NCCTOOLS

by Alexander I. Rozhenko

Release 3.5

The ‘ncctools’ collection consists of a number of packages extracted from NCC style (developed by Alexander I. Rozhenko in 1992–1996 under LaTeX-2.09) while re-implementation it for LaTeX 2ε. Many new packages were also added later.

The collection now contains 25 packages providing the following:

`afterpackage`	Implements `\AfterPackage` command specifying additional commands which should be applied to a package after its loading. The customization commands are specified without pre-loading packages they concerned to and are applied right after loading the respective packages.
`dcounter`	Dynamic counters. The counter declared as dynamic is really created at the first use and receives at that moment the count style established by the `\countstyle` command. The special use of `\counstyle` command with optional parameter allows modify the subordination of existing counter. For example, while using the book class you can reject the subordination of the section counter to the chapter counter and re-subordinate figures, tables and equations to sections. The package is used in the `nccthm` package.
`desclist`	Implements the `desclist` environment. It is considered as an improvement of the `description` environment. The appearance of item markers is easy customizable on the fly. An optional parameter allows set a marker prototype for calculation of hang indentation skip. The `description` environment is redefined to use an optional parameter also.
`extdash`	The package implements the commands, `\Hyphdash`, `\Endash`, `\Emdash`, and their star-forms, to control hyphenation of compound words and ordinary words dashed by em-dash. You can also use the shortcuts `\-/` `\=/` `\--` `\==` `\---` `\===` instead. You can also decrease the length of em-dash by the `cyremdash` option to satisfy the Russian typesetting rules.
`manyfoot`	The package implements a command, `\newfootnote`, that adds footnote levels to the standard LaTeX’s footnote mechanism. Footnotes of every additional level are automatically grouped together on a LaTeX 2ε output page and are

separated from another levels by the special vertical space and (maybe) rule. You can customize the typesetting style of additional footnotes choosing between ordinary footnotes and run-in paragraph footnotes (useful for critical editions). Service command `\DeclareNewFootnote` simplifies creation of new footnote levels with automatic footnote numbering. The possibility of customization inter-level footnote rules is allowed.

`mboxfill`	The package introduces `\mboxfill` command filling a free space with a pattern. All leader types are supported. Width of pattern can be specified by the same manner as in the `\makebox` command.
`nccbbb`	Implementation of poor Black Board Bold symbols. Ported from old NCC-LaTeX. It is useless in modern LaTeX but kept just in case.
`nccboxes`	Additional boxes from NCC-LaTeX. The `\jhbox` and `\jvbox` horizontally and vertically align a body with respect to a prototype. The `\jparbox` vertically aligns paragraph box with respect to a prototype. The `\addbox` adjusts height and depth of box. The `\pbox` is a simple version of one-column table. It is independent on `\arraystretch` value. The `\cbox` is intended for design of fancy headers in tables.
`ncccomma`	Implements the smart comma in math mode working as an ordinary character if a decimal character goes after it. Otherwise, the math comma works as a punctuation mark.
`ncccropbox`	Implements the `\cropbox` command preparing a box with crop marks at its corners looking like angles. Angle parameters are customizable.
`ncccropmark`	Implements the `\cropmark` command producing crop box around page text area (header area, footer area and marginal notes are optionally taken into consideration). The `\cropmark` command is useful as a parameter of the `\watermark` commands form the `watermark` package. It accurately interprets current state of two-column, two-side, and reverse-margin modes.
`nccfancyhdr`	Absolutely new implementation of functionality of the `fancyhdr` package. It is more transparent, simple, and non-aggressive (redefining of standard page styles is optional). Using the package with names of standard page styles as options, you can easy decorate your document with header/footer rules. For example, the command `\usepackage[headings]{nccfancyhdr}` sets the `headings` page style and provides it with the decorative rule at the header. Header width control is improved with two commands, namely `\extendedheaders` (extended upon marginal notes) and `\normalheaders`. The `\thispagestyle` command correctly works with the fancy page style (in fancyhdr, it didn't work because of use of global definitions). A new page style

can be easy created with the help of the `\newpagestyle` command and fancy mark commands.

nccfloats	Wraps LaTeX floats with service commands `\fig`, `\tabl`, `\figs`, `\tabls`, introduces the `\minifig` and `\minitabl` commands preparing figure and table in a minipage with possible use of `\caption` within, and `\sidefig` and `\sidetabl` used for placement of minifloats next to surrounding text on the outer side of page.
nccfoots	The package implements commands for generating footnotes with manual marks. For example, to mark footnote by star you can write `\Footnote{$*$}{Footnote text}`.
nccmath	Extension of the `amsmath` package. Its main aim is to combine AMS's typesetting of display equations and NCC-LaTeX's one. In `amsmath`, the `eqnarray` environment leaves unchanged. This package redefines `eqnarray` to allow using of `amsmath` tag control features and display breaks. Inter-column distance in `eqnarray` is reduced to the distance typical for relation operations. All columns are prepared in the `\displaystyle`. A new `darray` environment is a mix of the AMS's `aligned` environment and LaTeX's `array` environment. It is typed out in the same way as the `aligned` environment but has columns definition parameter as in `array` environment. The use of column specifications is restricted to the necessary commands only: `l`, `c`, `r`, `@`, and `*` are allowed. The implementation has no conflicts with the packages redefining arrays. The `fleqn` and `ceqn` environments allow dynamically change the alignment of display formulas to flushed left or to centered alignment. Some additional commands are introduced also.
nccparskip	Useful for documents with non-zero skips between paragraphs. In this case, the additional vertical space inserted by lists is unlikely. The package provides identical distance between all paragraphs except sectioning markup commands. It redefines control list commands and suppress `\topskip`, `\partopskip`, and `\itemsep` in lists. As a result, the distance between ordinary paragraphs and paragraphs prepared by lists is the same. The `\SetParskip{distance}` command controls this distance.
nccpic	Envelop for the `graphicx` package. It customizes graphics extensions list for dvips driver. You need not specify a graphics file extension when use the `\includegraphics` command. Depending on a dvi-driver specified, a graphics file with an appropriate extension is searched. So, you only need to create a number of versions of a graphics file in different formats (for example, '.bmp' for dvips or Yap and '.png' for pdftex). After that you can produce resulting '.ps' and '.pdf' file without any changes in the source file. The recommended storage for graphics files is the 'graphics/' subdirectory of the directory the '.tex' file is translated. Some additional commands are introduced also.

`nccrules` Implements two commands, `\dashrule` and `\dashrulefill`, which compose dashed (multi)lines. Two footnote rule generation commands, `\newfootnoterule` and `\newfootnotedashrule`, are useful in conjunction with the `manyfoot` package.

`nccsect` Extension of LaTeX's section, caption, and toc-entries generation technique. The package contains many improvements in comparison with the base LaTeX's implementation. The most interesting of them are:

- simple declaring of sections of any level (including sections of 0th level and captions for floats);
- user-controlled typeout for display sections (user can select one of the following typeout styles: `hangindent`, `hangindent*`, `parindent`, `parindent*`, `hangparindent`, `hangparindent*`, `center`, and `centerlast`);
- new section styles can be easy constructed with the help of two style definition commands;
- customizing of section or caption tag by the manner similar to $\mathcal{AMS}$ equation tag;
- simple declaring of toc-entries using prototypes for calculation of hang indentations;
- `\numberline` command newer overlaps the text going after;
- `\PnumPrototype` command is used for calculation of right margin in table of contents;
- different captions for different float types;
- simple handling of new types of floats (after registration of a new float in the package, you can declare a caption and toc-entry for it; be sure that the `\chapter` command will automatically produce a vertical skip in a toc for the new float also).

`nccstretch` Implements the `\stretchwith` command that stretches a text inserting something between every pair of neighbour tokens.

`nccthm` Yet another extension to the `\newtheorem` command. The following orthogonal properties of theorems are used:

numbering mode is standard or *apar* (a number before header);

theorem type defines an appearance of a theorem (what fonts are used for title, comment, and body). The 'theorem' and 'remark' types a predefined;

indent style	margin, nomargin, indent, noindent (selected in package options);
break style	do break after header or not? Hardcoded when a new theorem is declared. Can be overridden on the fly for concrete math statement.

Easy customization of spacing and commands inserted at the end of theorem headers. Two types of Q.E.D. symbol (white and black). The `\proof` command is introduced. The `\newtheoremtype` command allows create new theorem types. The `\like`*type* command is automatically created for every new type. It simplifies typeout of rare math statements. Instead of creation of a new theorem environment you can use

`\liketheorem{Title}{Number}[Comment]`

for it, or `\likeremark`, etc. Every theorem environment and theorem type can be redefined with `\renewtheorem` and `\renewtheoremtype`. Counters of all theorem-like environments are dynamic (the `dcounter` package is used). So, they are created at the first use. This is very helpful for package writer. A number of theorem-like environments can be created in a package or class, and a user only selects the count style for them (in simple case this can be done with only one `\countstyle` command in the document preamble).

`textarea` Allows expand the text area of a document on other areas: footer, header, and margin. There are two ways for such expansion — temporary expansion of a current page on the header using a special negative vertical skip, or global change of page layout parameters with possibility of later restoration them to the original values. All these actions should be applied at the beginning of page.

`tocenter` Provides two commands, namely `\ToCenter` and `\FromMargins`, simplifying the customization of page layout. It is now easy to change text width and height and center the text area on the page (the header, footer, and marginal note fields are optionally taken into consideration). Other way is like-Word declaring of paper layout: you set margins from the left, right, top, and bottom edges of page.

`topsection` Implements an unnumbered top-level section (`\chapter` in books and reports or `\section` in articles) in class-independent way. Such a section is used as a title for standard document sections such as table of contents, index, bibliography, etc. Using this command allows simplify the definition or redefinition of standard document environments.

`watermark` Provides watermarks on output pages. A watermark is some text or picture printed at the background of paper. A watermark is usually stored in the page header. This way is inconvenient: it needs redefinition of page style commands. We use another method that is orthogonal to page mark commands and needs no redefinition of page marks. The left and right watermarks are allowed. Temporary `\thiswatermark` acts on the current page only. Using this way it is easy to replace a page header on a page by your own page header with the `\thispageheading` command.