

BLUe's Reports

Good Work

Kees van der Laan

Hunzeweg 57,
9893 PB Garnwerd, The Netherlands
cgl@rc.service.rug.nl

Abstract

The backgrounds, use, design, and coding of BLUe's Reports format have been discussed. The purpose is to format a report as a compatible extension of the default note format of BLUe's format. As such it is an example of how to customize BLUe's format system with a format of your own. The differences have to do with

not only using titles but also reusing the titles for table of contents, list of examples, and running headlines the preliminary pages and cover pages which are absent with the default note format appendices may contain collected information from the script such as answer to the exercises (similar as in *The T_EXbook*), table of examples, table of contents, and an index.

Keywords: Active list separators, compatible extension, databases, data integrity, education, fifo, house style, lazy evaluation, list element tag, macro writing, no-nonsense, on the fly, pattern matching, plain T_EX, reports, reusable software parts, separation of concerns, set macros, software engineering.

1 Introduction

What makes a report different from a note, form an article? What is the added value? BLUe's format default note style allows `\onecol`, but there is a lot more to be added if we really want to format a script in such a way that a report will emerge. The following list summarizes the differences.

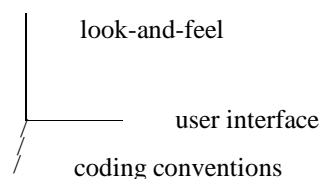
- The front matter is more elaborate (preface page, title page, abstract page, foreword page, contents page).
- The copy proper is divided into chapters, with the tree of sections and subsections underneath, and appendices as special chapters. The latter contain for example dynamically generated tables of contents, examples, figures, etc, and an index. Also answers to exercises can be collected in an appendix.
- The cover matter (front, back and the inside covers).

This sets the scene when we like T_EX to take care of it all.¹

¹The advantage of handling all by T_EX is that we don't have to bother with jargon from other tools. Another advantage is the stability, because T_EX has been frozen.

²Of course. . . .

When designing a format the following three issues have to be dealt with, preferably as a mutual orthogonal set.



In the sequel I'll talk about the

- use (and customization)
- look-and-feel
- user interface, and
- coding.

For the `\report` format the new aspect is the look-and-feel of a report, because the coding conventions have already been decided upon in BLUe's format, ditto for the user-interface philosophy.

The more complex codings are a result from the

- two-part macros as basis
- one-part macros on top
- storing of the 'arguments' with the right catcodes, and
- writing of titles to a file with page numbers attached.

The creation of an index in one-pass is not special for `\report` and has been treated in 'BLUe's Index.' The use of `\sortindex` and `\pasteupindex` has been explained in 'Publishing with T_EX,' the user's guide for BLUe's format system.

1.1 Notations and definitions

`\ea` denotes `\expandafter`. `\nx` denotes `\noexpand`.

An important notion is what I call 'list element tag.' This element is a kameleon and takes any actual meaning when processing the list elements. It comes close to the concept of an active list separator. Knuth used this mechanism already² in *The T_EXbook* Appendix D.2 List macros, the `\s`, and stressed the potential with the following remark. 'But in fact, the `\s` separators are enormously useful, because we can define `\s` to be any desired one-argument macro, and then we can execute the list!' I prefer the name list element tag instead of (active) separator, because it does not really separate — the first element is different. Note that the idea is not restricted to one argument. The list element tag has been applied in for example `\contentsdatabase` with the list element tag `\def`. Agreed, the use of a list element tag is unusual, but once you get the hang of it, you can read through it, and you will recognize the essential issues.³ Knuth used the list element tag, `\ansno`, for formatting the answers of the exercises for *The T_EXbook*. `\ansno` takes two arguments ended by period. and colon: respectively, and followed by the answer proper, typeset on the fly.

2 Use

First a disclaimer. A report format is only handy when there are chapters, and when we like to see the preliminary pages in print as well. If not, stay with the default note format and switch to `\onecol`. This prompts for proofing via the default format.

3 The markup of a report

Example (*Just a chapter*)

To typeset a chapter, forget about `\beginscript`, which typesets the cover, and the preliminary pages.⁴

```
\input blue.tex \report
\chapterhead{...}
<copyproper>
\bye
```

Example (*Template for a report script*)

```
\input blue.tex
%\loadindexmacros%When creating an index
\report           %Preliminary matter follows
\title{...}
\subtitle{...}
\keywords{...}
\abstract{...}
\bibliographydata{ISBN: ...}
\acknowledgements{...}
\pictures{...}
\preface{...}
\begincontents%\se is subentry descriptive tag
  Preface
```

³The history is that I started with loading all the references from my literature database. At that time T_EX definitions were appropriate as entries. Only when I considered loading selectively from the database, I found using the `\def` as list element tag. Later I realized the vulnerability of `\def` especially when the OTR can come in between. The is heavily used for e-o-l so I can't follow Knuth in that. Now I use `\lst`.

⁴Ditto for `\endscript`, which takes care of the back cover.

⁵If you don't like this static contents page idea, let `\pasteupcontents` equal `\relax`.

```
Table of contents
Introduction
\se BLUe's format
...
\se J: Table of Contents
\endcontents
%\let\coverpic=\<name>pic
\beginscript
\input pwt.int   %Introduction
...
\setupappendices %Appendices A, B, C,...
\pasteupanswers %Answer to Exercises
...
\pasteuptoe      %Table of Examples
%\sortindex\pasteupindex
\pasteuptoc      %Table of Contents
\endscript
```

In the format the necessary files have been opened via

```
\immediate\openout\<name>=<filename>
```

with for example `<name> = ans, toc, toe`, and `<filename> = answers, contents, and examples`, respectively.

3.1 Proofing a single chapter

Chapters can be proofed separately as follows.

```
\input blue.tex           %\loadindexmacros
\report                   %After \loadindexmacros
\input <filewithchapter>%\sortindex\pasteupindex
\bye
```

We can easily test a few chapters in a row as follows. Omit `\bye` and when T_EX prompts for an `\end` insert 'i' for insert and after the insert prompt supply `\input <nextchapter>`. The testing of index entries per chapter on the fly is convenient.

4 Customization

Remember that BLUe's format system is a personalized format, it knows about you. Candidates for customization are the preliminary pages and the back cover. For the front cover a picture can be assigned to `\coverpic`. The re-design of `\beginscript` and `\endscript` is beyond the scope of this paper.

5 Look-and-feel

The design in broad terms yields a cover page (with title matter and cover picture), an inside cover (with keywords, bibliography information, acknowledgements), a title page, (a translator page if relevant), a preface page, and a contents page. This static contents page can assist the writing as such, to maintain the survey. `\report` also generates a table of contents on the fly with page numbers attached.⁵

Details of the layout are treated along with the coding.

6 User interface

The approach initiated in BLUe's format of providing information via token variables⁶ in the preliminary part has been followed. The `\beginscript` typesets the preliminary pages.

The appendices are set via `\pasteup<name>` now and then, to abstract from details. `\setupappendices` provides for letters as numbering.

Another problem I had to face was how to cope with names. In BLUe's format I used as principle a 'root with affixes.' I kept to that approach although not unduly. I tried to remain consistent but at the user level the names should not be too long. At the lower level it is nice to have long and systematic names, which prevents unintended double meanings.

7 Coding

In BLUe's format two-part macros are the starting point. A one-part macro is provided on top of it. The aim is to allow processing on the fly.⁷ Specification of information and actual typesetting have been separated for matter which does not belong to the copy proper, such as abstract, acknowledgements, keywords, preface and the like.

7.1 New control sequences

Examples of new control sequences are for example those associated with `\preface` and `\chapterhead`.

Preface matters

The preface can be specified via

```
\beginpreface
...           or       \preface{...}
\endpreface
```

The coding reads

```
\def\beginpreface{\setbox\prefacebox
  \vbox\bgroup}
%
\def\endpreface{\egroup}
%
\def\preface#\{\beginpreface\bgroup
  \aftergroup\endpreface
  \afterassignment\ignorewhitespace
  \let\dummy=}
%
\def\pasteuppreface{\centerline{\chapterfont
  \the\prefacename}\bigskip
  \unvbox\prefacebox}
```

The one-part macro on top is apart from some checking and neglecting of white space, equivalent to

```
\def\preface{\aftergroup\endpreface
```

⁶A white lie, it looks like it. . . .

⁷When arguments are not processed on the fly the catcodes of the tokens of the argument are set while read by T_EX's parameter scanning mechanism. That can be different from what is meant. An example is the circumflex in in-line math. In-line verbatim occurs much especially when T_EXies write in T_EX about T_EX. Although I don't like use of different fonts in a heading, I nevertheless provided for the opportunity to do so.

⁸My 'Paradigm: Headache?' was complex enough, it paid attention to part of it, neglecting the storing and reusing aspects!

```
\beginpreface}
```

Although the use via `\preface{...}` might suggest that catcodes will be assigned too early to the argument tokens, this is not the case, due to the special coding. To understand the coding these kinds of mechanisms must be known and recognized. The preface is pasted up in `\beginscript`.

Chapter head matters

The required functionalities — the specs so to say — are to

- start a new page
- store the title for reuse (running head, table of contents)
- maintain and store the chapter counter (for use with the exercises, table of contents, and the like)
- write the title and title counter value to the contents file
- typeset the title on a new page with some white space after
- reset counters which start anew per chapter (headings, exercise number, and the like)
- provide the right `\headline` and `\footline` with the title, chapter title, and page number
- suppress the headline on the first page of a chapter (or supply for an otherwise special headline)
- process the title on the fly to allow for example verbatim
- condition the writing of the title and title counter value to the examples file.

Quite something, isn't? That is why title matters are not trivial.⁸ The conditioning of writing the title to the examples file is there, because we don't like the title etc. to appear in that file if the chapter lacks examples.

```
\def\beginchapterhead{\the\prechapterhead
  \storechaptertitle}
%
\def\storechaptertitle#1\endchapterhead{%
  \global\chaptertitle={#1}\endchapterhead}
%
\def\endchapterhead{\centerline{\chapterfont
  \the\chapternumbering\the\chaptertitle}
  \the\postchapterhead\ignorewhitespace}
```

This approach looks promising. What has to be done before is the concern of `\prechapterhead`. `\storechaptertitle` does what its name suggests. The formatting of the number of the chapter is the concern of `\chapternumbering`. `\postchapterhead` takes care of leftovers. Default `\chapternumbering` is empty, that is, no number will be typeset.

Undoubtedly, the careful reader would object that the title is stored and that the catcodes are set before processing. That is right, of course, but via `\everychapterhead` —

in `\prechapterhead` — we can guarantee that symbols will have the right catcodes. `\storechaptertitle` is there to allow the short variant `\chapterhead{...}`. The combination of storing and in-line verbatim does not work. Beware.⁹

```
\prechapterhead{\newpage\null\vskip4pc
  \bgroup\catcode\^=7
  \the\everychapterhead\the\thischapterhead
  \global\advance\chaptercnt1 }
%
\postchapterhead{\egroup
  \headcnt0 \exercisecnt0 \examplecnt0
  \write\toc{\nx\separator
    {\alfanum\the\chaptercnt}
    {\the\chaptertitle}}
  %suppress headline first page chapter
  \headline{\global\headline
    {\hbox to\pagewd
      {\sl\the\chaptertitle\hfill
        \the\title}}}%
  \footline{\hbox to\pagewd{\small
    \rlap{Draft \today}\hfill\dash
    {\oldstyle\the\pagenumber}\dash
    \hfill\llap{\copyright
      \the\crownr}}}%
  %Example is redefined here such that on
  %first use if ever in the chapter, it
  %redefines itself into the regular macro,
  %while writing the chapter titles to the
  %table-of-examples, ToE, file.
  %In this way chapter titles of chapters
  %without examples will not be written to
  %the ToE.
  \def\example{\write\toe{\nx\separator
    {\alfanum\the\chaptercnt}
    {\the\chaptertitle}}
    \let\example\regularexample
    \example}
  \thischapterhead{\vskip3pc}
```

The material is enclosed within a group to keep the effect of `\thischapter` local.

`\example` is defined such that only when it is used the title and number of the chapter will be written to the file associated with `\toe`. `\regularexample` writes the example title and number to the file associated with `\toe`.

Intriguing is `\alfanum`. The purpose is to control whether a number or a letter should appear. This is particularly useful with appendices.

```
\def\setupappendices{\thischapterhead=
  {\chaptercnt64 \gdef\alfanum{\char}%
  \global\chapternumbering=
  {\alfanum\the\chaptercnt: }}}
```

And what about the one-part macro on top? The above was already complicated. What is the problem for a one-part macro? For most of the cases we can do with a simple variant. However, when we like to process math, especially the circumflex, as part of the title more work has to be done. The mechanism has been explained in ‘Paradigm: Two-part macros.’

⁹For me this does not hinder because I don't like these kinds of font changes in a chapter header.

¹⁰The circumflex in math in the title will hiccup. Most of the time we don't need that.

¹¹What does on-the-fly mean anyways, when we have to store the title for multiple use? The answer is that we would like to allow for math too. No verbatims however.

```
\def\chapterhead{\bgroup
  \def\storechaptertitle##1{\egroup
    \global\chaptertitle{##1}%
  \endchapterhead}\beginchapterhead}
```

As a tribute to `manmac` I also defined the following limited variant.¹⁰

```
\def\bluechapter#1\par{\beginchapterhead
  #1\unskip\endchapterhead}
```

A trivial user interface on top which counteracts the curly braces mania at the expense of processing on the fly.¹¹

7.2 Redefined control sequences

Important are the redefinitions of the various headers, and the `\beginscript` and `\endscript` macros. Details are the redefinitions induced by my making `\parindent` equal to zero, while I wish to retain indented `\items` and their derivatives. For that purpose I had to redefine `\hang`, `\itemitem`, and `\textindent`. I also had to adjust `\exercise`, `\answer`, and `\ansno` from `manmac` to this environment.

Pasteup matters

I do like these macroscopic control sequences very much. Their use is intuitive.

```
\def\pasteuptoc{%
  \chapterhead{Table of \the\contentsname}
  \immediate\closeout\toc%Table Of Contents
  {\ninepoint\def\contentsno##1.##2:
    {\noindent\hbox to\generalindent
      {\oldstyle##1}.\oldstyle##2}\hss}}
  \def\separator##1{\smallbreak\noindent
  \hbox to\generalindent{\oldstyle##1}\hss}}
  \def\she{\noindent\hbox
    to1.5\generalindent{\hss}}
  \def\sshe{\noindent\hbox
    to2\generalindent{\hss}}
  \parindent0pc\obeylines\sl
  \input contents\relax}}
```

Note the use of control sequences for language-dependent names. The list element tag, `\contentsno`, gets its meaning, next to other quantities left undefined so far. The contents proper is processed on the fly. Only the numbers are parsed as arguments.

Note that next to the dynamically generated ToC, there is the possibility to pasteup a static ToC, via `\pasteupcontents`. The control sequences `\begincontents` and `\endcontents` have been redefined in order to set the static contents via a `\valign`. No short variant is possible for these.

Header matters

Header macros have to be redefined because the header titles are also used in the the table of contents. In contrast with the default `blue.tex` the headers maintain a number, to

be used in the table of contents. Typical for the class of headers are the macros associated with `\head`. Because of the storing of the head title, the title is not processed on the fly. The one-part macro is not straightforward, because I allowed for adjusting catcodes via `\everyhead`.¹² This is handy when we like also math as part of the title.

```
\def\beginhead{\the\prehead
  \global\advance\headcnt1
  \storeheadtitle}
%
\def\storeheadtitle#1\endhead{%
  \global\headtitle=#1\endhead}
%
\def\endhead{\centerline{\headfont
  \the\headtitle}\the\posthead}
%
\posthead{\egroup\xdef\writetoc{%
  \write\toc{\contentsno\nx\alphanum
  \the\chaptercnt.\the\headcnt:
  \the\headtitle\nx\nx\nx
  \pagenorepresentation{\nx\number
  \pageno}}}\writetoc
  \nobreak\medskip\noindent
  \ignorewhitespace}
%and adapted from blue.tex the default
\prehead{\vskip0pt plus9ex
  \penalty-250\vskip0pt plus-9ex
  \bgroup\catcode'\^=7
  \the\everyhead\the\thishead
  \bigskip\noindent}
```

`\prehead` is left invariant, that is, the discouraging of a page break after a heading has been left invariant, as the amount of white space before the heading. The catcode of the circumflex has regained its original value to allow for math as part of the title. The group with `\egroup` in `\posthead` is inserted to localize the effect of the category change of the circumflex, and the inserted values of `\everyhead` and `\thishead`. Note that `\contentsno`, the list element tag, will perform a similar function as Knuth's `\ansno`, namely, to process the list of elements contained in the file contents which is associated with `\toc`.¹³ Also the markup for the formatting of the table of contents has been abstracted in descriptonal markup. This is related to the concept of lazy evaluation.

Note too that no `\immediate \write` can be used, because of the asynchronism of the OTR. The page number expansion must also be delayed and done in the OTR.

The one-part macro reads as follows.

```
\def\head{\bgroup
  \def\storeheadtitle##1{\egroup
  \global\headtitle{##1}\endhead}\beginhead}
```

Script matters

`\beginscript` takes care of typesetting the preliminary pages, and `\endscript`'s concern is to format the back cover. Within the approach taken the coding is not difficult, just tedious. I tended to ignore an element when not filled-in by a user.

```
\def\beginscript{\lastscript\begingroup
```

¹²It is analogous to the one part macro `\chapterhead` and `\example`.

¹³It is not expanded because at the moment of writing to a file it is equal to `\relax`.

```
\the\everyscript\the\thisscript
\pageno-1 \headline{\}\footline{}
\ifproof\rightline{\rlap{\sl Cover}}\fi
\kern7\bigskipamount
\centerline{\chapterfont\the\title}
\bigskip
\centerline{\subtitelfont\Dash
  \the\subtitle\Dash}
\vskip\bigskipamount
\centerline{by}
\vskip\bigskipamount
\centerline{\subtitelfont\the\author}
\vfill
\ifx\coverpic\undefined
  \else$$\coverpic$$\fi
\vfill
\if!\the\translator!
  \else\rightline{\subtitelfont
  Translated by \the\translator}\fi
\newpage
\ifproof\rightline{\rlap{\sl Inside
  Cover}}\fi
\kern3\bigskipamount
\pasteupkeywords
\smallskip
\the\bibliographydata
\vfill\vfill
\pasteupacknowledgements
\newpage
\ifproof\rightline{\rlap{\sl Title
  Page}}\fi
\kern7\bigskipamount
\centerline{\chapterfont\the\title}
\bigskip
\centerline{\subtitelfont\Dash
  \the\subtitle\Dash}
\vskip\bigskipamount
\centerline{by}
\vskip\bigskipamount
\centerline{\subtitelfont\the\author}
\vfill\pasteupabstract\vfill\vfill
\ifvoid\forewordtranslatorbox
  \else \newpage
\ifproof\rightline{\rlap
  {\sl Foreword Translation}}\fi
\vfill
\unvbox\forewordtranslatorbox
\vfill
\rightline{\vtop{\hbox{\the\translator}
  \hbox{\the\author}}}\vfill
\fi
\ifvoid\prefacebox\else\newpage
\ifproof\rightline{\rlap{\sl Preface
  Page}}\fi
\vfill\pasteuppreface\vfill\vfill
\fi
\newpage
\ifproof\rightline{\rlap{\sl Contents
  Page}}\fi
\vfill\pasteupcontents
}%end \beginscript
%
\def\endscript{\xcol=\maxcols
  \vfill\eject\endgroup\backcover
  \tracingstats=1 \stop\thisscript{}}
```

`\backcover` is default `\relax`. For an example see `\pwtbackcover` in the `fmt.dat` file.

Example matters

What are the required functionalities? I could think of the following.

- no page break immediate after the heading
- maintain an example counter
- write the counter value and the title of the example to the file examples which is associated with `\toe`, and attach the page number to it, for a proper list of examples¹⁴
- typeset the example name, and the title
- it should be possible to have in-line math as part of the title.

Together with `\report` I had to do something about the circumflex, because when expanded during the writing to the file it yielded an error messages. The easiest thing was to make it equal to `\relax` locally. However, what about the asynchronism of the OTR? No problem, because when it is read in `\pasteuptoe` it will be processed as in ordinary text. It is just the expansion while writing to a file which has to be suppressed. Despite that I chose at last for a local catcode change.

I could not provide for storing the title and also allowing for verbatims to be handled appropriately.

```
\def\beginexample{\vskip0pt plus5ex
\penalty-100\vskip0pt plus-5ex
\medskip
\global\advance\examplecnt1
\bgrou\catcode'\^=7 \the\everyexample
\the\thisexample\storeexampletitle}
%
\def\storeexampletitle#1\endexample{%
\global\exampletitle{#1}\endexample}
%
\def\endexample{%
\xdef\writeexa{write\toe{\exampleno
\alfanum\the\chaptercnt.\the\examplecnt:
\the\exampletitle
\nx\nx\nx\pagenorepresentation
{\the\count0}}}\writeexa
{\the\examplename} {\sl(\the\exampletitle)}
\egroup
\nobreak\endgraf\noindent\ignorewhitespace}
```

For consistency reasons the following has been added.

```
\def\example{\bgrou
\def\storeexampletitle##1{\egrou
\global\exampletitle{##1}\endexample}
\beginexample}
```

As a tribute to manmac the following limited variant has been added. No math with the circumflex is allowed.¹⁵

```
\def\blueexample#1\par{\example{#1\unskip}}
```

8 ToC tags

During this work I realized that many a tag should be included in the default `blue.tex`. An example of the latter is `\summary`. I also extended the font switching macros with `\large` and `\Large`. Both are also included in `blue.tex`.

`\large` is used with transparencies, and `\Large` is in use for typesetting chapter headings.

For the list of tags I refer to ‘Publishing with T_EX.’

The macros are part of `fmt.dat` — BLUE’s format database of formats — and can be listed with appropriate page numbers by `gfile.tex`. A table of contents of the macros borrowed from `fmt.dat` has been included. Make your crib out of it.

```
%Table of Contents report format
%Token vars (declared in blue.tex)
% \prefacename
% \bibliographydata
% \chaptertitle
% \prechapterhead
% \postchapterhead
% \headtitle
% \subheadtitle
% \subsubheadtitle
% \prepoint
% \prefil
% \postfil
%Box vars (declared in blue.tex)
% \prefacebox
% \contentsbox
%New dimens (declared in blue.tex)
% \generalindent
% \gutter
% \xshift
% \yshift
% \subsubheadindent
%New counts (declared in blue.tex)
% \headcnt
% \chaptercnt
% \examplecnt
% \exercisecnt
%New writes (declared in blue.tex)
% \toc
% \toe
%Fonts
% \chapterfont\Large.....4
% \subtitlefont.....5
%Chapter title matters
% \chapno (manmac)
% \prechapterhead.....102-109
% \postchapterhead.....111-138
% \beginchapterhead....140-141
% \storechaptertitle....143-144
% \endchapterhead.....146-148
% \chapterhead.....150-154
% \setupappendices.....156-159
% \pasteupanswers.....161-165
% \pasteuptoe.....167-176
% \pasteuptoc.....178-191
% \pagenorepresentation.193-194
%Summary
% \beginsummary (blue.tex)
% \endsummary (blue.tex)
% \summary (blue.tex)
% \beginquote.....252-254
% \endquote (blue.tex)
%Header matters
%Head
% \beginhead.....302-304
% \storeheadtitle.....306-307
% \endhead.....309-310
% \head.....312-315
% \posthead.....317-324
```

¹⁴ A detail in here is to provide for a toggle whether a number or a character should show up in the list.

¹⁵ When no index is wanted the circumflex can be assigned category code 7 at the beginning of the script and all is fine.

```

%Subhead
% \beginsubhead.....351-353
% \storesubheadtitle...355-356
% \endsubhead.....358-359
% \subhead.....361-364
% \postsubhead.....366-371
%Subsubhead
% \beginsubsubhead.....401-403
% \storesubsubheadtitle.405-407
% \endsubsubhead.....409-410
% \subsubhead.....412-417
% \postsubsubhead.....419-424
%Inner markup
% \begindemo (blue.tex)
% <escapechar>yields (blue.tex)
% \enddemo (blue.tex)
% \beginexample.....526-531
% \storeexampletitle...533-534
% \endexample.....536-544
% \example.....546-549
% \hang.....553
% \itemitem.....555-557
% \textindent.....559-561
% \exercise.....576-581
% \answer.....583-587
% \ansno.....589-592
%Generalized \samplebox
% \point (blue.tex)
% \gbox (blue.tex)
%Centering verbatims
% \boxlines (blue.tex)
% \begincenterverbatim(blue.tex)
% \endcenterverbatim (blue.tex)
%Preliminary pages matter
% \def\beginscript.....703-772
%Closing pages matter
% \endscript.....901-903
% \pwtbackcover.....905-929
%Auxiliaries
% \beginkeywords.....802-803
% \preacknowledgements..805-806
% \pasteupacknowledgements808/812
% \beginpreface.....814-815
% \endpreface.....817
% \preface.....819-822
% \pasteuppreface.....824-825
% \beginforewordtranslator827/9
% \endforewordtranslator...831
% \forewordtranslator...833-836
% \pasteupforewordtranslator838/9
% \begincontents.....841-847
% \endcontents.....849
% \pasteupcontents.....851
% NTG info (tools.dat)
%Defaults.....1000-10xx
% (\she.....1037
% \sshe.....1038)
%end Table of Contents

```

9 Acknowledgements

Many a suggestion with respect to simple markup tags in the user interface was done by Erik Frambach. Jos Winnink proofed the article, as usual. Thank you!

10 Conclusion

BLUe's `\report` format emerged because I needed it to format the user's guide 'Publishing with T_EX.'

An important observation is that titles are reused a lot, and have to be stored. For storing I used the mechanism of

- a box and paste up later
- toks variables, with the right category codes, and combined with
- writing to a file and including the file later.

The storing via an `\insert` similar to plain's footnotes is perhaps the way to allow for storing and processing on the fly, although the `\insert` mechanism has been designed with the OTR processing in mind. This has been postponed because I don't need nor wish verbatims in titles. In footnotes, yes I do, and that is very convenient. For marginnotes it might be relevant too. More on this later in 'Paradigm: Eating and preserving.'

Writing to a file and attaching page numbers to the entries needs awareness of the asynchronism of the OTR, in actual fact that a `\write` must be used and not an `\immediate \write`.

In one week I could write and format the articles 'BLUe's format Databases,' 'BLUe's Letters,' and 'BLUe's Reports,' next to polishing a little on the formats in BLUe's format system. The last article was correctly formatted immediately.¹⁶

At last, I reached the stage which I had mastered before with other programming languages: a blue script runs correctly when proofed for the first time.

After all those years I feel fairly confident with T_EX.

References

The T_EXbook and L^AT_EX user's guide are omni-present and not explicitly listed. For my works consult `lit.dat`, via the use of `\search` for example.

¹⁶The bad news is that polishing the contents of the articles did cost me another couple of weeks. That has nothing to do with T_EX, but all with the writing process as such, if not with the learning on the fly.