
with a section, but with no decoration or run-in heading.
```

You can also use the additional syntax <...> to specify a custom decoration instead of the default one. (But you should only use this sparingly! Redefine `\phfeaSectionDecorationSymbol` if you want to change the symbol for all sections!) For instance:

```
\section<\guilsinglright>{} \emph{Manuscript in preparation.}
```

Here's a summary of the argument structure for `\section` and `\paragraph`:

`\section`|`\paragraph` *[* `*` *] [* `!` *] [* <⟨*decoration*⟩> *] [* [⟨*alt title*⟩] *]* {⟨*title*⟩}

The optional star and optional alternative title are passed on to the standard LaTeX sectioning commands. The alternative title is typically used for the table of contents and for PDF bookmarks.

`\phfeaSectionDecoration` `\phfeaParagraphDecoration`

You can use the `\phfeaSectionDecoration{<symbol>}` and `\phfeaParagraphDecoration{<symbol>}` to manually place a section or paragraph decoration at any place:

```
\phfeaSectionDecoration{$\to$} Check out our website at
\href{https://github.com/phfaist/phfqitltx}{github.com/phfaist/phfqitltx}
```

If the `sectiondecorations=false` class option was specified, the macro `\phfeaSectionDecoration{...}` produces no output. Similarly, `\phfeaParagraphDecoration{...}` does nothing if `paragraphdecoration=false` was specified.

`\phfeaSectionDecorationSymbol` `\phfeaParagraphDecorationSymbol`

The default section and paragraph symbols (including size and vertical adjustments) are stored in the `\phfeaSectionDecorationSymbol` and `\phfeaParagraphDecorationSymbol` macros. You can redefine these macros to set custom decoration symbols:

```
\renewcommand{\phfeaSectionDecorationSymbol}{\Large$\Rightarrow$}
\renewcommand{\phfeaParagraphDecorationSymbol}{\raisebox{1pt}{\tiny $>$}}
```

You can customize section and paragraph headings, including spacing and style, by redefining the following macros.

\phfeaSectionBeforeSkip
\phfeaSectionAfterHSkip

The macro \phfeaSectionBeforeSkip (it's a macro, not a length) is used to specify the vertical spacing before a new section heading. The macro (not length) \phfeaSectionAfterHSkip is the horizontal space between the end of the section heading and the beginning of the section text contents. You can redefine these macros to set your custom spacings. (If you're looking to compress the vertical spacings to save on the number of pages, look at the compressverticalspacings= class option and the \phfeaVerticalSpacingCompressionFactor macro.)

\phfeaParagraphBeforeSkip
\phfeaParagraphAfterHSkip

The macros \phfeaParagraphBeforeSkip and \phfeaParagraphAfterHSkip behave similarly for paragraphs.

\phfeaSectionStyle
\phfeaParagraphStyle

The macros \phfeaSectionStyle and \phfeaParagraphStyle are used to set the style of the section and paragraph headings. By default, the default sans/heading style is used at the normal size and in bold face series for sections, and the normal sans style is used at a smaller size for paragraph headings.

\phfeaSectionFormatHeading
\phfeaParagraphFormatHeading

The macros \phfeaSectionFormatHeading{...} and \phfeaParagraphFormatHeading{...} can be redefined to format the section and paragraph headings, respectively. You can redefine these macros to add punctuation, to capitalize the title, etc. For instance:

```
\renewcommand\phfeaSectionFormatHeading[1]{\MakeUppercase{#1}\ ---}
```

The macros \phfeaSectionFormatHeading{...} and \phfeaParagraphFormatHeading{...} are not called if the section/paragraph title argument is empty.

5.2 Vertical spacing

\phfeaVerticalSpacingCompressionFactor

To conveniently globally adjust the vertical spacings in the document (including the section and paragraph vertical spacings, as well as the spacing above and below theorems, itemize, and enumerate environments), you can also use the compressverticalspacing=X class option. Alternatively, you can redefine the macro \phfeaVerticalSpacingCompressionFactor to the desired compression factor:

```
\renewcommand\phfeaVerticalSpacingCompressionFactor{0.7}
```

\phfeaDefineTheoremStyle

Because the theorem style phfextendedabstracthm is defined at the time that the class is loaded, any customization of \phfeaVerticalSpacingCompressionFactor and \phfeaListsVerticalSkip that happen later in the preamble aren't taken into account. If you change these values in your preamble, you should

call \phfeaDefineTheoremStyle to redefine the theorem style after your customization.

\phfeaDisplayVerticalSpacingFactorWeight

The vertical spacing factor also affects the vertical spacing around equations, but to a lesser extent. (Compressing the space around the equations by too much would not look nice.) For the spacing between equations, we take the weighted average of 1 and the vertical spacing compression factor, where the weight is given in the macro \phfeaDisplayVerticalSpacingFactorWeight. A weight of 0 means the vertical compression factor doesn't affect the vertical spacing around equations at all; a weight of 1 means the spacing around the equations is scaled exactly by the vertical compression factor.

\phfeaParskipVerticalSpacingFactorWeight

A similar mechanism happens for how we adjust \parskip, the spacing between paragraphs.

5.3 List environments

enumerate*

This class provides an enumerate-like environment which typesets its items inline, as a list. For example, here is an inline paragraph with (a) one, (b) two, and (c) three points.

The enumerate* can be used exactly like you'd use the enumerate environment from the enumitem package, for instance:

```
here is an inline paragraph with \begin{enumerate*}[label=(\alph*)]
\item one,
\item two,
\item three
\end{enumerate*}
points.
```

Check enumitem's documentation for inline lists. You can specify for instance the keys before={{}}, itemjoin={{ }}, and itemjoin*={{ and }} either as an optional argument to \begin{enumerate*} or using \setlist:

```
\setlist[enumerate*]{%
  itemjoin*={{ et }}%
}
```

\phfeaListsVerticalSkip
\phfeaListsItemSep
\phfeaListsParSep

For (non-inline) list environments such as itemize and enumerate, you can redefine \phfeaListsVerticalSkip, \phfeaListsItemSep, and \phfeaListsParSep to set the topsep, itemsep and parsep properties of all of enumitem's list environments. These correspond to the vertical space above and below lists, the space between items, and the space between paragraphs within an item. All these spacings automatically get compressed according to the \phfeaVerticalSpacingCompressionFactor.

Note that \phfeaListsVerticalSkip is also used for the spacing above and below theorem environments.

5.4 Customization of the main title and general headings style

\phfeaHeadingStyle The macro \phfeaHeadingStyle is defined to be \sffamily, unless the sansstyle=false class option is provided in which case the macro expands to nothing. You can redefine it to give a different general style to your main title and your section and paragraph headings.

\phfeaTitleStyle The macro \phfeaTitleStyle sets the font style for the main document title. By default the macro is defined to \phfeaHeadingStyle\Large. Redefine this macro to change the title style.

For instance, if you prefer REVTEX's own simple boldface title, you can use:

```
\renewcommand\phfeaTitleStyle{\bfseries}
```

■ 6 Implementation

Here come the gory details.

Class options

We process these first, to see which options we should pass on to REVTEX.

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{%
  family=phfea,%
  prefix=phfeaopt@%
}
\DeclareStringOption[]{papertype}
\DeclareStringOption[11pt]{ptsize}
\DeclareBoolOption[true]{sectiondecorations}
\DeclareBoolOption[true]{paragraphdecorations}
\DeclareVoidOption{noheadingdecorations}{%
  \phfeaopt@sectiondecorationsfalse
  \phfeaopt@paragraphdecorationsfalse
}
\DeclareBoolOption[true]{loadtheorems}
\DeclareBoolOption[true]{sansstyle}
\DeclareStringOption[1]{compressverticalspacing}
\DeclareBoolOption[true]{usehyperref}
\DeclareStringOption[extended]{pkgset}
\ProcessKeyvalOptions*
```

Load REVTEX, the base class

```
\providecommand\phfea@revtexopts{%
  aps,pra,%
```

```
  notitlepage,reprint,%
  onecolumn,tightenlines,%
  superscriptaddress,%
  nofootinbib%
}
\PassOptionsToClass{%
  \phfea@revtexopts,%
  \phfeaopt@ptsize,%
  \phfeaopt@papertype,%
}{revtex4-2}
\LoadClass{revtex4-2}
```

We will also need the xparse command for defining some of our commands.

```
\RequirePackage{xparse}
```

Load phfnote for the basic document setup

```
\PassOptionsToPackage{%
  preset=reset,%
  pkgset=\phfeaopt@pkgset,%
  \ifphfeaopt@usehyperref
    hyperrefdefs={defer,noemail},%
  \fi
}{phfnote}
\RequirePackage{phfnote}
```

Page geometry

Set a default page geometry. Works both for A4 paper and for letter paper. It's optimized for 11pt size, though.

```
\RequirePackage{geometry}
\geometry{hmargin=0.75in,vmargin=0.75in,marginparwidth=0.5in,marginparsep=0.125in}
```

Default sans serif font

`\phfeaHeadingStyle` Unless instructed not to do so, set the sans serif font family to be used for section headings and the main title.

```
\ifphfeaopt@sansstyle
  \def\phfeaHeadingStyle{\sffamily}
\else
  \def\phfeaHeadingStyle{}
\fi
```

Default title format

\phfeaTitleStyle Change REVTEX title format. Title style can be customized by redefining \phfeaTitleStyle.

```
\def\phfeaTitleStyle{\phfeaHeadingStyle\Large}
\def\frontmatter@title@format{\phfeaTitleStyle\centering\parskip\z@skip}
```

Vertical spacing compression factor

Define a general factor by which the section and paragraph spacings will be compressed. This macro is set by the compressverticalspacing=X package option.

```
\edef\phfeaVerticalSpacingCompressionFactor{\phfeaopt@compressverticalspacing}
```

Tool for scaling glue expressions (for use with our vertical compression factor):[1]

```
\def\phfea@scaleglue#1#2{% {factor}{glueexpr}
  \glueexpr#2*\numexpr\dimexpr#1pt\relax\relax/65536\relax
}
\def\phfea@scalegluedpt#1#2{% {factor given as dimexpr in pt}{glueexpr}
  \glueexpr#2*\numexpr#1\relax/65536\relax
}
```

Adjust spacing around display equations according to the vertical compression factor. Do this only at the beginning of the document, since the user might still want to adjust \phfeaVerticalSpacingCompressionFactor in their preamble.

\phfeaDisplayVerticalSpacingFactorWeight We only apply a fraction of the scaling, because it's ugly if we compress equations too much. Define \phfeaDisplayVerticalSpacingFactorWeight as w and α as the vertical scaling factor. The new skips are computed as

$$\texttt{oldskip} \quad \to \quad (1-w)\,\texttt{oldskip} + w\,\alpha\,\texttt{oldskip}\,.$$

(For $w = 0$ we have oldskip → oldskip. For $w = 1$ the full scaling factor is applied, oldskip → α oldskip.) By default, $w = 1/2$:

```
\def\phfeaDisplayVerticalSpacingFactorWeight{.5}
```

Tool to compute the new spacing for each of the relevant display-related skips:

```
\def\phfea@adjustskipweighted#1#2{%
  #1=\glueexpr
    \phfea@scalegluedpt{%
      \dimexpr 1\p@ - #2\p@\relax
    }{#1}%
```

[1] Thanks *https://tex.stackexchange.com/a/198966/32188*!

```
    +
    \phfea@scaleglue{%
      #2%
    }{%
      \phfea@scaleglue{%
        \phfeaVerticalSpacingCompressionFactor
      }{%
        #1
      }%
    }%
    \relax
}
```

And schedule this adjustment to be carried out at the beginning of the document.

```
\AtBeginDocument{%
  \phfea@adjustskipweighted\abovedisplayskip\phfeaDisplayVerticalSpacingFactorWeight
  \phfea@adjustskipweighted\belowdisplayskip\phfeaDisplayVerticalSpacingFactorWeight
  \phfea@adjustskipweighted\abovedisplayshortskip\phfeaDisplayVerticalSpacingFactorWeight
  \phfea@adjustskipweighted\belowdisplayshortskip\phfeaDisplayVerticalSpacingFactorWeight
}
```

\phfeaParskipVerticalSpacingFactorWeight

A similar mechanism affects how we adjust the paragraph skip length \parskip.

```
\def\phfeaParskipVerticalSpacingFactorWeight{1}
\AtBeginDocument{%
  \phfea@adjustskipweighted\parskip\phfeaParskipVerticalSpacingFactorWeight
}
```

Setup specific for sectioning

By design, there are only two sectioning levels in a phfextendedabstract document: sections (\section) and paragraphs (\paragraph). Any other sectioning command (e.g., \subsection) will produce an error.

Neither of these two section levels is numbered.

```
\setcounter{secnumdepth}{0}
\setcounter{tocdepth}{1}
```

\phfeaSectionBeforeSkip
\phfeaSectionAfterHSkip
\phfeaSectionStyle
\phfeaSectionDecoration
\phfeaSectionDecorationSymbol

Some helper and customization macros for \section.

```
\def\phfeaSectionBeforeSkip{1.5ex plus 0.8ex minus 0.25ex}
\def\phfeaSectionAfterHSkip{1em plus 0.2em}
\def\phfeaSectionStyle{\normalfont\normalsize\phfeaHeadingStyle\bfseries}
\def\phfeaSectionFormatHeading#1{#1}
\def\phfeaSectionDecorationSymbol{%
  \raisebox{0.2ex}{{\notesmaller[0.4]{\ensuremath{\blacksquare}}}}}
\ifphfeaopt@sectiondecorations
  \def\phfeaSectionDecoration#1{%
```

```
    \makebox[\z@][r]{{#1}\hspace*{1.5ex}}%
  }
\else
  \def\phfeaSectionDecoration#1{}
\fi
```

\phfeaParagraphBeforeSkip
\phfeaParagraphAfterHSkip
\phfeaParagraphStyle
\phfeaParagraphDecoration
\phfeaParagraphDecorationSymbol

Same helper and customization macros, now for \paragraph.

```
\def\phfeaParagraphBeforeSkip{0.6ex plus 0.4ex minus 0.1ex}
\def\phfeaParagraphAfterHSkip{0.75em plus 0.15em}
\def\phfeaParagraphStyle{\normalfont\normalsize\phfeaHeadingStyle\small}
\def\phfeaParagraphFormatHeading#1{#1}
\def\phfeaParagraphDecorationSymbol{%
  \raisebox{0.2ex}{{\notesmaller[0.6]{\ensuremath{\triangleright}}}}}
\ifphfeaopt@paragraphdecorations
  \def\phfeaParagraphDecoration#1{%
    \makebox[\z@][r]{{#1}\hspace*{1ex}}%
  }
\else
  \def\phfeaParagraphDecoration#1{}
\fi
```

Now we redefine \section and \paragraph for formatting. Also allow \section! and \paragraph! that remove the decoration and \section<...> and \paragraph<...> that replace the decoration. In all cases, the indent is removed if the main argument (section/paragraph title) is an empty argument. We also accept a starred version and an optional argument, and pass those on to \@startsection; these options are sometimes useful in order to work around fragile tokens in the title, etc.

We start with the generic code that will call \@startsection for both sections and paragraphs. The generic code will be given the level number (1 or 2) and will produce the correct \@startsection code for that level, fetching its information from the correctly named macros. We first define some simple definitions that specify where to look for those level-specific macros.

```
\def\phfea@ss@levelname#1{\ifcase#1\or section\or paragraph\fi}
\def\phfea@ss@levelName#1{\ifcase#1\or Section\or Paragraph\fi}
\def\phfea@ss@get#1#2{% <levelno><MacroPostfixName>
  \csname phfea\phfea@ss@levelName{#1}#2\endcsname}
\def\phfea@ss@getnoexpand#1#2{% <levelno><MacroPostfixName>
  \expandafter\noexpand\csname phfea\phfea@ss@levelName{#1}#2\endcsname}
```

\phfea@startsection

We now define the main internal \phfea@startsection macro which will produce the correct \@startsection command for the given section level. The arguments to \phfea@startsection are:

#1: level number, 1 or 2;

#2: an optional star;

#3: an optional `!` character (meaning, don't display the decoration);

#4: an optional `<`⟨*decoration tokens*⟩`>` argument to replace the decoration;

#5: an optional argument, the alternative title to write in the AUX file (e.g. for use in the table of contents);

#6: a mandatory argument, the title of the section/paragraph.

```
\newtoks\phfea@ss@decorationtoks
\newtoks\phfea@ss@alttitle
\newtoks\phfea@ss@title
\NewDocumentCommand{\phfea@startsection}{m s t! d<> o m}{%
```

Store the alternative title and title in token registers.

```
  \phfea@ss@decorationtoks={#4}%
  \phfea@ss@alttitle={#5}%
  \phfea@ss@title={#6}%
```

Inform the user in case of option inconsistencies or redundancies.

```
  \IfNoValueF{#4}{%
    \IfBooleanT{#3}{%
      \PackageWarning{phfextendedabstract}{section/paragraph: `!'
        modifier ignored when custom decoration `<...>' is specified}%
    }%
  }%
```

Now define a temporary macro `\x` that will contain the correct code for a call to LaTeX' low-level `\@startsection` command. Remember to prefix all macros by `\noexpand` unless they should be expanded in the preparation of that code for `\@startsection`. LaTeX' `\@startsection` takes a lot of arguments, let's be careful not to miss any. Also, we can use the convenient construct `\phfea@ss@getnoexpand{#1}{XyzProperty}` which in this `\edef` will expand to `\noexpand\phfea`⟨`Section`|`Paragraph`⟩`XyzProperty`.

```
  \edef\x{%
    \noexpand\@startsection{\phfea@ss@levelname{#1}}%
```

First come the ⟨*level number*⟩ and ⟨*indent*⟩ arguments:

```
      {#1}%
      {\z@}%
```

Then comes the ⟨*before skip*⟩ argument. Use `\phfea@scaleglue` to account for the vertical compression factor.

```
      {%
        \noexpand\phfea@scaleglue{\noexpand\phfeaVerticalSpacingCompressionFactor}{%
          \noexpand\glueexpr\phfea@ss@getnoexpand{#1}{BeforeSkip}\noexpand\relax
        }%
      }%
```

Then comes the ⟨*after skip*⟩ argument. Set the after skip to zero if the title is empty, otherwise specify the given `\phfeaSection/ParagraphAfterHSkip` as a negative value to indicate we want a run-in heading. Note that while the syntax '`1.5\glueexpr...`' doesn't preserve plus/minus stretchability components as it coerces the glue to a dimen, the syntax '`-\glueexpr...`' does preserve these components (presumably because −1 is integer?).

```
{\if\relax\detokenize{#6}\relax
    \noexpand\z@
  \else
    -\noexpand\glueexpr\phfea@ss@getnoexpand{#1}{AfterHSkip}\noexpand\relax
  \fi}%
```

The ⟨*style*⟩ argument to specify the section heading's font style. We provide a decoration, possibly a custom decoration if provided with the `<...>` syntax; we remove the default decoration if the no-decoration argument (a `!` character) was specified.

```
{%
  \IfNoValueTF{#4}{%
    \IfBooleanTF{#3}{% with "!", no custom decoration --> nothing
    }{% no "!", no custom decoration
      \phfea@ss@getnoexpand{#1}{Decoration}%
      \phfea@ss@getnoexpand{#1}{DecorationSymbol}%
    }%
  }{% with custom decoration (ignores "!" argument)
    \phfea@ss@getnoexpand{#1}{Decoration}%
    {\the\phfea@ss@decorationtoks}%
  }%
  \phfea@ss@getnoexpand{#1}{Style}%
}%
```

An optional `*` indicates the starred version of the sectioning commands. If the user increased the `secnumcounter` to have numbered sections, then the starred variant gives a section heading without any section number or TOC entry.

```
\IfBooleanT{#2}{*}%
```

Now comes the optional ⟨*alternative title*⟩ argument that will be written to the AUX file. We always provide this argument to `\@startsection`, specifying by default the main title argument `#6` (non-starred variant) or an empty argument (for a starred variant). We do this because otherwise, `\@startsection` will copy all the other commands we add in its argument (e.g., title formatting, etc.) and will write them to the AUX file.

```
\IfValueTF{#4}{[\the\phfea@ss@alttitle]}{%
  \IfBooleanTF{#2}{[]}{[\the\phfea@ss@title]}%
}%
```

Finally, this is the main ⟨*section title*⟩ argument. If the title is not empty, we also provide the title wrapped in a corresponding

\phfeaSection/ParagraphFormatHeading macro call. Everything should be protected in \texorpdfstring calls to avoid *pdflatex* from complaining about invalid junk in PDF bookmark strings.

```
      {%
        \if\relax\detokenize{#6}\relax\else
          \noexpand\texorpdfstring{%
            \phfea@ss@getnoexpand{#1}{FormatHeading}{\the\phfea@ss@title}%
          }{%
            \the\phfea@ss@title
          }%
        \fi
      }%
  }%
  %\message{**** EMITTING @startsection: |\detokenize\expandafter{\x}| ***}%
  \x
}
```

\section
\paragraph

Define \section and \paragraph as sectioning levels 1 and 2.

```
\def\section{\phfea@startsection{1}}
\def\paragraph{\phfea@startsection{2}}
```

Provide an explanative error message if the user attempts to use \subsection, \subsubsection or \subparagraph.

```
\def\phfea@nosectioncmd#1{%
  \ClassError{phfextendedabstract}{%
    %
    There is no `\string#1' command in `phfextendedabstract'
    documents.  You can only use `\string\section' and
    `\string\paragraph'.  If you find yourself needing additional
    sectioning levels, it might be that your extended abstract is too
    detailed and you should stick to a higher level description of
    your results.  If you do need additional section levels, it is
    likely that you will be better off with a different document
    class.  If you'd like to stick with `phfextendedabstract' and you
    really know what you're doing, then you could redefine the
    sectioning commands as necessary based on `\string\@startsection'
    as is done in standard LaTeX classes.  Good luck!
    %
  }{}%
}
\def\phfea@nosectioncmd@def#1{%
  \def#1{\phfea@nosectioncmd#1}%
}
\phfea@nosectioncmd@def\part
\phfea@nosectioncmd@def\chapter
\phfea@nosectioncmd@def\subsection
\phfea@nosectioncmd@def\subsubsection
\phfea@nosectioncmd@def\subparagraph
```

Set up itemization and enumeration environments

Provide customizable lengths for lists via macros (item sep, paragraph sep and vertical skip above and below list environments).

\phfeaListsVerticalSkip
\phfeaListsItemSep
\phfeaListsParSep

```
\def\phfeaListsVerticalSkip{0.6ex plus 0.4ex minus 0.1ex}
\def\phfeaListsItemSep{0.3ex plus 0.15ex minus 0.1ex}
\def\phfeaListsParSep{0.7\parskip}
```

Prepare the commands to run to configure enumitem correctly in an internal macro which we will call if enumitem is indeed loaded.

```
\def\phfea@setup@enumitem{%
```

Apply the spacings.

```
  \setlist{%
    itemsep={%
      \phfea@scaleglue{%
        \phfeaVerticalSpacingCompressionFactor}{\phfeaListsItemSep}},
    parsep={%
      \phfea@scaleglue{%
        \phfeaVerticalSpacingCompressionFactor}{\phfeaListsParSep}},
    topsep={%
      \phfea@scaleglue{%
        \phfeaVerticalSpacingCompressionFactor}{\phfeaListsVerticalSkip}},
  }%
```

enumerate* Create the enumerate* list enumeration environment.

```
  \newlist{enumerate*}{enumerate*}{1}
  \setlist[enumerate*]{
    label={(\roman*)},
    before={},
    itemjoin={{ }},
    itemjoin*={{ and }}
  }%
}
```

And now, check if enumitem is loaded and apply the definitions.

```
\@ifpackageloaded{enumitem}{\phfea@setup@enumitem}{}
```

Setup specific for theorems

Create a new theorem style called extendedabstracthm. Note that if amsthm (or similar) wasn't loaded, then \newtheoremstyle isn't defined. In that case, we simply won't define the new theorem style right now.

```
\def\phfeaDefineTheoremStyle{%
  \newtheoremstyle{phfextendedabstractthm}%
    {\phfea@scaleglue{%
        \phfeaVerticalSpacingCompressionFactor}{\phfeaListsVerticalSkip}}%
    {\phfea@scaleglue{%
        \phfeaVerticalSpacingCompressionFactor}{\phfeaListsVerticalSkip}}%
    {\itshape}%
    {\z@}%
    {\bfseries\itshape}%
    {:}%
    {0.8em}%
    {\thmname{##1\thmnumber{ ##2}\thmnote{ (##3)}}}%
}
```

Ensure we create our new theorem style (only if `\newtheoremstyle` is available) and load our phfthm package. Make sure we don't load phfthm if we were asked not to load theorems in the class options.

```
\ifdefined\newtheoremstyle
  \phfeaDefineTheoremStyle
\fi
\ifphfeaopt@loadtheorems
  \ifdefined\newtheoremstyle\else
    \ClassError{phfextendedabstract}{%
      %
      Impossible to load theorems (‘loadtheorems=true’) because there
      is no \string\newtheoremstyle\space command that was defined.
      Consider setting ‘pkgset=...’ so that a theroems-related package
      (e.g., amsthm) is loaded!  (e.g. ‘pkgset=minimal’, ‘pkgset=rich’
      or ‘pkgset=extended’).  Alternatively, set the class option
      ‘loadtheorems=false’ and you can then manually define any
      theorem environments you’d like using your favorite pacakges.
      %
    }{}
  \fi
  \PassOptionsToPackage{proofref=false,theoremstyle=phfextendedabstractthm}{phfthm}
  \RequirePackage{phfthm}
\fi
```

Change History

v1.0

General: Initial version 1

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

\@ifpackageloaded 227
\@startsection 134, 191
10pt (pkg. opt.) 4
11pt (pkg. opt.) 4
12pt (pkg. opt.) 4

A

a4paper (pkg. opt.) 4
\abovedisplayshortskip 79
\abovedisplayskip 77
amsthm 17
\AtBeginDocument 76, 83

B

\belowdisplayshortskip 80
\belowdisplayskip 78
\bfseries 90, 236
\blacksquare 93

C

caption 2
\centering 50
\chapter 200
\ClassError 180, 246
compressverticalspacing (pkg. opt.) 4
\csname 117, 119

D

\DeclareBoolOption ... 8, 9, 14, 15, 17
\DeclareStringOption ... 6, 7, 16, 18
\DeclareVoidOption 10
\detokenize 142, 165, 174
\dimexpr 53, 62

E

\endcsname 117, 119
\ensuremath 93, 106
enumerate* (environment) *8*, 219
enumitem 2, 3, 5, 8, 17
environments:
  enumerate* *8*, 219
\expandafter 119, 174

F

\frontmatter@title@format 50

G

geometry 3
\geometry 43
\glueexpr 53, 56, 60, 139, 145

H

\hspace 96, 109
hyperref 2, 5

I

\IfBooleanT 128, 160
\IfBooleanTF 149, 162
\ifcase 114, 115
\ifdefined 241, 245
\IfNoValueF 127
\IfNoValueTF 148
\ifphfeaopt@loadtheorems 244
\ifphfeaopt@paragraphdecorations 107
\ifphfeaopt@sansstyle 44
\ifphfeaopt@sectiondecorations 94
\ifphfeaopt@usehyperref 37
\IfValueTF 161
\itshape 234, 236

L

\Large 49
letterpaper (pkg. opt.) 4
\LoadClass 32
loadtheorems (pkg. opt.) 4

M

\makebox 96, 109
\message 174
microtype 5

N

\NewDocumentCommand 123
\newlist 219
\newtheoremstyle .. 229, 241, 245, 249
\newtoks 120, 121, 122

\noexpand 119, 134, 138, 139, 143, 145, 166
noheadingdecorations (pkg. opt.) . . 4
\normalfont 90, 103
\normalsize 90, 103
\notesmaller 93, 106
\numexpr 53, 56

O

\or 114, 115

P

\p@ 62
package options:
10pt 4
11pt 4
12pt 4
a4paper 4
compressverticalspacing 4
letterpaper 4
loadtheorems 4
noheadingdecorations 4
papertype 4
paragraphdecorations 4
pkgset 5
ptsize 4
sansstyle 5
sectiondecorations 4
usehyperref 5
packages:
amsthm 17
caption 2
enumitem 2, 3, 5, 8, 17
geometry 3
hyperref 2, 5
microtype 5
phfextendedabstract 1–3, 12
phfnote 2, 5, 10
phfqitltx 1
phfthm 3, 4, 18
revtex4-2 2
xparse 10
\PackageWarning 129
papertype (pkg. opt.) 4
\paragraph 5, 177, 184
paragraphdecorations (pkg. opt.) . . 4
\parskip 50, 84, 206
\part 199
\PassOptionsToClass 27
\PassOptionsToPackage 34, 258
\phfea@adjustskipweighted . . . .
. . . . . 59, 77, 78, 79, 80, 84
\phfea@nosectioncmd 179, 197
\phfea@nosectioncmd@def . . . . . .
. . . . . 196, 199, 200, 201, 202, 203
\phfea@revtexopts 20, 28
\phfea@scaleglue 52,
65, 68, 138, 210, 213, 216, 230, 232
\phfea@scalegluedpt 55, 61
\phfea@setup@enumitem 207, 227
\phfea@ss@alttitle . . . 121, 125, 161
\phfea@ss@decorationtoks . . . . .
. . . . . 120, 124, 156
\phfea@ss@get 116
\phfea@ss@getnoexpand 118,
139, 145, 151, 152, 155, 158, 167
\phfea@ss@levelName . . 115, 117, 119
\phfea@ss@levelname 114, 134
\phfea@ss@title 122, 126, 162, 167, 169
\phfea@startsection . . 120, 177, 178
\phfeaDefineTheoremStyle 7, 228, 242
\phfeaDisplayVerticalSpacingFactorWeight
. . . . . 8, 58, 77, 78, 79, 80
\phfeaHeadingStyle . 9, 44, 49, 90, 103
\phfeaListsItemSep 8, 204, 211
\phfeaListsParSep 8, 204, 214
\phfeaListsVerticalSkip . . . . . .
. . . . . 8, 204, 217, 231, 233
\phfeaopt@compressverticalspacing
. 51
\phfeaopt@papertype 30
\phfeaopt@paragraphdecorationsfalse
. 12
\phfeaopt@pkgset 36
\phfeaopt@ptsize 29
\phfeaopt@sectiondecorationsfalse
. 11
\phfeaParagraphAfterHSkip . . 7, 101
\phfeaParagraphBeforeSkip . . 7, 101
\phfeaParagraphDecoration . . 6, 101
\phfeaParagraphDecorationSymbol
. . . . . 6, 101
\phfeaParagraphFormatHeading 7, 104
\phfeaParagraphStyle 7, 101
\phfeaParskipVerticalSpacingFactorWeight
. . . . . 8, 82
\phfeaSectionAfterHSkip 7, 88
\phfeaSectionBeforeSkip 7, 88
\phfeaSectionDecoration 6, 88
\phfeaSectionDecorationSymbol
. . . . . 6, 88
\phfeaSectionFormatHeading . . 7, 91

\phfeaSectionStyle 7, 88
\phfeaTitleStyle 9, 49
\phfeaVerticalSpacingCompressionFactor 7, 51, 69, 138, 211, 214, 217, 231, 233
phfextendedabstract 1–3, 12
phfnote 2, 5, 10
phfqitltx 1
phfthm 3, 4, 18
pkgset (pkg. opt.) 5
\ProcessKeyvalOptions 19
\providecommand 20
ptsize (pkg. opt.) 4

R

\raisebox 93, 106
\relax . 53, 56, 62, 74, 139, 142, 145, 165
\RequirePackage 1, 33, 41, 42, 259
revtex4-2 2
\roman 221

S

sansstyle (pkg. opt.) 5
\section 5, 177, 183
sectiondecorations (pkg. opt.) 4
\setcounter 86, 87
\setlist 208, 220
\SetupKeyvalOptions 2
\sffamily 45
\small 103
\space 249
\string 182, 183, 184, 191, 249
\subparagraph 203
\subsection 201
\subsubsection 202

T

\texorpdfstring 166
\the 156, 161, 162, 167, 169
\thmname 239
\thmnote 239
\thmnumber 239
\triangleright 106

U

usehyperref (pkg. opt.) 5

X

\x 133, 174, 175
xparse 10

Z

\z@ 96, 109, 136, 143, 235
\z@skip 50