

The apxproof package

Pierre Senellart

`pierre@senellart.com`

<http://github.com/PierreSenellart/apxproof>

2025/09/30 v1.2.6

Abstract

This package makes it easier to write articles where proofs and other material are deferred to the appendix. The appendix material is written in the L^AT_EX code along with the main text which it naturally complements, and it is automatically deferred. The package can automatically send proofs to the appendix, can repeat in the appendix the theorem environments stated in the main text, can section the appendix automatically based on the sectioning of the main text, and supports a separate bibliography for the appendix material.

1 Usage

The `apxproof` package is intended to simplify the writing of articles where some of the content needs to be deferred to an appendix. This is in particular useful for the submission of scientific articles to conferences or journals that limit the number of pages in the main text but allow an extra appendix, where proofs of theorems and other material can be added.

1.1 Basics

To use `apxproof`, first load it in the header of your document:

```
\usepackage{apxproof}
```

On its own, this does not do anything and should not change the appearance of your document. To add an appendix with some material from your document, use the `toappendix` environment:

```
toappendix (env.)  \begin{toappendix}
                   ...
                   \end{toappendix}
```

The content will appear at the end of your document, in an automatically generated section that refers to the current section in the main text.

Example 1. *Throughout this documentation, all examples produce content deferred to the appendix, at the very end of this document.*

```
\begin{toappendix}
```

```
This content is in the appendix.
\end{toappendix}
```

When the content to put in appendix is an entire section, make sure that `\section` is the very first command that appears within the `toappendix` environment. It will disable the automatic production of a section heading.

1.2 Repeated Theorems and Proofs

In some scientific papers that include proofs, it is common to defer proofs to the appendix. This can easily be achieved using the `appendixproof` environment:

```
appendixproof (env.)    \begin{appendixproof}
                        ...
                        \end{appendixproof}
```

This behaves like the `toappendix` environment, except that a proof environment is generated.

Example 2. *We now send a proof to the appendix:*

```
\begin{appendixproof}
This proof is in the appendix.
\end{appendixproof}
```

When deferring proofs to the appendix, an annoying problem is that the statement of the theorem remains in the main text; it is hard to read a proof that is far away from the statement it proves. `apxproof` solves this issue by allowing statements of theorems to be *repeated*: once in the main text, and once in the appendix before the proof of the statement. To use this feature, you can define a new *repeated theorem* environment using the `\newtheoremrep` command:

```
\newtheoremrep    \newtheoremrep{<name>}[<counter>]{<title>}[<countersec>]
```

Usage is exactly the same as that of AMS L^AT_EX's `\newtheorem` macro:

- `<name>` (e.g., `theorem`) is the name of an environment that is created for this kind of theorem;
- `<counter>` (e.g., `definition`) is an optional counter describing from which kind of environment the numbering of these environments should be inherited;
- `<title>` (e.g., `Theorem`) is the title that will be used to display this theorem environment;
- `<countersec>` (e.g., `section`) is an optional counter of a sectioning command indicating that counters for this theorem should be prefixed by this counter (and reset at each occurrence of the sectioning command).

`<counter>` and `<countersec>` should not be used together. What differs from `\newtheorem` is that, when the following is written:

```
\newtheoremrep{foobar}{Foobar}
```

then *two* environments are defined: the `foobar` environment, which behaves as if `\newtheorem` had been used, and the `foobarrep` environment, which results in the statement of this environment being repeated in the appendix.

One interesting feature of `apxproof` is that in most situations, there is no need to `proof (env.)` use the `appendixproof` environment. Indeed, the `proof` environment is redefined by `apxproof` to automatically put the proof either in the main text (if it follows a regular theorem) or in the appendix (if it follows a repeated theorem).

Example 3. Assume we have first defined a repeated theorem environment `foobar` as above. We can now use this theorem environment, first for a regular theorem in the main text, then for a theorem repeated in the main text and in the appendix:

```
\begin{foobar}
This foobar is a regular one, in the main text.
\end{foobar}
\begin{proof}
This is the proof of the regular foobar.
\end{proof}
```

We obtain:

Foobar 1. *This foobar is a regular one, in the main text.*

Proof. This is the proof of the regular foobar. □

Now, if we use a repeated theorem:

```
\begin{foobarrep}
This foobar is repeated in the appendix.
\end{foobarrep}
\begin{proof}
This is the proof of the repeated foobar.
\end{proof}
```

We now obtain:

Foobar 2. *This foobar is repeated in the appendix.*

Note that, since `hyperref` is loaded, there are hyperlinks created between the statements of the theorems in the main text and in the appendix.

When the proof is deferred to the appendix, it is common practice to add a proof sketch in the main text. `apxproof` defines a simple `proofsketch` environment for this purpose:

```
proofsketch (env.) \begin{proofsketch}
...
\end{proofsketch}
```

The proof sketch is typeset similarly to a proof, but is always in the main text. `inlineproof (env.)` Similarly, an `inlineproof` environment is provided so as to be able to have both a proof in the appendix (using the regular `proof` environment, or alternatively the `appendixproof` environment) and a different proof in the main text (using the `inlineproof` environment).

Example 4. *Here are simple examples of proof sketches and inline proofs:*

```
\begin{proofsketch}
This is a proof sketch.
\end{proofsketch}
```

Proof sketch. This is a proof sketch. □

```
\begin{inlineproof}
This is an inline proof.
\end{inlineproof}
```

Proof. This is an inline proof. □

1.3 Bibliography

By default, `apxproof` automatically adds a bibliography in the appendix with only the references cited in the appendix material. This allows for a clean separation of references used solely in the main text, and those used in the appendix.

Example 5. *Assume we have citations both in the main text and in the appendix.*

```
This is a citation in the main text~\cite{lamport86}.
\begin{toappendix}
This is a citation in the appendix~\cite{proofsAreHard}.
\end{toappendix}
```

This is a citation in the main text [1].

The bibliography in the appendix can use a different style and heading than the bibliography in the main text (and, by default, it does). See Section 1.5 for how to configure the appearance of that bibliography.

option `bibliography` In order to use a single appendix for the main text and the bibliography, one can specify the value `common` to the `bibliography` option when loading the package. (By default this option is set to `separate`.)

1.4 Mode

option `appendix` An optional `<mode>` can be specified when loading the package:

```
\usepackage[appendix=<mode>]{apxproof}
```

`<mode>` can take one of the following three values:

append This is the default. Appendix material gathered by `apxproof` is appended to the main text.

inline In this mode, `apxproof` simply inlines the content along with the main text.

strip This mode functions similarly to **append** except that the appendix is not appended at the end of the document. All appendix material is therefore removed.

1.5 Customization

`apxproof` provides a few macros that can be redefined (using `\renewcommand`) to customize the appearance of the appendix:

- `\mainbodyrepeatedtheorem` `\mainbodyrepeatedtheorem` is a macro that is executed at the beginning of the body of every repeated theorem. This can be used to notify the reader that the theorem is repeated in appendix in some way, e.g., with a margin note.
 - `\appendixsectionformat` `\appendixsectionformat{<number>}{<title>}` is a macro that indicates how to format the section titles in the Appendix, given the number and title of the section in the main text. By default, they appear as “Proofs for Section *<number>* (*<title>*)”.
 - `\appendixrefname` `\appendixrefname` contains the heading that is displayed before the bibliography. By default, this is “References for the Appendix”. (Note that this command is also defined and used by the `memoir` document class.)
 - `\appendixbibliographystyle` `\appendixbibliographystyle` contains the `.bst` bibliography style that is used in the bibliography in appendix. By default, this is `alpha`.
 - `\appendixbibliographyprelim` `\appendixbibliographyprelim` contains arbitrary code that is executed just before the production of the bibliography in appendix, which can be used to configure the way it is displayed.
 - `\appendixprelim` `\appendixprelim` contains arbitrary code that is executed just before the production of the appendix, which can be used to configure the way it is displayed. By default, this command contains `\clearpage\onecolumn` (the appendix is typeset on a new page in single-column mode) but redefining this option allows changing this behavior.
- option* Another customization capability concerns *numbered equations* that are present within repeated theorems. An optional `repeqn` option can be specified when loading the package, which controls whether equation numbers should be as in the main text (by setting this option to `same`, the default) or independently numbered (by setting this option to `independent`). In the latter case, whenever a referenceable counter is set with `\label{<counter>}`, `\ref{<counter>}` references the counter in the main text, while `\ref{<counter>-apx}` references the counter in the appendix (except in `inline` mode, where both have the same effect).
- option* Another customization option concerns hyperlinking. Usually, when `hyperref` is loaded, `foobarrep` environments in the main text have their number link to their repetition in the appendix. To suppress this behavior and have `foobarrep` environments treated as if `hyperref` were not loaded, one can specify the value `no` to the `forwardlinking` option when loading the package. (By default this option is set to `yes`.)

1.6 Advanced Features

We now describe a few advanced macros and environments, the usage of which is limited to special cases:

- `nestedproof` (*env.*) `nestedproof` is an environment that can be used within a `proof` environment deferred in the appendix; this is required because, for technical reasons, no `proof` environment can be nested within a deferred `proof` environment.

`\noproofinappendix` `\noproofinappendix` can be used inside repeated theorems that are not followed by a `proof` or `appendixproof` environment; the point is to ensure that a further `proof` environment cannot be mistakenly understood as a proof of the repeated theorem. It should not be needed in most situations as `apxproof` tries figuring out when a proof follows a repeated theorem automatically, but may occasionally be needed in complex scenarios.

`\nosectionappendix` `\nosectionappendix` is to be used inside a section that *does* contain appendix material, but for which a section in the appendix should not be created. This should be rarely needed. When this command is present, appendix material is appended to the end of the previously created section.

2 Supported Document Classes

Because `apxproof` modifies sectioning commands, bibliographies, and proofs, it may not work straight away with arbitrary document classes. It has currently been tested with and is supported for the following document classes:

- L^AT_EX standard document classes (e.g., `article.cls`)
- **KOMA-Script** (e.g., `scrartcl.cls`, `scrbook.cls`)
- **memoir.cls**
- **ACM SIG Proceedings** (e.g., `sig-alternate.cls`, `acmart.cls`)
- **Springer's Lecture Notes in Computer Science** (e.g., `llncls.cls`)
- **Schloß Dagstuhl's Leibniz International Proceedings in Informatics** (e.g., `lipics.cls`, `lipcs-v2016.cls`)

Other classes may work out of the box. Adding support for specific classes is possible and can be requested from the author of this package.

3 Known Issues and Limitations

We report here some issues we are currently aware of:

- When using `hyperref`, the appendix in the bibliography is not hyperlinked. This is to avoid possible issues with multiply defined bibliography entries.
- `appendixproof`, `proof`, `toappendix` environments cannot be nested. This is a limitation of the `fancyvrb` package that `apxproof` relies on. Note the existence of the `nestedproof` environment for nested proofs.
- `apxproof` poorly interacts with SyncT_EX: identifying which source line has produced which box does not work for appendix content managed by `apxproof` or repeated theorems. No obvious fix is known, though this issue will be investigated in the long term.
- Unless the `bibliography` option is set to `common`, the `bibunits` package is used to generate a second bibliography. This means any package, such as `biblatex`, that is incompatible with `bibunits` will not be compatible with `apxproof` unless `bibliography` is set to `common`.

Issues not listed here should be reported to the author.

4 License

Copyright © 2016–2025 by Pierre Senellart.

This work may be distributed and/or modified under the conditions of the L^AT_EX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of L^AT_EX version 2005/12/01 or later.

5 Contact

- <https://github.com/PierreSenellart/apxproof>
- Pierre Senellart <pierre@senellart.com>

Bug reports and feature requests should preferably be submitted through the *Issues* feature of GitHub.

6 Acknowledgments

Thanks to Antoine Amarilli for feedback and proofreading. Thanks to K. D. Bauer for the implementation of the forward-linking mechanism, and for various bugfixes. Thanks to Leonid Kostrykin for an initial implementation of the `forwardlinking` option.

7 Implementation

We now describe the entire code of the package, in a literate programming fashion. Throughout the package, we use the `axp@` prefix to identify local macros and environment names, which are not meant to be used by the final user.

7.1 Dependencies

We first load a few package dependencies:

- `environ` to easily define the repeated theorem environments.

```
1 \RequirePackage{environ}
```

- `etoolbox` to define simple toggles.

```
2 \RequirePackage{etoolbox}
```

- `fancyvrb` for the bulk of the work of exporting appendix material in an auxiliary file.

```
3 \RequirePackage{fancyvrb}
```

- `ifthen` for easier comparison of character strings.

```
4 \RequirePackage{ifthen}
```

- `kvoptions` to manage options passed to the package.

```
5 \RequirePackage{kvoptions}
```

- `catchfile` to be able to check the content of files `\input` within appendix content.

```
6 \RequirePackage{catchfile}
```

- `amsthm` for its `\newtheorem` macro. Some document classes (e.g., `lipics`) preload `amsthm`: this is fine, `\RequirePackage{amsthm}` will simply have no effect. On the other hand, some other document classes (e.g., `llncs` or `sig-alternate`) define a `proof` environment that conflicts with `amsthm`, so we have to undefine this environment before loading `amsthm`. In that case, we reestablish the existing proof environments, in case they had been customized (e.g., `sig-alternate`)

```
7 \@ifpackageloaded{amsthm}{
8   }{
9     \let\apx@oldamsthmproof\proof
10    \let\apx@oldamsthmendproof\endproof
11    \let\proof\undefined
12    \let\endproof\undefined
13  }
14  \RequirePackage{amsthm}
15  \ifdefined\apx@oldamsthmproof
16    \let\proof\apx@oldamsthmproof
17    \let\endproof\apx@oldamsthmendproof
18  \fi
```

7.2 Option Processing

Many names throughout the package use an atrobase (`@`) to avoid name conflict with user-defined names. To simplify the compilation of the documentation, we simply make it a regular character in all the rest.

```
19 \makeatletter
```

We setup the processing of options using `keyval` facilities.

```
20 \SetupKeyvalOptions{
21   family=axp,
22   prefix=axp@
23 }
```

We declare the following options:

- `appendix`, with a default value of `append` (other possible values: `strip`, `inline`);
- `bibliography`, with a default value of `separate` (other possible value: `common`);
- `repeqn`, with a default value of `same` (other possible value: `independent`).

```
\axp@appendix
```

```
24 \DeclareStringOption[append]{appendix}
```



```

\axp@bibliography
25 \DeclareStringOption[separate]{bibliography}

\axp@repeqn
26 \DeclareStringOption[same]{repeqn}

\axp@forwardlinking
27 \DeclareStringOption[yes]{forwardlinking}

28 \ProcessLocalKeyvalOptions*

We check that the value of the options are valid, and add a message to the
compilation log.
29 \ifthenelse{\equal{\axp@appendix}{append}}{
30   \message{apxproof: Appendix material appended to the document}
31 }{\ifthenelse{\equal{\axp@appendix}{strip}}{
32   \message{apxproof: Appendix material stripped}
33 }{\ifthenelse{\equal{\axp@appendix}{inline}}{
34   \message{apxproof: Appendix material inlined within the document}
35 }{
36   \errmessage{Error: unsupported option appendix=\axp@appendix\ for
37   package apxproof}
38 }}}
39 \ifthenelse{\equal{\axp@bibliography}{separate}}{

The external bibunits package is used to add a second bibliography for the appendix
material.
40 \RequirePackage{bibunits}
41 \message{apxproof: Separate bibliography for appendix material}
42 }{\ifthenelse{\equal{\axp@bibliography}{common}}{
43   \message{apxproof: Common bibliography for appendix and main text}
44 }{
45   \errmessage{Error: unsupported option bibliography=\axp@bibliography\ for
46   package apxproof}
47 }}
48 \ifthenelse{\equal{\axp@repeqn}{same}}{
49   \message{apxproof: Repeated equations keep the same numbering}
50 }{\ifthenelse{\equal{\axp@repeqn}{independent}}{
51   \message{apxproof: Repeated equations are independently numbered}
52 }{
53   \errmessage{Error: unsupported option repeqn=\axp@repeqn\ for
54   package apxproof}
55 }}

\axp@forward@suppress
56 \newbool{axp@forward@suppress}
57 \ifthenelse{\equal{\axp@forwardlinking}{yes}}{
58 }{\ifthenelse{\equal{\axp@forwardlinking}{no}}{
59   \message{apxproof: Disable forward linking}
60   \global\booltrue{axp@forward@suppress}%
61 }{
62   \errmessage{Error: unsupported option forwardlinking=\axp@repeqn\ for
63   package apxproof}
64 }}

```

7.3 Macros Common to All Compilation Modes

`\xp@newtheoremrep@definetheorem` Common to all compilation modes, we define `\xp@newtheoremrep@definetheorem`. When called with first argument `foobar`, we first undefine the existing `foobar` environment (and its counter) if it has already been defined (e.g., by the document class), then invoke `\axp@newtheorem` for the regular version of the theorem `foobar`, saving and restoring any existing theorem counter unless the `\newtheoremdep` redefines the base counter.

```

65 \def\xp@newtheoremrep@definetheorem#1#2#3#4{%
66   \expandafter\let\csname #1\endcsname\undefined
67   \expandafter\let\csname the#1\endcsname\undefined
68   \ifcsname c@#1\endcsname
69     \expandafter\expandafter\expandafter\let\expandafter\expandafter
70       \csname c@axp@#1\endcsname\csname c@#1\endcsname
71     \expandafter\let\csname c@#1\endcsname\undefined
72   \fi
73   \axp@newtheorem{#1}{#2}{#3}{#4}%
74   \ifcsname c@axp@#1\endcsname
75     \ifx\relax#2\relax
76       \expandafter\expandafter\expandafter\let\expandafter\expandafter
77         \csname c@#1\endcsname\csname c@axp@#1\endcsname
78     \else
79       \fi
80   \fi
81 }
```

`\axp@newtheorem` We introduce an intermediate `\axp@newtheorem` command to define a new theorem, `\@axp@newtheorem` differently depending on whether there is a section counter or not. This will be `\@@axp@newtheorem` useful, in particular to allow changing this definition depending on the document class. This command uses two intermediary commands, `\@axp@newtheorem` and `\@@axp@newtheorem`, for the non-starred and starred versions.

```

82 \def\axp@newtheorem{\@ifstar\@axp@newtheorem\@axp@newtheorem}
83 \def\@axp@newtheorem#1#2#3#4{%
84   \ifx\relax#4\relax
85     \newtheorem{#1}[#2]{#3}%
86   \else
87     \newtheorem{#1}{#3}[#4]%
88   \fi
89 }
90 \def\@@axp@newtheorem#1#2{%
91   \newtheorem*{#1}{#2}%
92 }
```

`\newtheoremrep` We define the high-level `\newtheoremrep` to have the same syntax as `amsthm`'s `\axp@newtheoremreptmp` `\newtheorem`. For this purpose, we need a little trick to deal with the second and fourth optional arguments, which is what `\@oparg` and `\axp@newtheoremreptmp` are used for. `\axp@newtheoremrep` is defined differently depending on the compilation mode.

```

93 \newcommand\newtheoremrep[1]{%
94   \@oparg{\axp@newtheoremreptmp{#1}}[]%
95 }
96 \def\axp@newtheoremreptmp#1[#2]#3{%
97   \@oparg{\axp@newtheoremrep{#1}[#2]{#3}}[]%
```

98 }

`proofsketch` (*env.*) Simple `proofsketch` environment.

```
99 \newenvironment{proofsketch}{\begin{axp@oldproof}[Proof sketch]}\end{axp@oldproof}}
```

`\mainbodyrepeatedtheorem` We provide sensible defaults for these user-customizable macros. Even though

`\appendixrefname` they are not all useful in all modes, we define them for all modes so that a

`\appendixbibliographystyle` `\renewcommand` works in all cases.

```
\appendixbibliographyprelim 100 \newcommand{\mainbodyrepeatedtheorem}{}
\appendixprelim              101 \providecommand{\appendixrefname}{References for the Appendix}
\appendixsectionformat      102 \newcommand{\appendixbibliographystyle}{alpha}
```

```
103 \newcommand{\appendixbibliographyprelim}{}
104 \newcommand{\appendixprelim}{\clearpage\onecolumn}
```

```
105 \newcommand{\appendixsectionformat}[2]{Proofs for Section~#1\ (#2)}
```

```
106 \newcommand{\appendixbibliographyprelim}{}
107 \newcommand{\appendixprelim}{\clearpage\onecolumn}
```

```
108 \newcommand{\appendixsectionformat}[2]{Proofs for Section~#1\ (#2)}
```

```
109 \newcommand{\appendixsectionformat}[2]{Proofs for Section~#1\ (#2)}
```

`axp@oldproof` (*env.*) We save the definition of the existing proof environment.

```
106 \let\axp@oldproof\proof
```

```
107 \let\endaxp@oldproof\endproof
```

We define a utility macro that will be used to properly set the `\label` command (and its `amsmath` counterpart, `\label@in@display`) for equations within repeated theorems, depending on the compilation mode.

`\axp@redefinelabels`

```
108 \newcommand{\axp@redefinelabels}{%
```

```
109 \providecommand\label@in@display{}%
```

```
110 \ifthenelse{\equal{\axp@appendix}{inline}}{%
```

```
111 \let\axp@oldlabel\label
```

```
112 \let\axp@oldlabel@in@display\label@in@display
```

```
113 \renewcommand\label[1]{%
```

```
114 \axp@oldlabel{##1}%
```

```
115 \axp@oldlabel{##1-axp}%
```

```
116 }%
```

```
117 \renewcommand\label@in@display[1]{%
```

```
118 \axp@oldlabel@in@display{##1}%
```

```
119 \axp@oldlabel{##1-axp}%
```

```
120 }%
```

```
121 }{%
```

```
122 \let\axp@oldlabel\label
```

```
123 \let\axp@oldlabel@in@display\label@in@display
```

```
124 \renewcommand\label[1]{\axp@oldlabel{##1-axp}}%
```

```
125 \renewcommand\label@in@display[1]{\axp@oldlabel@in@display{##1-axp}}%
```

```
126 }%
```

```
127 }
```

7.3.1 Class-Specific Behavior

Finally, some class-specific behavior common to all compilation modes.

`llncs` and other Springer document classes

```
128 \ifdefined\spnewtheorem
```

`\@axp@newtheorem` It is necessary to use `\spnewtheorem` instead of `\newtheorem` in Springer document classes to obtain standard formatting.

```

129 \def\@axp@newtheorem#1#2#3#4{%
130   \ifx\relax#4\relax
131     \ifx\relax#2\relax
132       \spnewtheorem{#1}{#3}{\bfseries}{\itshape}%
133     \else
134       \spnewtheorem{#1}[#2]{#3}{\bfseries}{\itshape}%
135     \fi
136   \else
137     \spnewtheorem{#1}{#3}[#4]{\bfseries}{\itshape}%
138   \fi
139 }
140 \def\@axp@newtheorem#1#2{%
141   \spnewtheorem*{#1}{#2}{\upshape\bfseries}{\itshape}%
142 }

```

`proofsketch` We redefine the `proofsketch` environment, which is used differently in the base class.

```

143 \renewenvironment{proofsketch}{\begin{axp@oldproof}[sketch]}\end{axp@oldproof}}

```

We have to redefine the macro `\@thmcountersep` for proper sectioned counters.

```

144 \def\@thmcountersep{.}
145 \fi

```

acmart Some versions of `acmart` define a `\ACM@origsection` macro and verify that `\section` has the same content. We need to update `\ACM@origsection` accordingly.

```

146 \ifdefined\ACM@origsection
147   \AtBeginDocument{\let\ACM@origsection\section}
148 \fi

```

7.4 Inline Compilation Mode

```

149 \ifthenelse{\equal{\axp@appendix}{inline}}{

```

`\axp@newtheoremrep` In inline mode, `\axp@newtheoremrep` uses `\axp@newtheoremrep@definetheorem` to define the regular theorem environment and creates a repeated theorem environment that behaves exactly as the regular theorem environment, while calling `\axp@redefinelabels` to make sure that `-axp` variants of equation counters are defined.

```

150   \def\axp@newtheoremrep#1[#2]#3[#4]{%
151     \axp@newtheoremrep@definetheorem{#1}{#2}{#3}{#4}%
152     \NewEnviron{#1rep}[1][]{%
153       \ifx\relax##1\relax
154         \begin{#1}\axp@redefinelabels\BODY\end{#1}%
155       \else
156         \begin{#1}[##1]\axp@redefinelabels\BODY\end{#1}%
157       \fi
158     }
159   }

```

`inlineproof (env.)` In inline mode, these environments behave like the regular `proof` environment.

`nestedproof (env.)` 160 `\let\inlineproof\proof`

`appendixproof (env.)` 161 `\let\endinlineproof\endproof`

162 `\let\nestedproof\proof`

163 `\let\endnestedproof\endproof`

164 `\let\appendixproof\proof`

165 `\let\endappendixproof\endproof`

`toappendix (env.)` In inline mode, this environment and these macros are no-ops.

`\noproofinappendix` 166 `\newenvironment{toappendix}{}{}`

`\nosectioninappendix` 167 `\let\noproofinappendix\relax`

168 `\let\nosectioninappendix\relax`

169 }

7.5 Append or Strip Compilation Modes

170 {

We now deal with the case where `apxproof` really does something useful: either append the appendix material to the document, or strip it entirely.

7.5.1 Auxiliary File for the Appendix

`\axp@proofsfile` We open a new auxiliary file, with extension `.axp`, where the appendix material will be dumped.

171 `\AtBeginDocument{`

172 `\newwrite\axp@proofsfile`

173 `\immediate\openout\axp@proofsfile=\jobname.axp`

174 }

`proof (env.)` At the beginning of this file, we make `@` a regular character (since it will be used in several places for internal names) and reestablish the original definition of the `proof` environment and the `\section` macro.

175 `\AtBeginDocument{`

176 `\immediate\write\axp@proofsfile{%`

177 `\noexpand\makeatletter`

178 `\noexpand\let\noexpand\proof\noexpand\axp@oldproof`

179 `\noexpand\let\noexpand\endproof\noexpand\endaxp@oldproof`

180 `\noexpand\let\noexpand\claimproof\noexpand\axp@oldclaimproof`

181 `\noexpand\let\noexpand\endclaimproof\noexpand\endaxp@oldclaimproof`

182 `\noexpand\let\noexpand\section\noexpand\axp@oldsection`

183 }

184 }

`\axp@unactivateeightbit` We need an auxiliary macro to disable active characters that have the high bit set when writing to the `.axp` file. See <https://tex.stackexchange.com/a/145361/166858>

185 `\def\axp@unactivateeightbit{%`

186 `\count@=128%`

187 `\loop`

188 `\catcode\count@=12%`

189 `\ifnum\count@<255%`

190 `\advance\count@\@ne`

191 `\repeat}`

`axp@VerbatimOut` (*env.*) Using the functionalities of the `fancyvrb` package, we define a custom verbatim environment `axp@VerbatimOut` that writes every line to the `\axp@proofsfile`.
`\FVB@axp@VerbatimOut`
`\FVE@axp@VerbatimOut` We also use the previous macro to disable active characters with the eighth bit set, and we make sure the catcode of `@` is reset for every verbatim environment, in case it is used by the user (e.g., as in the `xypic` package). Finally, as an additional precaution, we reset `\FV@CatCodesHook` that is for example set by the `commandchars` or `commentchar` option of `\fvset`.

```

192 \DefineVerbatimEnvironment{axp@VerbatimOut}{axp@VerbatimOut}{}
193 \def\FVB@axp@VerbatimOut{%
194   \@bsphack
195   \begingroup
196     \axp@unactivateeightbit
197     \FV@DefineWhiteSpace
198     \def\FV@Space{\space}%
199     \FV@DefineTabOut
200     \def\FV@ProcessLine{\immediate\write\axp@proofsfile}%
201     \let\FV@FontScanPrep\relax
202     \let\@noligs\relax
203     \def\FV@CatCodesHook{}%
204     \FV@Scan}
205 \def\FVE@axp@VerbatimOut{%
206   \immediate\write\axp@proofsfile{\noexpand\makeatletter}%
207   \endgroup\@esphack}

```

`toappendix` (*env.*) The entire content of this environment is put in appendix, starting a new appendix section beforehand if needed.

```

208 \newenvironment{toappendix}
209   {\axp@writesection\axp@VerbatimOut}
210   {\endaxp@VerbatimOut}

```

7.5.2 Definition of New Theorems

`axp@seenrepththeorem` Used to indicate whether a repeated theorem was just typeset, without its proof.

```

211 \newtoggle{axp@seenrepththeorem}

```

`axp@rpcounter` Sequentially incremented for every repeated theorem, used to create labels.

```

212 \newcounter{axp@rpcounter}

```

`axp@equation` Used to save the value of the `equation` counter, when `repeqn` is set to `same`.

```

213 \newcounter{axp@equation}
214 \newcounter{axp@equationx}

```

`axp@newtheoremrep` With first argument `foobar`, we use `\axp@newtheoremrep@definetheorem` to define the regular version of the theorem `foobar`. We then patch `\begin{foobar}` so as not to expect a proof in the appendix and define an internal theorem `axp@foobarrp` that will be used in the appendix to restate the existing theorem.

```

215 \def\axp@newtheoremrep#1[#2]#3[#4]{%
216   \axp@newtheoremrep@definetheorem{#1}{#2}{#3}{#4}%
217   \expandafter\pretocmd\csname #1\endcsname{\noproofinappendix}{-}{-}%
218   \axp@newtheorem*{axp#1rp}{#3}%
219   \axp@forward@setup{#1}{#2}{#3}{#4}%

```

We then define a `foobarrep` environment that increments the `axp@rpcounter` and typeset the regular `foobar` theorem with a label derived from the counter, along with a possible custom command to identify repeated theorems. We distinguish the case when the theorem argument has a note and when it does not. We save the equation counter before typesetting the theorem environment, to reset it to the same value in the repeated environment when `repeqn` is set to `same`.

```

220 \NewEnviron{#1rep}[1][]{%
221 \ifthenelse{\equal{\axp@repeqn}{same}}{%
222 \setcounter{axp@equation}{\value{equation}}%
223 }{}%
224 \addtocounter{axp@rpcounter}{1}%
225 \ifx\relax##1\relax
226 \axp@with@forward{#1}{\begin{#1}}\label{axp@r\roman{axp@rpcounter}}%
227 \else
228 \axp@with@forward{#1}{\begin{#1}[\##1]}\label{axp@r\roman{axp@rpcounter}}%
229 \fi
230 \mainbodyrepeatedtheorem
231 \BODY\end{#1}%

```

We set the `axp@seenrepththeorem` toggle to indicate that we are looking for the proof of the theorem, then store in a macro the content of the theorem's body.

```

232 \global\toggletrue{axp@seenrepththeorem}%
233 \global\expandafter\let\csname rplet\roman{axp@rpcounter}%
234 \endcsname
235 \BODY

```

Possibly after starting a new appendix section if needed, we typeset a repeated version of the theorem using the `axp@foobarrp` environment and a reference to the previously defined label. We use `\axp@redefinlabels` in this environment to avoid multiply defined labels. We have to deal in a careful way with theorem notes: we want to use a theorem note to display the number of the repeated theorem, but theorem notes are usually typeset in a much different way (different font, parentheses) than theorem headings. In the case of the Springer document classes, we use the `\theopargself` macro to disable parentheses. For other document classes, we need to manually patch the `\thmhead` command at the right time. We also specially cover the case of the ACM document class where `\acmplainnotefont` is used instead of `\thm@notefont`.

```

236 \axp@writesection%
237 \immediate\write\axp@proofsfile{\noexpand\makeatletter}%
238 \ifthenelse{\equal{\axp@repeqn}{same}}{%
239 \immediate\write\axp@proofsfile{%
240 \noexpand\setcounter{axp@equationx}{\value{equation}}%
241 \noexpand\setcounter{equation}{\theaxp@equation}%
242 }%
243 }{}%
244 \ifbool{axp@forward@suppress}{%
245 \global\def\axp@refstar{\ref*}%
246 }{%
247 \global\def\axp@refstar{\ref}%
248 }
249 \immediate\write\axp@proofsfile{%
250 \ifdefined\theopargself
251 \noexpand\theopargself

```

```

252     \else
253       \noexpand\pretocmd{\noexpand\@begintheorem}{%
254         \noexpand\patchcmd{\noexpand\thmhead}{\noexpand\@acmplainnotefont}{-}{-}{-}%
255         \noexpand\patchcmd{\noexpand\thmhead}{\noexpand\the\noexpand\thm@notefont}{-}{-}{-}%
256         \noexpand\patchcmd{\noexpand\thmhead}{(}{-}{-}{-}%
257         \noexpand\patchcmd{\noexpand\thmhead}{)}{-}{-}{-}%
258       }{}{}
259     \fi
260     \noexpand\begin{axp@#1rp}
261     [%
262       \noexpand\axp@refstar{axp@r\roman{axp@rpcounter}}}%
263     \@ifnotempty{##1}{%
264       \ifdefined\theopargself
265       \else
266         \ifdefined\@acmplainnotefont
267           \noexpand\@acmplainnotefont
268         \else
269           \noexpand\ifdefined\noexpand\thm@notefont
270             \noexpand\the\noexpand\thm@notefont
271           \noexpand\fi
272         \fi
273       \fi
274     } (\unexpanded{##1})}%
275   }%
276 ]%
277 \noexpand\axp@forward@target{axp@fw@r\roman{axp@rpcounter}}{}%
278 \noexpand\axp@redefinelabels
279 \expandafter\noexpand\csname rplet\roman{axp@rpcounter}%
280 \endcsname
281 \noexpand\end{axp@#1rp}
282 }}%
283 \ifthenelse{\equal{\axp@repeqn}{same}}{%
284   \immediate\write\axp@proofsfile{%
285     \noexpand\setcounter{equation}{\value{axp@equationx}}}%
286   }%
287 }{}%
288 }%
289 }

```

7.5.3 Forward-Linking Mechanism

When `hyperref` is loaded, `foobarrep` environments in the main text have their number link to their repetition in the appendix.

`\axp@with@forward` In order to make the number of the `foobarrep` theorem a link to its repeated version, we temporarily redefine the `\thefoobar` command, or, if we inherited the counter from a `bazbar` environment, the `\thebazbar` command. This seems to be the only robust way to make the number a `\hyperlink`, without adding extensive dependence on internals of `amsthm`, the builtin `\newtheorem` and possibly document-class specific definitions.

In order to allow users to redefine `\thefoobar` without breaking this feature, we redefine `\thefoobar` only for the duration of the `\begin{foobar}` form, resetting it to the old value as soon as possible.

Redefining `\thefoobar` has the side effect of changing `\newlabel` entries in the `.aux` file, so we need to be able to disable addition of the hyperlink, which is why we use an intermediate `\axp@forward@link{<target>}{<text>}` macro. We also redefine `\theHfoobar` which is used by `hyperref` but not defined if `hyperref` was loaded after `\newtheoremrep` was used and `\protect` it to output it verbatim into the `.aux` file.

These hyperlinks are of course disabled in the `strip` compilation mode.

```

290 \newcommand{\axp@with@forward}[2]{%
291   \ifthenelse{\equal{\axp@appendix}{strip}}{#2}{
292     \global\booltrue{\axp@forward}%
293     \ifcsundef{\axp@old@the\csname axp@cn@#1\endcsname}{%
294       \csletcs{\axp@old@the\csname axp@cn@#1\endcsname}{the\csname axp@cn@#1\endcsname}%
295       \csletcs{theH\csname axp@cn@#1\endcsname}{the\csname axp@cn@#1\endcsname}%
296       \csdef{the\csname axp@cn@#1\endcsname}{%
297         \protect\axp@forward@link{\axp@fw@r\roman{\axp@rpcounter}}}%
298         {\csname axp@old@the\csname axp@cn@#1\endcsname\endcsname}%
299       }%
300     }{}%
301     #2%
302     \ifcsdef{\axp@old@the\csname axp@cn@#1\endcsname}{%
303       \csletcs{the\csname axp@cn@#1\endcsname}{\axp@old@the\csname axp@cn@#1\endcsname}%
304     }{}%
305     \global\boolfalse{\axp@forward}
306   }%

```

`\axp@forward@link` Dummy macro, for handling the unwanted change of the `\newlabel` entry in the `.aux` file caused by changing the definition of `\thefoobar`. We also use `\texorpdfstring` to detect whether the reference appears within a PDF bookmark, in which case we skip the test altogether.

```

307 \newbool{\axp@forward}
308 \newcommand{\axp@forward@link}[2]{%
309   \ifdefined\texorpdfstring\else\newcommand\texorpdfstring[2]{#2}\fi
310   \texorpdfstring{%
311     \ifboolexpr{ bool {\axp@forward} and not bool {\axp@forward@suppress} }{%
312       \ifcsdef{hyperlink}{%
313         \hyperlink{#1}{#2}%
314       }{%
315         #2%
316       }%
317     }{%
318       #2%
319     }%
320   }{%
321     #2%
322   }%
323 }%

```

`\axp@forward@target` Provides the needed `\hypertarget`. Intended to be written to the `.axp` file.

```

324 \newcommand{\axp@forward@target}[2]{%
325   \ifcsname hypertarget\endcsname
326     \hypertarget{#1}{#2}%
327   \else
328     #2%

```

```

329   \fi
330 }

```

`\axp@forward@setup` In order to support counter inheritance with the first optional argument of `\newtheoremrep`, we need access to the name of the counter. For compliance with the behavior of `\@axp@newtheorem`, the first optional argument (#2) is ignored if the second optional argument (#4) is given. We also check whether `aliascnt` was used to define this counter; if so, we use the original name as the counter name to handle the way such aliases work.

```

331 \newcommand{\axp@forward@setup}[4]{%
332   \csedef{axp@cn@#1}{\ifblank{#4}{\ifblank{#2}{#1}{\ifcsdef{AC@cnt@#1}{#1}{#2}}{#1}}%
333 }

```

7.5.4 Proof Environments

`\noproofinappendix` Utility macro that toggles `axp@seenreptheorem` to false.

```

334 \newcommand\noproofinappendix{%
335   \global\togglefalse{axp@seenreptheorem}%
336 }

```

`appendixproof (env.)` We dump the content of this in appendix, within an original `proof` environment, possibly after creating a new appendix section. We support optional arguments in proofs, but we need to be careful not to gobble any newline character, that would make it impossible to process the first line with `fancyvrb`.

```

337 \def\appendixproof{\catcode'\^M=\active\@ifnextchar[{\catcode'\^M=5\@appendixproof[axp@
338 \def\@appendixproof[#1]%
339   {%
340     \axp@writesection
341     \immediate\write\axp@proofsfile{%
342       \noexpand\makeatletter\noexpand\begin{#1}\noexpand\makeatother%
343     }%
344     \axp@VerbatimOut
345   }
346 \def\@appendixproof[#1][#2]%
347   {%
348     \axp@writesection
349     \immediate\write\axp@proofsfile{%
350       \noexpand\makeatletter\noexpand\begin{#1}[\unexpanded{#2}]\noexpand\makeatother%
351     }%
352     \axp@VerbatimOut
353   }
354 \def\endappendixproof
355   {%
356     \endaxp@VerbatimOut
357     \immediate\write\axp@proofsfile{%
358       \noexpand\end{axp@oldproof}%
359     }%
360     \noproofinappendix
361   }

```

`proof (env.)` This environment either puts the proof in appendix, if we are after a repeated theorem without its proof, or inlines it otherwise. We support optional arguments

in proofs, but we need to be careful not to gobble any newline character, that would make it impossible to process the first line with `fancyvrb`.

```

362 \def\proof{\catcode'\^M=\active\ltx@ifnextchar@nospace[\catcode'\^M=5\axp@@proof]{\catcode'\^M=5\axp@proof}%
363 \def\axp@proof
364   {%
365     \iftoggle{axp@seenreptheorem}{%
366       \appendixproof
367     }{%
368       \axp@oldproof
369     }%
370   }
371 \def\axp@@proof[#1]%
372   {%
373     \iftoggle{axp@seenreptheorem}{%
374       \appendixproof[#1]
375     }{%
376       \axp@oldproof[#1]
377     }%
378   }
379 \def\endproof
380   {%
381     \iftoggle{axp@seenreptheorem}{%
382       \endappendixproof
383     }{%
384       \endaxp@oldproof
385     }%
386   }

```

`claimproof` (*env.*) If the `claimproof` environment exists (lipics document class or user-defined), we also redefine it, in the same way.

```

387 \AtBeginDocument{
388   \ifdefined\claimproof
389     \let\axp@oldclaimproof\claimproof
390     \let\endaxp@oldclaimproof\endclaimproof
391     \def\claimproof{\catcode'\^M=\active\ltx@ifnextchar@nospace[\catcode'\^M=5\axp@@claimproof]{\catcode'\^M=5\axp@claimproof}%
392     \def\appendixclaimproof{\catcode'\^M=\active\ltx@ifnextchar[\catcode'\^M=5\@appendixproof]{\catcode'\^M=5\@appendixproof}%
393     \def\axp@claimproof
394       {%
395         \iftoggle{axp@seenreptheorem}{%
396           \appendixclaimproof
397         }{%
398           \axp@oldclaimproof
399         }%
400       }
401     \def\axp@@claimproof[#1]%
402       {%
403         \iftoggle{axp@seenreptheorem}{%
404           \appendixclaimproof[#1]%
405         }{%
406           \axp@oldclaimproof[#1]%
407         }%
408       }
409     \def\endclaimproof
410       {%

```

```

411     \iftoggle{axp@seenrepththeorem}{%
412         \endappendixclaimproof
413     }{%
414         \endaxp@oldclaimproof
415     }%
416 }
417 \def\endappendixclaimproof
418 {%
419     \endaxp@VerbatimOut
420     \immediate\write\axp@proofsfile{%
421         \noexpand\end{axp@oldclaimproof}}%
422     }%
423     \noproofinappendix
424 }
425 \fi
426 }

```

`inlineproof` (*env.*) These two environments are synonyms for the original `proof` environment.

`nestedproof` (*env.*)

```

427 \let\inlineproof\axp@oldproof
428 \let\endinlineproof\endaxp@oldproof
429 \let\nestedproof\axp@oldproof
430 \let\endnestedproof\endaxp@oldproof

```

7.5.5 Section Management

`axp@seccounter` Sequentially incremented for every section, used to create labels.

```

431 \newcounter{axp@seccounter}

```

`\axp@sectitle` Saves the title of the last encountered section.

```

432 \def\axp@sectitle{}

```

`\axp@section` This command `\axp@section` behaves similarly to `\axp@oldsection`, except that `\axp@@sectiontestinput` it first tests whether a `\section` follows, and if so, does not produce anything. This `\axp@@sectiontestsection` is useful to avoid producing empty sections in the appendix. Using the `catchfile` package, we also check whether a `\section` is within an `\input` that immediately follows.

```

433 \def\axp@section#1{%
434     \@ifnextchar\input
435     {\axp@@sectiontestinput{#1}}%
436     {\axp@@sectiontestsection{#1}}%
437 }
438 \def\axp@@sectiontestinput#1\input#2{%
439     \CatchFileDef{\axp@tmp}{#2}{}%
440     \def\axp@tmptmp{\axp@@sectiontestsection{#1}}%
441     \expandafter\axp@tmptmp\axp@tmp%
442 }
443 \def\axp@@sectiontestsection#1{\@ifnextchar\section{\makeatother}{\axp@oldsection{#1}\makeatother}

```

`\axp@oldsection` We redefine the `\section` command to create a label based on `axp@seccounter` and to store its title in `\axp@sectitle`. In order to support starred and unstarred `\@section` versions, as well as the optional short-title argument, the intermediate macros `\@@section` `\@section` and `\@@section` are needed.

```

444 \let\axp@oldsection\section

```

```

445 \def\section{\@ifstar\@section\@@section}
446 \newcommand{\@section}[2][\relax]{\axp@@section*{#1}{#2}}%
447 \newcommand{\@@section}[2][\relax]{\axp@@section{}{#1}{#2}}%
448 \newcommand{\axp@@section}[3]{%
449   \global\def\axp@sectitle{#3}%
450   \ifx\relax#2\relax
451     \axp@oldsection#1{#3}%
452   \else
453     \axp@oldsection#1[{#2}]{#3}%
454   \fi
455   \addtocounter{axp@seccounter}{1}%
456   \label{axp@s\roman{axp@seccounter}}}%
457 }

```

`\nosectionappendix` We remove the current section title, to indicate no section should be created in the appendix.

```

458 \newcommand{\nosectionappendix}{
459   \global\def\axp@sectitle{}%
460 }

```

`\axp@writesection` If `\axp@sectitle` is not empty, we create a new section in the appendix, referring to the main text section.

Here, we wrap `\ref{axp@si}` into `\axp@protectref@i`, in order to protect the label name from wrongly being converted to uppercase, e.g., in `fancyhdr` with `\pagestyle{fancy}`.

This macro is defined both in the `.aux` file (in order to ensure availability when typesetting the `\tableofcontents`), and immediately before typesetting the appendix section (to ensure availability in the `\section` command).

```

461 \newcommand\axp@writesection{%
462   \ifx\axp@sectitle\@empty
463   \else
464     \edef\axp@tmp{%
465       \noexpand\global\noexpand\def
466       \expandonce{\csname axp@protectref@\roman{axp@seccounter}\endcsname}{%
467         \noexpand\ref{axp@s\roman{axp@seccounter}}}%
468       }%
469     }%
470     \immediate\write\@auxout{\expandonce\axp@tmp}
471     \immediate\write\axp@proofsfile{%
472       \expandonce\axp@tmp~J%
473       \noexpand\axp@section{%
474         \noexpand\appendixsectionformat{%
475           \protect
476           \expandonce{\csname axp@protectref@\roman{axp@seccounter}\endcsname}%
477           }{\expandonce\axp@sectitle}%
478         }%
479       }%
480     \nosectionappendix
481   \fi
482 }

```

Finally, in a somewhat ad hoc manner, we disable the whole section management for `\tableofcontents`, which may be typeset using a section heading, but for which automatic section management does not make sense.

```

\axp@oldtableofcontents
\tableofcontents 483 \let\axp@oldtableofcontents\tableofcontents
484 \def\tableofcontents{\let\section\axp@oldsection\axp@oldtableofcontents}}

```

7.5.6 Append Compilation Mode

```

485 \ifthenelse{\equal{\axp@appendix}{append}}{
\axp@oldbibliography Unless the bibliography option is set to common, we need to set the appendix
\ bibliography bibliography source to be the same as that of the main text, thanks to bibunits's
\defaultbibliography macro.
486 \ifthenelse{\equal{\axp@bibliography}{separate}}{
487 \let\axp@oldbibliography\ bibliography
488 \renewcommand\ bibliography[1]{%
489 \defaultbibliography{#1}%
490 \axp@oldbibliography{#1}%
491 }
492 }{}

```

After the end of the main text, we add the appendix (after the command `\appendixprelim` is issued) within a `bibunit` environment so as to typeset a separate bibliography for the appendix (unless the `bibliography` option is set to `common`). This is done using `\pretocmd` on `\@enddocumenthook` instead of `\AtEndDocument` because we want the code to be run before any code in the `\@enddocumenthook` that has been set in the document class, as in the `amsart` document class. There is an extra test to ensure an empty bibliography environment is not produced. The name of the bibliography is changed to `\appendixrefname`; in most document classes, it is called `\refname` but it is occasionally (`scrartcl`, `scrreprt`) called `\bibname`. An ad-hoc test is added to fix a conflict with the `natbib` package which redefines `bibcite` at the end of the document.

```

493 \pretocmd{\@enddocumenthook}{%
494 \ifdefined\NAT@testdef
495 \renewcommand\bibcite[2]{%
496 \global\@namedef{b@#1\@extra@binfo}{#2}%
497 }
498 \fi
499 \appendixprelim
500 \appendix
501 \ifthenelse{\equal{\axp@bibliography}{separate}}{
502 \begin{bibunit}[\appendixbibliographystyle]
503 }{}
504 \immediate\closeout\axp@proofsfile
505 \input{\jobname.axp}
506 \ifthenelse{\equal{\axp@bibliography}{separate}}{
507 \ifdefined\refname
508 \renewcommand{\refname}{\appendixrefname}
509 \else\ifdefined\bibname
510 \renewcommand{\bibname}{\appendixrefname}
511 \fi\fi
512 \let\axp@oldthebibliography\thebibliography
513 \let\endaxp@oldthebibliography\endthebibliography
514 \renewenvironment{thebibliography}[1]{%
515 \def\axp@tmp{#1}%

```

```

516         \ifx\axp@tmp\empty
517         \gdef\axp@noappendixbibliography1\relax
518         \else
519         \begin{axp@oldthebibliography}{#1}%
520         \fi
521     }{%
522         \ifdefined\axp@noappendixbibliography\relax\else\end{axp@oldthebibliography}%
523     \fi}
524     \appendixbibliographyprelim
525     \putbib
526     \end{bibunit}
527     \ifdefined\NAT@testdef
528     \let\bibcite\NAT@testdef
529     \fi
530     }{}
531 }{}{}
532 }{}

```

7.5.7 Class-Specific Behavior

We conclude with some class-specific behavior.

ACM Document Classes (old versions, till 2017)

```

533 \ifdefined\acmtitlebox

```

We first redefine the `proofsketch` environment, which is used differently in the base class.

```

534 \renewenvironment{proofsketch}{\begin{axp@oldproof}[sketch]}\end{axp@oldproof}}

```

We adjust the styling of theorems for the needs of `apxproof`.

```

535 \newtheoremstyle{mystyle}
536 {6pt}
537 {6pt}
538 {\itshape}
539 {10pt}
540 {\scshape}
541 {.}
542 {.5em}
543 {}
544 \theoremstyle{mystyle}

```

`\thebibliography` The section title of the bibliography is in uppercase in these document classes. In addition, the `\thebibliography` macro hard-codes twice the section title, so we

`\appendixrefname` un-hardcode it so that it can be modified in the appendix.

```

545 \patchcmd{\thebibliography}{References}{\protect\refname}{}{}
546 \patchcmd{\thebibliography}{References}{\protect\refname}{}{}
547 \newcommand{\refname}{REFERENCES}
548 \renewcommand{\appendixrefname}{REFERENCES FOR THE APPENDIX}
549 \fi

```

lipcs

550 \ifdefined\lipics@opterrshort

\appendixbibliographyprelim The default bibliography in the lipics document class formatting is not compatible with the alpha bibliography style. We fix this here.

```
551 \renewcommand{\appendixbibliographyprelim}{%
552 \global\let\@oldbiblabel\@biblabel
553 \def\@biblabel{\hspace*{-2em}\small\@oldbiblabel}%
554 }

555 \fi
556 }
```

Change History

v1.0.0	newtheoremrep	2
General: Initial released version . .	Fix compilation of proofsketch environment in inline mode . . .	3
v1.0.1		
General: Prevent empty bibliography environment; fix typos	v1.0.6	
	General: Better support of Springer document classes	11
v1.0.2	Deal with document classes where the bibliography is called \bibname	22
axp@newtheoremrep: Fix display of repeated theorem counter in some document classes	Support of new ACM document class (acmart.cls)	23
Fix missing space between repeated theorem counter and theorem note	\axp@newtheorem: Introduce intermediary command for theorem macro	10
v1.0.3	\axp@writesection: Fix extraneous space after section number in appendix titles . . .	21
General: Note on entire sections in appendix	axp@newtheoremrep: Better handling of note-free theorems in document classes that treat theorems differently when they have an empty note	15
\appendixbibliographyprelim: Support for lipics-v2016	Fix incorrect use of \noexpand in optional argument of macro environment	15
proofsketch: Ignore spaces after beginning of Proof sketch . . .		
v1.0.4	v1.1.0	
General: More faithful theorem style for ACM templates	General: Added forward-link mechanism (K. D. Bauer) . . .	16
More robust coherent styling of proof sketches	\appendixsectionformat: Fix missing space in default \appendixsectionformat . . .	11
Re-establish custom proof environments	\axp@proofsfile: Initialization deferred to \AtBeginDocument for compatibility with \dumped precompiled preambles (K. D. Bauer)	13
Show options commented on in margin and index		
\appendixprelim: Configurable appendix style		
\axp@bibliography: bibliography option		
v1.0.5		
General: Ability to specify a sectioning counter in		

<code>\axp@redefinlabels</code> : Fix <code>\label</code> not being disabled in <code>amsmath</code> environments, where <code>\label@in@display</code> is used instead (K. D. Bauer)	11		
<code>\axp@repeqn</code> : <code>repeqn</code> option	9		
<code>\axp@unactivateeightbit</code> : Fix compilation of non-ASCII characters with <code>\usepackage[utf8]{inputenc}</code>	13		
<code>\axp@writesection</code> : Make <code>\axp@tmp</code> wrapper more robust. Resolves issues from use of section title in <code>fancyhdr</code> , and in <code>\tableofcontents</code> (K. D. Bauer).	21		
<code>\FVE@axp@VerbatimOut</code> : Make <code>apxproof</code> compatible with independent use of <code>fancyvrb</code>	14		
<code>\section</code> : Fix handling of fragile macros within section headings. See #22.	20		
Rewrote definition of <code>\section</code> to enable optional argument. See #23. (K. D. Bauer)	20		
<code>\tableofcontents</code> : Disable section management for table of contents	22		
v1.2.0			
General: Do not load <code>bibunits</code> if <code>bibliography</code> is set to <code>common</code>	9		
<code>\axp@newtheoremrep</code> : Fix formatting of theorems without notes in some document classes in inline mode	12		
<code>\axp@newtheoremrep@definetheorem</code> : Restore predefined theorem counters	10		
<code>\axp@redefinlabels</code> : Fix extra spacing erroneously introduced within the <code>\axp@redefinlabels</code> macro	11		
<code>\mainbodyrepeatedtheorem</code> : Configurable repeated theorem command	11		
<code>proofsketch</code> : Fix proof sketches in inline compilation mode for Springer document classes	12		
<code>axp@newtheoremrep</code> : Fix display of theorem notes	15		
		v1.2.1	
		General: Ad hoc fix for <code>natbib</code> package conflict	22
		<code>\appendixrefname</code> : Fix compatibility with <code>memoir</code> document class	11
		<code>\FVE@axp@VerbatimOut</code> : Fix compatibility with <code>xypic</code> package	14
		Fix compatibility with other uses of <code>fancyvrb</code> that set <code>\FV@CatCodesHook</code>	14
		v1.2.2	
		General: Compatibility with AMS document classes: do not use <code>\AtEndDocument</code>	22
		<code>\axp@section</code> : Detect a section within an included file to avoid produced useless sections	20
		<code>axp@newtheoremrep</code> : Fix handling of optional arguments of repeated theorems containing optional arguments	15
		v1.2.3	
		General: More robust redefinition of <code>thebibliography</code> environment, for compatibility with <code>tocbibind</code>	22
		<code>\axp@forwardlinking</code> : <code>forwardlinking</code> option	9
		v1.2.4	
		<code>appendixproof</code> : Support optional arguments in proofs	18
		<code>claimproof</code> : Support for <code>claimproof</code> environment from <code>lipics</code>	19
		v1.2.5	
		<code>claimproof</code> : Support for locally defined <code>claimproof</code> environment	19
		<code>axp@forward</code> : Remove forward linking command from PDF bookmarks	17
		v1.2.6	
		General: Fix compilation issues in recent versions of <code>acmart</code>	12
		<code>\axp@forward@setup</code> : Support for <code>aliascnt</code> counters	18
		<code>\axp@newtheoremrep@definetheorem</code> : Undefine counter printing command to avoid warning	10

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	
\@@appendixproof	337, 346, 392
\@@axp@newtheorem	<u>82</u> , <u>129</u>
\@@section	<u>444</u>
\@acmplainnotefont	254, 266, 267
\@acmttitlebox	533
\@appendixproof	337, 338, 392
\@auxout	470
\@axp@newtheorem	<u>82</u> , <u>129</u>
\@begintheorem	253
\@biblabel	552, 553
\@bsphack	194
\@enddocumenthook	493
\@esphack	207
\@extra@binfo	496
\@ifnextchar	337, 392, 434, 443
\@ifpackageloaded	7
\@ifstar	82, 445
\@namedef	496
\@ne	190
\@noligs	202
\@oldbiblabel	552, 553
\@oparg	94, 97
\@section	<u>444</u>
\@thmcountersep	144
_	36, 45, 53, 62, 105
\^	337, 362, 391, 392
A	
\ACM@origsection	146, 147
\active	337, 362, 391, 392
\advance	190
\appendix	500
appendix (option)	4
\appendixbibliographyprelim	5, <u>100</u> , 524, <u>551</u>
\appendixbibliographystyle	5, <u>100</u> , 502
\appendixclaimproof	392, 396, 404
\appendixprelim	5, <u>100</u> , 499
\appendixproof	164, 337, 366, 374
appendixproof (env.)	2, <u>160</u> , <u>337</u>
\appendixrefname	5, <u>100</u> , 508, 510, <u>545</u>
\appendixsectionformat	5, <u>100</u> , 474
\apx@oldamsthmndproof	10, 17
\apx@oldamsthmproof	9, 15, 16
\AtBeginDocument	147, 171, 175, 387
\axp@@section	446, 447, 448
\axp@claimproof	391, 401
\axp@@proof	362, 371
\axp@@sectiontestinput	<u>433</u>
\axp@@sectiontestsection	<u>433</u>
\axp@appendix	24, 29, 31, 33, 36, 110, 149, 291, 485
\axp@bibliography	25, 39, 42, 45, 486, 501, 506
\axp@claimproof	391, 393
\axp@equation	<u>213</u>
\axp@equationx	<u>213</u>
\axp@forward	<u>307</u>
\axp@forward@link	297, <u>307</u>
\axp@forward@setup	219, <u>331</u>
\axp@forward@suppress	56
\axp@forward@target	277, <u>324</u>
\axp@forwardlinking	27, 57, 58
\axp@newtheorem	73, <u>82</u> , 218
\axp@newtheoremrep	97, <u>150</u> , <u>215</u> , 215
\axp@newtheoremrep@definetheorem	65, 151, 216
\axp@newtheoremreptmp	93
\axp@noappendixbibliography	517, 522
\axp@oldbibliography	<u>486</u>
\axp@oldclaimproof	180, 389, 398, 406
\axp@oldlabel	111, 114, 115, 119, 122, 124
\axp@oldlabel@in@display	112, 118, 123, 125
\axp@oldproof	106, 178, 368, 376, 427, 429
axp@oldproof (env.)	<u>106</u>
\axp@oldsection	182, 443, <u>444</u> , 484
\axp@oldtableofcontents	483
\axp@oldthebibliography	512
\axp@proof	362, 363
\axp@proofsfile	<u>171</u> , 176, 200, 206, 237, 239, 249, 284, 341, 349, 357, 420, 471, 504
\axp@redefinlabels	<u>108</u> , 154, 156, 278
\axp@refstar	245, 247, 262
\axp@repeqn	26, 48, 50, 53, 62, 221, 238, 283
\axp@rpcounter	<u>212</u>
\axp@seccounter	<u>431</u>
\axp@section	<u>433</u> , 473
\axp@sectitle	<u>432</u> , 449, 459, 462, 477
\axp@seenreptheorem	<u>211</u>
\axp@tmp	439, 441, 464, 470, 472, 515, 516
\axp@tmpptmp	440, 441
\axp@unactivateeightbit	<u>185</u> , 196

<code>\xpp@VerbatimOut</code>	209, 344, 352	<code>toappendix</code>	1, 166, 208
<code>xpp@VerbatimOut (env.)</code>	192	<code>\errmessage</code>	36, 45, 53, 62
<code>\xpp@with@forward</code>	226, 228, 290	<code>\expandonce</code>	466, 470, 472, 476, 477
<code>\xpp@writesection</code> 209, 236, 340, 348, 461			
B			
<code>\bfseries</code>	132, 134, 137, 141	F	
<code>\bibtex</code>	495, 528	<code>forwardlinking (option)</code>	5
<code>\bibliography</code>	486	<code>\FV@CatCodesHook</code>	203
<code>bibliography (option)</code>	4	<code>\FV@DefineTabOut</code>	199
<code>\bibname</code>	509, 510	<code>\FV@DefineWhiteSpace</code>	197
<code>\BODY</code>	154, 156, 231, 235	<code>\FV@FontScanPrep</code>	201
<code>\boolfalse</code>	305	<code>\FV@ProcessLine</code>	200
<code>\booltrue</code>	60, 292	<code>\FV@Scan</code>	204
		<code>\FV@Space</code>	198
		<code>\FVB@xpp@VerbatimOut</code>	192
		<code>\FVE@xpp@VerbatimOut</code>	192
C			
<code>\CatchFileDef</code>	439	G	
<code>\catcode</code>	188, 337, 362, 391, 392	<code>\gdef</code>	517
<code>\claimproof</code>	180, 388, 389, 391		
<code>claimproof (env.)</code>	387	H	
<code>\clearpage</code>	104	<code>\hspace</code>	553
<code>\closeout</code>	504	<code>\hyperlink</code>	313
<code>\count@</code>	186, 188, 189, 190	<code>\hypertarget</code>	326
<code>\csdef</code>	296		
<code>\csedef</code>	332	I	
<code>\csletcs</code>	294, 295, 303	<code>\ifblank</code>	332
		<code>\ifbool</code>	244
D			
<code>\DeclareStringOption</code>	24, 25, 26, 27	<code>\ifboolexpr</code>	311
<code>\defaultbibliography</code>	489	<code>\ifcsdef</code>	302, 312, 332
<code>\DefineVerbatimEnvironment</code>	192	<code>\ifcsname</code>	68, 74, 325
		<code>\ifcsundef</code>	293
		<code>\ifnum</code>	189
E			
<code>\empty</code>	516	<code>\inlineproof</code>	160, 427
<code>\endappendixclaimproof</code>	412, 417	<code>inlineproof (env.)</code>	3, 160, 427
<code>\endappendixproof</code>	165, 354, 382	<code>\input</code>	434, 438, 505
<code>\endxpp@oldclaimproof</code>	181, 390, 414	<code>\itshape</code>	132, 134, 137, 141, 538
<code>\endxpp@oldproof</code> 107, 179, 384, 428, 430			
<code>\endxpp@oldthebibliography</code>	513	L	
<code>\endxpp@VerbatimOut</code>	210, 356, 419	<code>\label</code> 111, 113, 122, 124, 226, 228, 456	
<code>\endclaimproof</code>	181, 390, 409	<code>\label@in@display</code> 109, 112, 117, 123, 125	
<code>\endinlineproof</code>	161, 428	<code>\lipics@opterrshort</code>	550
<code>\endnestedproof</code>	163, 430	<code>\loop</code>	187
<code>\endproof</code>	10,	<code>\ltx@ifnextchar@nospace</code>	362, 391
	12, 17, 107, 161, 163, 165, 179, 379		
<code>\endthebibliography</code>	513	M	
environments:		<code>\mainbodyrepeatedtheorem</code>	5, 100, 230
<code>appendixproof</code>	2, 160, 337	<code>\message</code>	30, 32, 34, 41, 43, 49, 51, 59
<code>xpp@oldproof</code>	106		
<code>xpp@VerbatimOut</code>	192	N	
<code>claimproof</code>	387	<code>\NAT@testdef</code>	494, 527, 528
<code>inlineproof</code>	3, 160, 427	<code>\nestedproof</code>	162, 429
<code>nestedproof</code>	5, 160, 427	<code>nestedproof (env.)</code>	5, 160, 427
<code>proof</code>	3, 175, 362	<code>\newbool</code>	56, 307
<code>proofsketch</code>	3, 99	<code>\newtheorem</code>	85, 87, 91
		<code>\newtheoremrep</code>	2, 93
		<code>\newtheoremstyle</code>	535

<code>\noproofinappendix</code>		S	
.....	6, 166, 217, 334, 360, 423	<code>\scshape</code>	540
<code>\nosectionappendix</code> ...	6, 166, 458, 480	<code>\section</code>	147, 175, 443, 444, 484
		<code>\setcounter</code>	222, 240, 241, 285
O		<code>\SetupKeyvalOptions</code>	20
<code>\onecolumn</code>	104	<code>\small</code>	553
<code>\openout</code>	173	<code>\spnewtheorem</code> ..	128, 132, 134, 137, 141
options:		T	
appendix	4	<code>\tableofcontents</code>	483
bibliography	4	<code>\texorpdfstring</code>	309, 310
forwardlinking	5	<code>\the</code>	255, 270
repeqn	5	<code>\theexp@equation</code>	241
P		<code>\thebibliography</code>	512, 545
<code>\pretocmd</code>	217, 253, 493	<code>\theopargself</code>	250, 251, 264
<code>\ProcessLocalKeyvalOptions</code>	28	<code>\theoremstyle</code>	544
<code>\proof</code>	9,	<code>\thm@notefont</code>	255, 269, 270
11, 16, 106, 160, 162, 164, 178, 362		<code>\thmhead</code>	254, 255, 256, 257
<code>proof (env.)</code>	3, 175, 362	<code>toappendix (env.)</code>	1, 166, 208
<code>\proofsketch</code>	143	<code>\togglefalse</code>	335
<code>proofsketch (env.)</code>	3, 99	<code>\toggletrue</code>	232
<code>\providecommand</code>	101, 109	U	
<code>\putbib</code>	525	<code>\upshape</code>	141
R		V	
<code>\ref</code>	245, 247, 467	<code>\value</code>	222, 240, 285
<code>\refname</code>	507, 508, 545	W	
<code>\repeat</code>	191	<code>\write</code> ..	176, 200, 206, 237, 239, 249,
<code>repeqn (option)</code>	5	284, 341, 349, 357, 420, 470, 471	

References

- [1] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison–Wesley Pub. Co., Reading, MA, 1986.

A Proofs for Section 1 (Usage)

Proof.

□

Foobar 2. *This foobar is repeated in the appendix.*

Proof.

□