The metastr Package

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland

nbeisert@itp.phys.ethz.ch

2020/09/02, v1.1.2

Abstract

metastr is a LaTeX 2_ε package to store and compose strings in a structured way. This can serve several purposes such as: manage and write document metadata; use templates for formatting document data; assist in assembling and displaying document license information; facilitate basic internationalisation and localisation.

Contents

1 Introduction

This package provides some basic functionality to store and compose strings. The main goal is to keep relevant information for the document in a structured way such that it can be accessed and used by conveniently using some standardised methods.

The package has the following goals, tasks and features:

- manage document metadata and write them to the PDF output file;
- set up and use templates for formatting document data, e.g. for title pages;
- assist in assembling and displaying document license information;
- facilitate basic internationalisation and localisation;
- provide preset texts and common license statements in different languages.

Using the structures provided by the packages makes particular sense if you can rely on predefined text and formatting or if you have a couple of similar documents for which you can define suitable templates.

2 Usage

To use the package metastr, add the command

```
\usepackage{metastr}
```

to the preamble of the LaTeX document.

2.1 Defining Strings

`\metadef` `\metaset`

The package supplies registers for storing data. Registers need to be declared before they can be filled or used (unless the package option `checkdef=false` is set, see section 2.7). A new register *reg* is declared by the command:

```
\metadef{reg}
```

The register *reg* can be filled with the value *def* by the command:

```
\metaset{reg}{def}
```

The package declares a couple of registers for storing standard metadata. The basic set of registers consists of:

`title`	document title
`subtitle`	document subtitle
`author`	document author
`location`	location associated to the document
`date`	document date
`subject`	'subject' of the document
`keywords`	(a list of) keywords describing the document
`titlematter`	composition register for title matter
`titletext`	composition register for title
`authortext`	composition register for author
`datatext`	composition register for location and date

The register `titletext` composes information for printing the title (such as `title`, `subtitle` but also `author`, `location` and `date`). It can serve a similar purpose as the LaTeX command `\maketitle`.

Finally, there are some auxiliary registers:

`language`	main language of the document
`url`	URL of the document or additional info on it
`urlmessage`	message to describe the document URL
`source`	name of the source file
`draft`	indicator of draft version
`linebreak`	expands to linebreak when printed or space otherwise

The register `language` specifies the main language used in the document. This should be a two-letter language code (*ln*) potentially followed by two-letter country code (*ln*-*CN*) such as `en` or `en-GB`. The language has some impact on selecting register variants, see section 2.3 and section 2.6, it should therefore be set by the command:

`\metasetlang`[`*`]{*ln-CN*}

The starred version declares the language for PDF metadata rather than the document contents.

2.2 Working with Strings

`\metaget` A register *reg* can be read out by the macro:

`\metaget[]{`*reg*`}{`*def*`}`

Note that the (empty) argument in square brackets is mandatory, it cannot be left out, see section 2.3 for further details on its purpose. This is because `\metaget` must be robust so that its output can be processed for writing (optional arguments make a macro fragile). In case the register *reg* has not been filled, `\metaget` returns nothing.

`\metaif` Sometimes one may want to test whether a register is filled or not, e.g. in order to display a default value otherwise. This can be achieved by the conditional:

`\metaif[]{`*reg*`}{`*true*`}{`*false*`}`

If the register *reg* is filled, return *true* otherwise *false*. Again, the argument in square brackets is mandatory.

`\metaunset` The following command cleans a register *reg* which has previously been filled:

\metaunset{*reg*}

Note that cleaning is different from filling an empty string when it comes to the conditional `\metaif` which evaluates true for an empty string but false for a clean register.

`\metaappend` `\metaprepend` `\metaaddsep`

The content of registers can be manipulated by some commands. To append or prepend a string to a register, use the commands:

\metaappend{*reg*}{*def*}
\metaprepend{*reg*}{*def*}
\metaaddsep{*reg*}{*sep*}{*def*}

The latter command `\metaaddsep` is designed to compose lists with separators, it appends the separator *sep* and the value *def* unless the register is clean, in which case it is set to *def* without the separator *sep*.

2.3 String Variants

A versatile feature of the registers is that they can be provided in several variants. These variants can be used for producing different representations of the same register depending on the intended situation. For example, a title could be given in a fully formatted version `print` for printing, a bare version for metadata and a shortened version for headings. Similarly, translations to different languages could be stored as different versions of the same register, see section 2.6. Moreover, certain attributes related to the registers could be stored in additional variants.

`titletext`

For example, the register `titletext` exists in the default variant (mainly intended for writing out metadata) and the `print` variant (for printing out a combination of title data on the title page). The default variant expands to a combination of `draft`, `title` and `subtitle` (as far as filled):

[*draft*:]*title*[– *subtitle*]

The `print` variant is accessed by the command:

\metapick[print]{titletext}

It expands to two lines containing `title` and `subtitle` + `draft` (as far as filled):

title
subtitle \par *draft*

In fact, the legacy behaviour of `titletext` also adds two lines containing `author` and `location` + `date`, as in the new composition register `titlematter` to be explained below; this behaviour needs to be disabled explicitly for purposes of backwards compatibility by the package option `titlematter`. The formatting style of each line is given by the variant `style` of the first register on this line; the vertical space above each line is produced by the variant `skip`. Two items on a single line are separated by the variant `sep` of the second register; an unfilled `sep` variant puts the two items on individual lines (by default this applies to `subtitle` and `draft`).

`titlematter`

The composition register `titlematter` (in `print` variant):

\metapick[print]{titlematter}

collects extended information on the documents for display on the title page (it substitutes the LaTeX command `\maketitle` in the class article whose display it mimics):

title
subtitle \par *draft*
author
location, *date*

By adjusting the definitions, the layout of the title display on the title page can be adjusted conveniently.

Variants are always specified by an argument `[`*var*`]` in square brackets preceding the register `{`*reg*`}`. This argument is optional for commands which set registers and which can be fragile; it is however *mandatory* for macros which read the register content and whose output needs to be expandable into the output stream (even though the register is not optional, it is more uniform to stick with square brackets to specify the variant). The main declaration involving variants is:

`\metaset[`*var*`]{`*reg*`}{`*def*`}`

This command defines the register *reg* in variant *var* as *def*. The default variant is the empty string, while the variant `print` is intended for printed output. The variant string is obtained by:

`\metaget[`*var*`]{`*reg*`}`

`\metapick` The variant mechanism can become powerful through macros which fall back to default variants if the desired variant has not been filled explicitly:

`\metapick[`*var*`]{`*reg*`}`

This macro tests whether the variant *var*, the language variant *ln* specified through `\metasetlang{`*ln*`}`, the generic (empty) variant or a fallback language variant have been specified. If so, their value is returned (in this order of preference). Importantly, the intended variant *var* is passed along to the evaluation of *reg* as the argument '`#1`' in the definition string *def* of `\metaset`. This allows to define a register in one generic variant which composes other registers in more specific variants. To that end reference registers should be accessed by the construct:

`\metaset{`*reg1*`}{...\metapick[#1]{`*reg2*`}...}`

When this register is accessed by `\metapick[`*var*`]{`*reg1*`}`, it will read the default variant of *reg1* which will pass on to *reg2* in *var* (rather than in the default variant). A useful application within title declarations is `\metapick[#1]{linebreak}` which expands to a line break when the title is displayed and to a space when the title is used elsewhere.

`\metaifpick` There also exist a corresponding conditional:
`\metacompose`

`\metaifpick[`*var*`]{`*reg*`}{`*true*`}{`*false*`}`

This command tests whether any of the above variants *var* have been filled. Another convenient macro to more efficiently compose strings is:

`\metacompose[`*var*`]{`*reg*`}{`*prefix*`}{`*postfix*`}{`*empty*`}`

It returns the intended register value with prefix string *prefix* and suffix string *suffix* if any of the above variants have been filled; otherwise it returns *empty*. For example, the prefix and/or suffix could be separators for displaying the content of an optional register.

In dealing with variants, the following commands specify the variant *var* as an optional argument `[`*var*`]`:

```
\metaset[var]{reg}{def}
\metaunset[var]{reg}
\metaappend[var]{reg}{def}
\metaprepend[var]{reg}{def}
\metaaddsep[var]{reg}{sep}{def}
```

For the following macros, specifying the variant *var* as `[var]` is *mandatory*:

```
\metaget[var]{reg}
\metapick[var]{reg}
\metaifpick[var]{reg}{true}{false}
\metacompose[var]{reg}{prefix}{postfix}
```

Here, the default variant is accessed by an empty argument *var*.

`titletext` `\metatitleline` `\metatitlelinetwo`

To illustrate a construction using variants, let us consider the above register `titletext` (with activated package option `titlematter`). It is defined in the generic variant as:

```
\metaset{titletext}{%
  \metacompose[#1]{draft}{}{\metaget[sep]{draft}}{}%
  \metapick[#1]{title}%
  \metacompose[#1]{subtitle}{\metaget[sep]{subtitle}}{}{}}
```

This expands to the prefix '*draft*: ' (if available), the main title '*title*' and the suffix ' – *subtitle*' (if available). The `print` variant to output a full title for the document is defined by:

```
\metaset[print]{titletext}{%
  \metatitleline[print]{title}%
  \metatitlelinetwo[print]{subtitle}[print]{draft}
```

Here, the macros `\metatitleline`[`two`] produce a title line consisting of one or two items. The single-item version is defined as:

```
\def\metatitleline[#1]#2{%
  \metacompose[#1]{#2}
   {\metaget[skip]{#2}\begingroup\metaget[style]{#2}}
   {\par\endgroup}{}}
```

If register `#2` is filled, this expands to the vertical skip defined by the variant `skip` and an encapsulated paragraph of the register value in the layout defined by the variant `style`.

`authortext` `datetext` `titlematter`

The composition registers `authortext` and `datetext` work in a similar fashion as `titletext` and compose the registers `author` and `location` + `date`, respectively. The composition register `titlematter` (in variant `print`) prints out the title, author and date information and serves a similar purpose as the LaTeX command `\maketitle`:

```
\metaset{titlematter}{%
  \metapick[#1]{titletext}%
  \metapick[#1]{authortext}%
  \metapick[#1]{datatext}%
  \metaget[skip]{titlematter}}
```

2.4 Write Document Metadata

The contents of certain registers can be written out to PDF files as metadata using the package hyperref and the extension hyperxmp.

\metawritepdfinfo The basic metadata registers are written out by \metawritepdfinfo using hyperref. The mapping between metastr registers and hyperref \hypersetup options is given by:

titletext	→	pdftitle
authortext	→	pdfauthor
subject	→	pdfsubject
keywords	→	pdfkeywords

Here, titletext is used instead of title to compose information from the registers draft, title and subtitle (as far as filled). Similarly, authortext composes information on the author, typically just author. Note that \metawritepdfinfo will be effective only when invoked before the contents of the first page are written out.

\metawritepdfaux Auxiliary metadata is written out using hyperxmp by the command \metawritepdfaux with the mapping:

url	→	pdfurl
source	→	pdfsource

etawritepdfpreamble Some metadata must be written out sufficiently early, i.e. in the document preamble, in order to go into effect. These include the language settings, and they are written out by \metawritepdfpreamble with the mapping:

language	→	pdflang
language variant [meta]	→	pdfmetalang
	→	keeppdfinfo

Note that pdfmetalang is a setting of hyperxmp and will be ignored if the package is not loaded. Furthermore, the hyperxmp option keeppdfinfo will be set unless the package option xmppdfinfo=false is set.

metawritepdfcontact A contact can be specified within PDF files in a standardised format using hyperxmp. The command \metawritepdfcontact passes on the following registers with the mapping:

contactaddress	→	pdfcontactaddress
contactpostcode	→	pdfcontactpostcode
contactcity	→	pdfcontactcity
contactregion	→	pdfcontactregion
contactcountry	→	pdfcontactcountry
contactemail	→	pdfcontactemail
contacturl	→	pdfcontacturl

\metawritepdfrights A document copyright statement, see section 2.5, is recorded within the PDF file by \metawritepdfrights using hyperxmp with the mapping:

rightstext	→	pdfcopyright
licenseurl	→	pdflicenseurl

\metawritepdf writepdf Finally, it makes sense to write out PDF metadata automatically. This is controlled by filling or clearing certain variants *var* of the register writepdf:

```
    \metaset[var]{writepdf}{}
or  \metaunset[var]{writepdf}
```

If the variant auto is filled (default), PDF metadata is written automatically at the beginning of the document block by calling \metawritepdf. The command \metawritepdf calls the

commands `\metawritepdf...` depending on whether the variants `info`, `aux`, `preamble`, `contact`, `rights` of the register `writepdf` are filled; the variants `info`, `aux`, `preamble` are enabled by default, the variants `contact`, `rights` need to be enabled explicitly.

Note that the basic metadata such as `author` and `title` do not have to be defined already in the preamble, but (depending on the combination of drivers and packages) they can be set before the contents of the first page are shipped out to the PDF file. If the basic registers are to be declared on the first page, one should disable their automatic writing by `\metaunset[info]{writepdf}`. When the corresponding registers have been filled, (but no later than the end of the first page), they need to be written manually by invoking `\metawritepdfinfo`.

2.5 Copyright and Licenses

Specifying a copyright statement and a license is very useful because it makes the allowed (re)use of the provided material evident to the reader. However, it also takes some efforts to set things up properly. The package metastr provides some default texts to state the license for a couple of well-established licenses. For instance, the set of Creative Commons licenses has become a standard to mark the intended (re)use of a document involving creative content. For documents related to software, there is a number of standard software licenses to choose from.

Registers. The package declares the following registers to state the copyright:

`copyrightmark`	'©' (`print` variant) or "Copyright"
`copyrightdate`	copyright date
`copyrightowner`	copyright owner
`copyrightstatement`	combines: `...mark` + `...date` + `...owner`
`copyrightmessage`	a message explaining the copyright situation

The package declares the following registers to state the license:

`licensemessage`	licensing message
`licenseprovider`	license provider
`licenseversion`	license version
`licenselogo`	license logo inclusion
`licenselogomessage`	display the license logo
`licenseurl`	URL with license text and details
`licenseurlmessage`	display the license URL

Furthermore, there are some related auxiliary registers:

`partof`	specifies the work this document is a part of
`partofmessage`	a message declaring being part of
`attributionmessage`	a message declaring attributions
`rightstext`	a composition template for all of the above

The above information is compiled automatically in the register `rightstext`. It can be written as PDF metadata as well as printed with formatting:

```
\metapick[print]{rightstext}
```

Presets. The package provides a couple of presets for commonly used copyright statements and licenses. These are selected by:

```
\metacopyright{preset}
\metalicense{preset}
```

The following *preset* values provide the associated copyright statements:

`plain`	This work is protected by copyright.
`parts`	This work as well as its parts is protected by copyright.
`doc`	This document is protected by copyright.
`doc-parts`	This document as well as its parts is protected by copyright.
`reserved`	All rights reserved.
`publicdomain`	This work is dedicated to the public domain.

The following *preset* values provide the associated license statements:

`consent`	Reproduction of any part of this work in any form without prior written consent *of the author* is not permissible.
`consent-noncom`	Reproduction of any part of this work in any form without prior written consent *of the author* is permissible only for private, scientific and non-commercial use.
`lppl`	This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version *1.3* of this license or (at your option) any later version. http://www.latex-project.org/lppl.txt

The license URL will be selected where available. The italicised parts of the license statement can be customised by the registers `licenseversion` and `licenseprovider`.

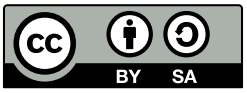
Creative Commons Licenses. A Creative Commons license can be selected by the command:

```
\metalicensecc{license}
```

The parameter *license* specifies the type of CC license:

`by`	Attribution
`by-sa`	Attribution-ShareAlike
`by-nd`	Attribution-NoDerivatives
`by-nc`	Attribution-NonCommercial
`by-nc-sa`	Attribution-NonCommercial-ShareAlike
`by-nc-nd`	Attribution-NonCommercial-NoDerivatives
`zero`	CC0 public domain declaration
`pd`	generic public domain declaration

The appropriate license URL and CC logo is selected by the command as well, e.g.



https://creativecommons.org/licenses/by-sa/4.0/

Note that `pd` is not a CC license, but it declares that the document is in the public domain by `\metacopyright{publicdomain}` and it selects the corresponding CC logo for public domain content.

A version of the CC license can be specified by the register `licenseversion`. The default version is `4.0` (international), further available versions are `3.0` (unported) as well as `2.5`, `2.0`, `1.0` (generic). For the CC0 license `zero`, the only available version is `1.0` (universal) which is the default.

Displaying the logo requires (manual) loading of the package graphicx; furthermore the package doclicense containing the logo files must be present. The display of the logo can be disabled by the package option `cclogo=false`. The logo display is coded by the following definitions which can be customised:

```
\metaset[print]{licenselogomessage}{%
  \centerline{\metapick[#1]{licenselogo}}}
\metaset[cmd]{licenselogo}{\includegraphics{#1}}
```

Various registers and variants of the selected CC license exist. The registers specific to CC licenses are:

`licencecc`	CC license identifier
`licenceccver`	CC license version
`licenceccfull`	full license descriptor

The variants specific to CC licenses are:

ln	representation in language *ln*
`icon`	CC icon (package ccicons required), e.g.
`url`	license URL
`ident`	CC identifier, e.g. 'BY-SA'
`short`	short form, e.g. 'CC BY-SA'
`logo`	logo filename (package doclicense)

For example, a full license descriptor is displayed by `\metapick[]{licenseccfull}`: Creative Commons License "Attribution-ShareAlike 4.0 International". The license icon can be displayed by `\metaget[icon]{licensecc}`: (this requires the package ccicons to be loaded). Note that displaying the full license message `licensemessage` in variant `print` in some languages may produce quotation marks not declared in default fonts causing an error; this can be avoided to some extent by loading an appropriate packages for internationalisation such as babel.

2.6 Languages

A principal application of the register variants is to implement internationalisation and localisation. Evidently, this is a tricky subject due to various particularities of languages, but the register variants can be used to specify and select different language representations for some commonly used text elements. For example, the copyright and license statements in section 2.5 are internationalised (to some extent). This makes them conveniently usable in the appropriate language. Note that the language presets to be loaded need to be specified explicitly by the package option `loadlang`, see section 2.7.

The idea is to understand the variant *var* of a register *reg* to be its representation in the language *var*=*ln*[-*CN*]. The default (empty) variant as well as specific purpose variants (such as `print`) should be provided in the document language or a fallback language (such as English). The macro `\metapick[`*var*`]{`*reg*`}` then selects the appropriate language

representation or falls back to the default language. Here *var* can specify a particular language or a particular purpose. Then, `\metapick` will pick (in this order of preference):

- the language or purpose *var*,
- the document language specified by `\metasetlang` (if available),
- the document language specified by `\metasetlang` with country code stripped (if available),
- the default variant,
- the fallback language (first of package option `loadlang`).

Note that nesting of `\metapick` via `\metapick[#1]{`*reg*`}` passes along the original variant *var* in the parameter `#1`. This mechanism allows to specify some non-specific elements in a universal language while the appropriate language is selected where available.

`\metaterm` `\metatranslate` `\metasettest`

The package reserves registers of the form `term-`*term* for storing terms in various (language) representations. A couple of such term registers describing common entities in typesetting are defined by the package:

`title`	Title	`appendix`	Appendix
`abstract`	Abstract	`page`	Page
`copyright`	Copyright	`figure`	Figure
`preface`	Preface	`table`	Table
`part`	Part	`contents`	Contents
`chapter`	Chapter	`listfigure`	List of Figures
`section`	Section	`listtable`	List of Tables
`subsection`	Subsection	`references`	References
`paragraph`	Paragraph	`index`	Index
		`draft`	DRAFT

These are provided in different languages for convenient internationalisation (this can be viewed as a low-key implementation of some of the features of the babel package). Additional term registers can be defined by the user. Term registers are accessed by the macros:

```
\metaterm{term}
\metatranslate[ln]{term}
\metasetterm[ln]{reg}{def}
```

The macro `\metaterm` obtains the term *term* in the default language (it invokes `\metapick[]` with empty variant), while `\metatranslate` uses any other language *ln*. The macro `\metasetterm` declares and fills a term register *term* (in a particular language *ln*). Note that therefore it is not necessary to declare term registers explicitly by `\metadef`.

The PDF metadata are written out in the metadata language variant specified by `\metasetlang*`; otherwise in the default document language specified by `\metasetlang` is used. Some registers can even be written out in several alternative language versions using the package hyperxmp, namely `title`, `subject` and `rightstext`. The set of alternative languages is specified by (before invoking the respective command `\metawritepdf...`):

```
\metaset[altlang]{title}{languages}
\metaset[altlang]{subject}{languages}
\metaset[altlang]{rightstext}{languages}
```

Here, *languages* is a comma-separated list of language identifiers and for each identifier *ln* the information is written out in the respective language variant.

2.7 Package Options

General options for the package can be selected by the commands:

```
\usepackage[opts]{metastr}
or  \PassOptionsToPackage{opts}{metastr}
```

`\PassOptionsToPackage` must be used before `\usepackage`. *opts* is a comma-separated list of options.

The following options are available:

- `hyperref`[`=true`|`false`] (no value implies `true`, initially set to `true`) – use the package hyperref to write metadata to PDF.
- `hyperxmp`[`=true`|`false`] (no value implies `true`, initially set to `true`) – use the auxiliary package hyperxmp to write additional metadata to PDF.
- `checkdef`[`=true`|`false`] (no value implies `true`, initially set to `true`) – check whether registers have been previously declared when filling them.
- `cclogo`[`=true`|`false`] (no value implies `true`, initially set to `true`) – display CC logo from doclicense package.
- `cclogocurr=dollar`|`euro`|`yen` (initially set to `dollar`) – select currency symbol for Creative Commons NonCommercial logos.
- `cclogoshape=box`|`slim` (initially set to `box`) – select shape of Creative Commons NonCommercial logos.
- `xmppdfinfo`[`=true`|`false`] (no value implies `true`, initially set to `true`) – write the basic PDF info block when using the auxiliary package hyperxmp; if this option is set, hyperxmp is loaded with the option `keeppdfinfo`.
- `draft`[`=true`|`false`] (no value implies `true`, initially set to `false`) – fill `draft` register with "DRAFT".
- `titlematter`[`=true`|`false`] (no value implies `true`, initially set to `false`) – when set to `true`, composition register `titletext` prints only the title; when set to `false`, `titletext` also prints authorship, location and date information (legacy behaviour).
- `course`[`=true`|`false`] (no value implies `true`, initially set to `false`) – Setup extended registers for course materials, see section 2.8.
- `loadlang=`*ln-1*|*ln-2*|...|*ln-n* (bar-separated list w/o spaces, initially set to `en`) – Load presets for languages *ln-1*, *ln-2*, ..., *ln-n*, see section 2.6. The first language *ln-1* serves as the fallback variant. Available internationalisations currently consist of:

`en`	English
`de`	German
`fr`	French
`es`	Spanish

Selected package options can be adjusted after the package has been loaded by the command:

```
\metasetup{opts}
```

Presently, only `draft` can be adjusted.

2.8 Extras

The package can provide some special purpose registers on request.

Course Materials. A couple of registers for course materials are provided upon setting the package option `course`:

`course`	title of course
`material`	description of document material
`period`	period where course takes place
`institution`	institution where course is given
`instructor`	instructor of course

The variant `course` of the register `titlematter` (or `titletext`) displays a compilation of these registers for display on a title page:

course
material `\par` *draft*
institution, *period*
instructor

Furthermore, the registers `title`, `subtitle`, `author`, `location` and `date` are diverted to `course`, `material`, `instructor`, `institution` and `period` respectively. Consequently, their values are automatically written as PDF metadata, but it is certainly possible to override them with custom values.

3 Information

3.1 Copyright

This work consists of the files `README.txt`, `metastr.ins` and `metastr.dtx` as well as the derived files `metastr.sty`, `metasamp.tex` and `metastr.pdf`.

3.2 Files and Installation

The package consists of the files:

`README.txt`	readme file
`metastr.ins`	installation file
`metastr.dtx`	source file
`metastr.sty`	package file
`metasamp.tex`	sample file
`metastr.pdf`	manual

The distribution consists of the files `README.txt`, `metastr.ins` and `metastr.dtx`.

- Run (pdf)LaTeX on `metastr.dtx` to compile the manual `metastr.pdf` (this file).

- Run LaTeX on `metastr.ins` to create the package `metastr.sty` and the samples consisting of `metasamp.tex`. Copy the file `metastr.sty` to an appropriate directory of your LaTeX distribution, e.g. *texmf-root*`/tex/latex/metastr`.

3.3 Related Packages

The package makes use of other packages available at CTAN:

- This package uses the package hyperref to write basic metadata to a PDF file. Compatibility with the hyperref package has been tested with v7.00c (2019/11/10).
- This package uses the package keyval from the graphics bundle to process the options for the package, environments and macros. Compatibility with the keyval package has been tested with v1.15 (2014/10/28).
- This package can use the package hyperxmp to write extended metadata to a PDF file. Compatibility with the hyperxmp package has been tested with v5.4 (2020/06/19).
- This package can use the Creative Commons license icon files included in the package doclicense. Compatibility with the doclicense package has been tested with v2.0.1 (2020/06/26).
- This package can use the Creative Commons license icon fonts included in the package ccicons. Compatibility with the ccicons package has been tested with v1.6 (2017/10/30).
- Icon files are displayed by means of the graphicx package. The package needs to be loaded explicitly. Compatibility with the graphicx package has been tested with v1.1a (2017/06/01).

There are several other LaTeX packages which store and write basic metadata for some specific purposes:

- The package hyperref writes the arguments of `\author` and `\title` unless the package option `pdfusetitle=false` is declared (at load time).
- The package hyperxmp writes the arguments of `\author` and `\title`.
- The package exframe writes the `\exercisedata` registers `author`, `title`, `subject` and `keyword` unless the package option `pdfdata=off` is specified.
- The package beamer writes the arguments of `\author`, `\title`, `\subject` and `\keywords`.
- The package gitver writes `pdfsubject` unless the package option `nopdfinfo` is specified.
- Various packages to prepare articles for publication in journals.

Their mechanisms may be in competition with the ones of the present package metastr. In order to make the packages work together on the same set of data, the most promising option which should work in many cases is the following: Fill the registers of metastr with the desired values. Then pass them on to the structures of the other package(s) using `\metaget` or `\metapick`. Since the latter commands are robust, the other structures ought to be able to handle them without further ado. To avoid potential conflicts, multiple writing of (basic) metadata should be disabled. For the metastr package this is achived by:

```
\metaunset[info]{writepdf}
or  \metaunset[auto]{writepdf}
```

3.4 Feature Suggestions

The following is a list of features which may be useful for future versions of this package:

- Presets for GNU and other software licenses.
- Registers for publication data.
- Further translations of copyright and license statements.
- Export translations to files
- Make use of the babel package for translations of basic terms.

3.5 Revision History

v1.1.2: 2020/09/02

- improve hook processing to set `pdflang` at the right moment

v1.1.1: 2020/08/27

- introduce `skip` variant for `titlematter` register
- adjust hook to new LaTeX 2$_\varepsilon$ (2020-10) hook processing

v1.1: 2020/06/28

- `\metasetup` to adjust some package options (`draft`)
- `titlematter`, `authortext`, `datatext` composition registers added; legacy behaviour of `titletext` preserved, can be changed by package option `titlematter`
- `linebreak` register added
- fix saving of PDF info (`keeppdfinfo`) with updated hyperxmp package
- fix package options `hyperref`, `hyperxmp`, `checkdef`, `xmppdfinfo`
- fix compatibility with updated doclicense package (v2)

v1.0: 2020/02/06

- first version, published on CTAN

A Sample

This section provides an example of how to apply some of the metastr mechanisms and licenses.

Some lines in the example are commented by `%%` for easy experimenting.

Preamble. Standard document class:

```
\documentclass[12pt]{article}
```

Use package geometry to set the page layout; adjust the paragraph shape:

```
\usepackage{geometry}
\geometry{layout=a4paper}
\geometry{paper=a4paper}
\geometry{margin=2.5cm}
\parindent0pt
\parskip1ex
```

Declare some options for the package hyperref; it does not hurt to load it explicitly although metastr will invoke it by default if not loaded:

```
\PassOptionsToPackage{bookmarks=true}{hyperref}
\usepackage{hyperref}
```

Set some options for the metastr package:

```
\PassOptionsToPackage{loadlang=en|de|fr|es}{metastr}
%%\PassOptionsToPackage{loadlang=en|fr|es}{metastr}
%%\PassOptionsToPackage{loadlang=de|en}{metastr}
\PassOptionsToPackage{cclogocurr=euro}{metastr}
%%\PassOptionsToPackage{cclogoshape=slim}{metastr}
```

Include the metastr package along with graphicx, babel and ccicons (where available):

```
\usepackage[titlematter]{metastr}
\usepackage{graphicx}
\usepackage[english]{babel}
\IfFileExists{ccicons.sty}{\usepackage{ccicons}}{}
```

Some Adjustments. Declare some term to be translated; doesn't hurt to declare a couple of variants:

```
\metasetterm[en]{Zurich}{Zurich}
\metasetterm[de]{Zurich}{Z\"rich}
\metasetterm[fr]{Zurich}{Zurich}
\metasetterm[es]{Zurich}{Z\'urich}
\metasetterm[it]{Zurich}{Zurigo}
\metasetterm[pt]{Zurich}{Zurique}
```

Define `subject` to combine `location` and `date` (as far as filled):

```
\metaset{subject}{\metacompose[#1]{location}{location: }
   {\metacompose[#1]{date}{, date: }{}{}}
   {\metacompose[#1]{date}{date: }{}{}}}
```

Adjust title display:

```
\metaset[skip]{subtitle}{\vspace{1ex}}
\metaset[skip]{author}{\vspace{2ex}}
\metaset[skip]{location}{\vspace{1ex}}
\metaset[skip]{date}{\vspace{1ex}}
\metaset[style]{title}{\LARGE\bfseries}
\metaset[style]{author}{\large\scshape}
\metaset[sep]{draft}{ -- }
%%\metaunset[sep]{date}
```

Write title also in english and german; write rights as PDF metadata also in English and Spanish:

```
\metaset[altlang]{title}{en,de}
\metaset[altlang]{rightstext}{en,es}
\metaset[rights]{writepdf}{}
```

Set Document Data. Set the document language:

```
\metasetlang{en}
%%\metasetlang{de}
%%\metasetlang{de-CH}
%%\metasetlang{fr}
```

Define some document data:

```
\metaset[en]{title}{A metastr Sample}
\metaset[de]{title}{Ein metastr Beispiel}
\metaset[print]{title}{A \textsf{metastr} Sample}
\metaset{subtitle}{Illustration of some features}
\metaset{author}{Niklas Beisert}
\metaset{keywords}{composition of title, application of licenses, translations}
\metaset{location}{\metatranslate[#1]{Zurich}}
\metaset{date}{2020/02/06}
\metaset{partof}{The metastr Package}
\metaset[print]{partof}{The \textsf{metastr} Package}
\metasetup{draft=true}
```

Copyright settings:

```
\metaset{copyrightowner}{\metapick[#1]{author}}
\metaset{copyrightdate}{2020}
\metacopyright{doc}
%%\metacopyright{reserved}
```

License settings:

```
%%\metaset{licenseversion}{1.2}
%%\metalicense{lppl}
```

Creative Commons License use:

```
%%\metaset{licenseversion}{3.0}
\metalicensecc{by-sa}
%%\metalicensecc{by-nc-sa}
%%\metalicensecc{zero}
%%\metalicensecc{pd}
```

Scale the CC logo a bit:

```
\metaset[cmd]{licenselogo}{\includegraphics[scale=0.75]{#1}}
```

Start document body:

```
\begin{document}
```

Header. Display title block:

```
\pdfbookmark[1]{\metaterm{title}}{title}
\begin{center}
\metapick[print]{titlematter}
\end{center}
```

Display fineprint in fine print:

```
\vspace{1ex}\hrule\par\vspace{1ex}
\begingroup\footnotesize
\pdfbookmark[1]{\metaterm{copyright}}{copyright}
\metapick[print]{rightstext}
\endgroup
\vspace{1ex}\hrule\par\vspace{1ex}
```

Content. Some useful content:

```
\section{Metadata Inspection}

The metadata stored in this example PDF can be inspected with
the tool \texttt{pdfinfo}:

\begin{tabular}{l}
\verb+pdfinfo metasamp.pdf+\\
\verb+pdfinfo -meta metasamp.pdf | less+
\end{tabular}
```

Translations. Demonstration of terms and translations:

```
\section{Translations}

\begin{tabular}{ll}
document language:&\metaterm{Zurich}\\
Spanish:&\metatranslate[es]{Zurich}
\end{tabular}
```

Creative Commons. Demonstrate some CC terms:

```
\metaif[]{cc@type}{ % only if a CC license is in use
\section{Creative Commons}
some representations of the selected license:
\begin{itemize}
\item license identifier:
  \metapick[]{licensecc}
\item \texttt{short} identifier:
  \metapick[short]{licensecc}
\item full form:
  \metapick[]{licenseccfull}
\item \texttt{ident} form:
  \metapick[ident]{licensecc}
\item \texttt{short} form:
  \metapick[short]{licenseccfull}
\IfFileExists{ccicons.sty}
  {\item \texttt{icon} forms:
   -- \metapick[icon]{licensecc}
   -- \metapick[icon]{licenseccfull} --}{}
\item \texttt{url} form:
  \metapick[url]{licenseccfull}
\end{itemize}
}{}
```

End of document body:

```
\end{document}
```

B Implementation

This section describes the implementation of the package `metastr.sty`.

B.1 Package Setup

The package declares a couple of setup options.

It loads the package keyval for extended options processing.

```
\RequirePackage{keyval}
```

hyperref
hyperxmp
checkdef
cclogo
cclogocurr
cclogoshape
xmppdfinfo

Store the selected package options in some corresponding internal macros:

```
\newif\ifmstr@opt@hyperref\mstr@opt@hyperreftrue
\newif\ifmstr@opt@hyperxmp\mstr@opt@hyperxmptrue
\newif\ifmstr@opt@checkdef\mstr@opt@checkdeftrue
\newif\ifmstr@opt@xmppdfinfo\mstr@opt@xmppdfinfotrue
\newif\ifmstr@opt@course\mstr@opt@coursefalse
\newif\ifmstr@opt@draft\mstr@opt@draftfalse
\newif\ifmstr@opt@titlematter\mstr@opt@titlematterfalse
\newif\ifmstr@opt@cclogo\mstr@opt@cclogotrue
\def\mstr@opt@cclogocurr{dollar}
\def\mstr@opt@cclogoshape{box}
\def\mstr@opt@loadlang{en}
\def\mstr@group{mstr@}
\define@key{\mstr@group}{hyperref}[true]{%
  \csname mstr@opt@hyperref#1\endcsname}
\define@key{\mstr@group}{hyperxmp}[true]{%
  \csname mstr@opt@hyperxmp#1\endcsname}
\define@key{\mstr@group}{checkdef}[true]{%
  \csname mstr@opt@checkdef#1\endcsname}
\define@key{\mstr@group}{xmppdfinfo}[true]{%
  \csname mstr@opt@xmppdfinfo#1\endcsname}
\define@key{\mstr@group}{cclogo}[true]{\csname mstr@opt@cclogo#1\endcsname}
\define@key{\mstr@group}{cclogocurr}{\def\mstr@opt@cclogocurr{#1}}
\define@key{\mstr@group}{cclogoshape}{\def\mstr@opt@cclogoshape{#1}}
\define@key{\mstr@group}{loadlang}{\def\mstr@opt@loadlang{#1}}
\define@key{\mstr@group}{course}[true]{\csname mstr@opt@course#1\endcsname}
\define@key{\mstr@group}{draft}[true]{\csname mstr@opt@draft#1\endcsname}
\define@key{\mstr@group}{titlematter}[true]{%
  \csname mstr@opt@titlematter#1\endcsname}
```

Pass undeclared options on to keyval processing:

```
\DeclareOption*{\expandafter\setkeys\expandafter\mstr@group%
  \expandafter{\CurrentOption}}
```

Process global options while loading package:

```
\ProcessOptions
```

\metasetup

`\metasetup` processes package options when the package has already been loaded:

```
\def\mstr@setup{mstr@setup}
\newcommand{\metasetup}[1]{\setkeys\mstr@setup{#1}}
```

B.2 Definitions

The following describes the basic definitions of the package.

Required Packages. The package loads the packages hyperref and hyperxmp (unless excluded):

```
\ifmstr@opt@hyperref\RequirePackage{hyperref}\fi
\ifmstr@opt@hyperxmp\RequirePackage{hyperxmp}\fi
\ifmstr@opt@xmppdfinfo\ifdefined\xmptilde\hypersetup{keeppdfinfo}\fi\fi
```

General Definitions.

`\mstr@exptwo` A macro to conveniently expand the third token in line:

```
\def\mstr@exptwo#1{\expandafter#1\expandafter}
```

`\mstr@csdo` `\mstr@csdotwo` Some macros to conveniently expand `\csname` arguments before expanding the macro:

```
\def\mstr@csdo#1#2{\expandafter#1\csname#2\endcsname}
\def\mstr@csdotwo#1#2#3{\mstr@exptwo#1#2\csname#3\endcsname}
```

`\mstr@iftext` Check whether macro `#1` equals text `#2`, then do `#3`:

```
\long\def\mstr@iftext#1#2#3{\def\mstr@tmp{#2}\ifx#1\mstr@tmp#3\fi}
```

Internal Definitions.

`\mstr@lang@main` `\mstr@lang@short` `\mstr@lang@fallback` `\mstr@lang@meta` Predefine language identifiers as empty. These define the language for the document text (with and without country code), document metadata and a fallback language:

```
\let\mstr@lang@main\@empty
\let\mstr@lang@short\@empty
\let\mstr@lang@fallback\@empty
\let\mstr@lang@meta\@empty
```

Interface Definitions.

`\metatilde` `\metacomma` Define a macro for the tilde character (mostly for use within URLs); recycle the definitions from hyperxmp if available:

```
\ifdefined\xmptilde
\let\metatilde\xmptilde
\let\metacomma\xmpcomma
\else
\def\metatilde{~}
\def\metacomma{,}
\fi
```

Declare Registers.

`\metadef` Declare a register:

```
\newcommand{\metadef}[1]{%
  \mstr@csdo\let{mstr@def@#1}\relax}
```

`\mstr@verify` Verify the declaration of a register; throw an error if undeclared; disable checking for package option `checkdef=false`:

```
\newcommand{\mstr@verify}[1]{%
  \ifcsname mstr@def@#1\endcsname\else
    \PackageError{metastr}{register `#1' undefined}{}%
  \fi}
\ifmstr@opt@checkdef\else\def\mstr@verify#1{}\fi
```

Set Registers.

`\mstr@setbare` Store the register value in the macro `\mstr@data@`*reg*`@`*var*; define one argument to pass along original variant:

```
\long\def\mstr@setbare[#1]#2#3{%
  \mstr@csdo\gdef{mstr@data@#2@#1}##1{#3}}
```

`\mstr@set` Set the register value; verify whether the register has been declared:

```
\long\def\mstr@set[#1]#2#3{\mstr@verify{#2}%
  \mstr@setbare[#1]{#2}{#3}}
```

`\metaset` Interface macro for setting register with optional variant argument:

```
\newcommand{\metaset}{\@ifnextchar[{\mstr@set}{\mstr@set[]}}
```

`\mstr@unset` Clear a register value:

```
\long\def\mstr@unset[#1]#2{\mstr@verify{#2}%
  \mstr@csdotwo\global\let{mstr@data@#2@#1}\@undefined}
```

`\metaunset` Interface macro for clearing register with optional variant argument:

```
\newcommand{\metaunset}{\@ifnextchar[{\mstr@unset}{\mstr@unset[]}}
```

Register Conditionals.

`\metaif` If-then-else structure checking whether register variant is filled:

```
\long\def\metaif[#1]#2#3#4{%
  \ifcsname mstr@data@#2@#1\endcsname #3\else #4\fi}
```

`\metaifpick` If-then-else structure checking if the register in either of the variants `#1`, `\mstr@lang@main`, `\mstr@lang@short`, default and `\mstr@lang@fallback` is filled; `#1` may in fact be a comma-separated list of variants (without spaces):

```
\long\def\mstr@ifloop[#1,#2]#3#4#5{%
  \metaif[#1]{#3}{#4}{\if @#2@#5\else\mstr@ifloop[#2]{#3}{#4}{#5}\fi}}
\long\def\metaifpick[#1]#2#3#4{%
  \mstr@ifloop
    [#1,\mstr@lang@main,\mstr@lang@short,,\mstr@lang@fallback,]
    {#2}{#3}{#4}}
```

Manipulate Registers.

`\mstr@append` Append some string to a register value:

```
\long\def\mstr@append[#1]#2#3{%
  \mstr@csdotwo\let\mstr@tmpa{mstr@data@#2@#1}%
  \def\mstr@tmpb##1{\mstr@set[#1]{#2}{##1#3}}%
  \mstr@exptwo\mstr@tmpb{\mstr@tmpa{##1}}}
```

`\mstr@prepend` Prepend some string to a register value:

```
\long\def\mstr@prepend[#1]#2#3{%
  \mstr@csdotwo\let\mstr@tmpa{mstr@data@#2@#1}%
  \def\mstr@tmpb##1{\mstr@set[#1]{#2}{#3##1}}%
  \mstr@exptwo\mstr@tmpb{\mstr@tmpa{##1}}}
```

`\mstr@addsep` Append a string to a register value separated by `#1` if the string was previously filled:

```
\long\def\mstr@addsep[#1]#2#3#4{%
  \metaif[#1]{#2}{\mstr@append[#1]{#2}{#3#4}}{\mstr@set[#1]{#2}{#4}}}
```

`\metaappend` `\metaprepend` `\metaaddsep` Interface macros for appending, prepending and adding with separator:

```
\newcommand{\metaappend}{%
  \@ifnextchar[{\mstr@append}{\mstr@append[]}}
\newcommand{\metaprepend}{%
  \@ifnextchar[{\mstr@prepend}{\mstr@prepend[]}}
\newcommand{\metaaddsep}{%
  \@ifnextchar[{\mstr@addsep}{\mstr@addsep[]}}
```

Read Register Values.

`\mstr@getbare` Read a register value while passing along the original variant as an argument:

```
\def\mstr@getbare[#1]#2#3{\csname mstr@data@#2@#1\endcsname{#3}}
```

`\metaget` Interface function to read register value with mandatory variant argument in square brackets; return nothing if register clean:

```
\def\metaget[#1]#2{%
  \metaif[#1]{#2}{\mstr@getbare[#1]{#2}{#1}}{}}%
```

`\metacompose` `\metapick` `\metapick` returns a filled register value among the variants `#1`, `\mstr@lang@main`, `\mstr@lang@short`, default and `\mstr@lang@fallback` (in this order of preference), otherwise it returns nothing; `#1` may in fact be a comma-separated list of variants (without spaces); `\metacompose` sandwiches the value between `#3` and `#4` if found, and otherwise returns `#5`:

```
\long\def\mstr@composeloop[#1,#2]#3#4#5#6#7{%
  \metaif[#1]{#4}{#5\mstr@getbare[#1]{#4}{#3}#6}
    {\if @#2@#7\else\mstr@composeloop[#2]{#3}{#4}{#5}{#6}{#7}\fi}}
\long\def\metacompose[#1]#2#3#4#5{%
  \mstr@composeloop
    [#1,\mstr@lang@main,\mstr@lang@short,,\mstr@lang@fallback,]{#1}
    {#2}{#3}{#4}{#5}}%
\def\metapick[#1]#2{\metacompose[#1]{#2}{}{}{}}
```

Language Selection.

`language` Declare language register:

```
\metadef{language}
```

\metasetlang Set language and extract short forms:

```
\def\mstr@lang@split#1#2-#3@{%
  \mstr@csdo\gdef{mstr@lang@#1}{#2}}
\newcommand{\mstr@setlang@main}[1]{%
  \metaset{language}{#1}%
  \gdef\mstr@lang@main{#1}%
  \mstr@lang@split{short}#1-@%
  \metaset[short]{language}{\mstr@lang@short}%
  \metaif[meta]{language}{}{\mstr@lang@split{meta}#1-@}}
\newcommand{\mstr@setlang@meta}[1]{%
  \metaset[meta]{language}{#1}%
  \mstr@lang@split{meta}#1-@%
  \metaset[metashort]{language}{\mstr@lang@meta}}
\newcommand{\metasetlang}{%
  \@ifstar\mstr@setlang@meta\mstr@setlang@main}
```

Terms.

\metaterm \metatranslate \metasetterm Macros for filling and reading term registers:

```
\newcommand{\metaterm}{\metatranslate[]}
\def\metatranslate[#1]#2{\metapick[#1]{term-#2}}
\long\def\mstr@setterm[#1]#2#3{%
  \metadef{term-#2}\mstr@setbare[#1]{term-#2}{#3}}
\newcommand{\metasetterm}{\@ifnextchar[{\mstr@setterm}{\mstr@setterm[]}}
```

Automatic Writing to PDF.

writepdf Declare register `writepdf` to control automatic writing of metadata to PDF files:

```
\metadef{writepdf}
\metaset[auto]{writepdf}{}
\metaset[preamble]{writepdf}{}
\metaset[info]{writepdf}{}
\metaset[aux]{writepdf}{}
```

\mstr@ifwritepdf Auxiliary macro to write some type of metadata if switch activated, disable switch afterwards:

```
\long\def\mstr@ifwritepdf[#1]#2{%
  \metaif[#1]{writepdf}{#2\metaunset[#1]{writepdf}}{}}
```

\metawritepdf Write selected types of metadata to PDF file:

```
\newcommand{\metawritepdf}{%
  \mstr@ifwritepdf[preamble]{\metawritepdfpreamble}%
  \mstr@ifwritepdf[info]{\metawritepdfinfo}%
  \mstr@ifwritepdf[aux]{\metawritepdfaux}%
  \mstr@ifwritepdf[contact]{\metawritepdfcontact}%
  \mstr@ifwritepdf[rights]{\metawritepdfrights}%
}
```

\mstr@begindoc tr@begindocpreamble Hook for writing data to PDF file; this is the last chance to write the preamble set of data to the PDF (`pdflang` must be declared before hyperxmp detects languages at the end of the preamble and before hyperref sets it at the beginning of the document):

```
\newcommand{\mstr@begindoc}{%
```

```
  \mstr@ifwritepdf[auto]{\metawritepdf}}
\newcommand{\mstr@begindocpreamble}{%
  \mstr@ifwritepdf[preamble]{\metawritepdfpreamble}}
```

Hook `\mstr@begindoc` to begining of `document` block (before hooks by hyperref and hyperxmp are called, just in case); hook `\mstr@begindocpreamble` to begining of `document` block before hooks by hyperxmp are called; legacy code for latex releases earlier than 2020-10 to add hook before all other hooks are called:

```
\ifdefined\AddToHook
\DeclareHookRule{begindocument}{metastr}{before}{hyperref}
\DeclareHookRule{begindocument}{metastr}{before}{hyperxmp}
\AddToHook{begindocument}{\mstr@begindoc}
\DeclareHookRule{begindocument/before}{metastr}{before}{hyperxmp}
\AddToHook{begindocument/before}{\mstr@begindocpreamble}
\else
\AtBeginDocument{\mstr@begindoc}
\begingroup
\toks@\expandafter{\expandafter\mstr@begindocpreamble\@begindocumenthook}
\xdef\@begindocumenthook{\the\toks@}
\endgroup
\fi
```

B.3 Basic Registers

The following defines a set of basic and auxiliary registers.

Declarations.

`draft` Declare register to state draft mode:

```
\metadef{draft}
```

Set draft text (if `draft` option set), declare `draft` option:

```
\newcommand{\mstr@set@draft}{%
  \ifmstr@opt@draft\metaset{draft}{\metatranslate[##1]{draft}}%
  \else\metaunset{draft}\fi}
\define@key{\mstr@setup}{draft}[true]{%
  \csname mstr@opt@draft#1\endcsname\mstr@set@draft}%
\mstr@set@draft
```

`title` `subtitle` `author` `date` `location` `subject` `keywords` Basic registers:

```
\metadef{title}
\metadef{subtitle}
\metadef{author}
\metadef{date}
\metadef{location}
\metadef{subject}
\metadef{keywords}
```

`linebreak` Declare an auxiliary register to break a line in `print` variant and display a space otherwise:

```
\metadef{linebreak}
\metaset[]{linebreak}{ }
\metaset[print]{linebreak}{\\}
```

Title and Author Composition.

titlematter
titletext
authortext
datatext

Declare register to compose title display (analogous to `\maketitle`):

```
\metadef{titlematter}
\metadef{titletext}
\metadef{authortext}
\metadef{datatext}
```

\metatitleline
\metatitlelinetwo

Macros to print a formatted title line with one or two items; variant `skip` produces vertical skip before the item, variant `style` sets the text style, variant `sep` defines the separator between two items or undefined for two independent lines:

```
\def\metatitleline[#1]#2{%
  \metacompose[#1]{#2}
   {\metaget[skip]{#2}\begingroup\metaget[style]{#2}}
   {\par\endgroup}{}}
\def\metatitlelinetwo[#1]#2[#3]#4{%
  \metaif[sep]{#4}
   {\metacompose[#1]{#2}
     {\metaget[skip]{#2}\begingroup\metaget[style]{#2}}
     {\metacompose[#3]{#4}{\metaget[sep]{#4}}{}{}\par\endgroup}
     {\metatitleline[#3]{#4}}}
   {\metatitleline[#1]{#2}\metatitleline[#3]{#4}}}
```

Set default layout and spacing:

```
\metaset[style]{title}{\LARGE}
\metaset[style]{subtitle}{\large}
\metaset[style]{draft}{\large}
\metaset[style]{author}{\large}
\metaset[style]{location}{\large}
\metaset[style]{date}{\large}
\metaset[skip]{subtitle}{\vspace{1.5em}}
\metaset[skip]{draft}{\vspace{1.5em}}
\metaset[skip]{author}{\vspace{3em}}
\metaset[skip]{location}{\vspace{1.5em}}
\metaset[skip]{date}{\vspace{1.5em}}
\metaset[sep]{subtitle}{ -- }
\metaset[sep]{date}{, }
\metaset[sep]{draft}{: }
```

Preset for `titlematter`, `titletext`, `authortext` and `datatext` in generic and `print` variants:

```
\metaset{titletext}{%
  \metacompose[#1]{draft}{}{\metaget[sep]{draft}}{}%
  \metapick[#1]{title}%
  \metacompose[#1]{subtitle}{\metaget[sep]{subtitle}}{}{}}
\metaset{authortext}{\metapick[#1]{author}}
\metaset{datatext}{%
  \metacompose[#1]{location}{}{\metaget[sep]{date}}{}%
  \metapick[#1]{date}}
\metaset{titlematter}{%
  \metapick[#1]{titletext}%
  \metapick[#1]{authortext}%
  \metapick[#1]{datatext}%
  \metaget[skip]{titlematter}}
\metaset[print]{titletext}{%
  \metatitleline[print]{title}%
```

```
  \metatitlelinetwo[print]{subtitle}[print]{draft}}
\metaset[print]{authortext}{%
  \metatitleline[print]{author}}
\metaset[print]{datatext}{%
  \metatitlelinetwo[print]{location}[print]{date}}
```

Legacy presets:

```
\ifmstr@opt@titlematter\else
\metaset{titlematter}{%
  \metapick[#1]{titletext}%
  \metaget[skip]{titlematter}}
\metaset[print]{titletext}{%
  \metatitleline[print]{title}%
  \metatitlelinetwo[print]{subtitle}[print]{draft}%
  \metapick[print]{authortext}%
  \metapick[print]{datatext}}
\fi
```

Further Registers.

`url` `urlmessage`

Registers for document URL and message to display it:

```
\metadef{url}
\metadef{urlmessage}
```

Print URL as hyperlink:

```
\metaset[print]{url}{\url{\metaget[]{url}}}
```

URL message default text (translated):

```
% \metaset{urlmessage}{%
%   The current version of this work can be found at:
%   \metapick[#1]{url}.}
```

`partof` `partofmessage`

Registers for document URL and message to display it:

```
\metadef{partof}
\metadef{partofmessage}
```

part of message default text (translated):

```
% \metaset{partofmessage}{%
%   This document is part of the work: \metapick[#1]{partof}.}
```

`source`

Register for source name:

```
\metadef{source}
```

Write to PDF.

`etawritepdfpreamble`

Write some registers to PDF that need to be written before the start of the document:

```
\newcommand{\metawritepdfpreamble}{\ifdefined\hypersetup
  \metaif[]{language}
    {\hypersetup{pdflang={\metaget[]{language}}}}{}%
  \ifdefined\xmptilde
    \metaif[meta]{language}
      {\hypersetup{pdfmetalang={\metaget[meta]{language}}}}{}%
  \fi\fi}
```

\metawritepdfinfo Write the basic registers to PDF; also write alternative language representations of `pdftitle` and `pdfsubject`:

```
\newcommand{\metawritepdfinfo}{\ifdefined\hypersetup
  \metaifpick[\mstr@lang@meta]{author}
    {\hypersetup{pdfauthor={\metapick[\mstr@lang@meta]{authortext}}}}{}%
  \metaifpick[\mstr@lang@meta]{title}
    {\hypersetup{pdftitle={\metapick[\mstr@lang@meta]{titletext}}}}{}%
  \metaifpick[\mstr@lang@meta]{subject}
    {\hypersetup{pdfsubject={\metapick[\mstr@lang@meta]{subject}}}}{}%
  \metaifpick[\mstr@lang@meta]{keywords}
    {\hypersetup{pdfkeywords={\metapick[\mstr@lang@meta]{keywords}}}}{}%
  \ifdefined\xmptilde
    \metaif[altlang]{title}{%
      \@for\mstr@tmp:=\mstr@data@title@altlang{}\do{%
        \metaifpick[\mstr@tmp]{title}
          {\XMPLangAlt{\mstr@tmp}{pdftitle=
            {\metapick[\mstr@tmp]{titletext}}}}{}}%
      \metaunset[altlang]{title}}{}%
    \metaif[altlang]{subject}{%
      \@for\mstr@tmp:=\mstr@data@subject@altlang{}\do{%
        \metaifpick[\mstr@tmp]{subject}
          {\XMPLangAlt{\mstr@tmp}{pdfsubject=
            {\metapick[\mstr@tmp]{subject}}}}{}%
      \metaunset[altlang]{subject}}{}%
  \fi\fi}
```

\metawritepdfaux Write auxiliary registers to PDF:

```
\newcommand{\metawritepdfaux}{\ifdefined\hypersetup\ifdefined\xmptilde
  \metaif[]{url}
    {\hypersetup{pdfurl={\metaget[]{url}}}}{}%
  \hypersetup{pdfsource={}}%
  \metaif[]{source}
    {\hypersetup{pdfsource={\metaget[]{source}}}}{}%
\fi\fi}
```

B.4 Copyright and License

The following defines some registers concerning copyright and licensing.

Rights Composition.

rightstext Declare a register to compose copyright and license information:

```
\metadef{rightstext}
```

Define generic version of composition register:

```
\metaset{rightstext}{%
  \metaifpick[]{partof}{\metacompose[#1]{partofmessage}{}{ }{}}{}%
  \metapick[#1]{copyrightstatement}%
  \metacompose[#1]{copyrightmessage}{ }{}{}%
  \metacompose[#1]{licensemessage}{ }{}{}%
  \metaif[]{licenseurl}{\metacompose[#1]{licenseurlmessage}{ }{}{}}{}%
  \metaif[]{url}{\metacompose[#1]{urlmessage}{ }{}{}}{}%
}
```

Define print version of composition register; variant `sep` contains code to separate parts of the message:

```
\metaset[skip]{rightstext}{\par\addvspace\medskipamount}
\metaset[print]{rightstext}{%
  \metaifpick[]{partof}{%
    \metacompose[#1]{partofmessage}{}{\metaget[skip]{rightstext}}{}}{}%
  \metacompose[#1]{copyrightstatement}{}{\metaget[skip]{rightstext}}{}%
  \metacompose[#1]{copyrightmessage}{}{ }{}%
  \metacompose[#1]{licensemessage}{}{ }{}%
  \metaif[]{licenselogo}{%
    \metacompose[#1]{licenselogomessage}
    {\metaget[skip]{rightstext}}{\metaget[skip]{rightstext}}{}}{}%
  \metaif[]{licenseurl}{\metacompose[#1]{licenseurlmessage}{ }{ }{}}{}%
  \metaif[]{url}{\metacompose[#1]{urlmessage}
    {\metaget[skip]{rightstext}}{}{}}{}
  \metacompose[#1]{attributionmessage}{\metaget[skip]{rightstext}}{}{}%
}
```

Copyright Composition.

`copyright...` Declare registers to specify document copyright:

```
\metadef{copyrightmark}
\metadef{copyrightdate}
\metadef{copyrightowner}
\metadef{copyrightstatement}
\metadef{copyrightmessage}
```

`copyrightmark` The copyright sign or word:

```
\metaset{copyrightmark}{Copyright}
\metaset[print]{copyrightmark}{\copyright}
```

`copyrightstatement` Assemble the copyright statement from available fragments; proper spacing makes this a bit tedious:

```
\metaset{copyrightstatement}{\metaifpick[#1]{copyrightdate}%
  {\metapick[#1]{copyrightmark} \metapick[#1]{copyrightdate}%
  \metacompose[#1]{copyrightowner}{ }{}{}.}
  {\metaifpick[#1]{copyrightowner}
   {\metapick[#1]{copyrightmark} \metapick[#1]{copyrightowner}.}{}}}
```

License Composition.

`license...` Declare registers to specify document license:

```
\metadef{licenseversion}
\metadef{licenseprovider}
\metadef{licensemessage}
\metadef{licenselogo}
\metadef{licenselogomessage}
\metadef{licenseurl}
\metadef{licenseurlmessage}
```

`attributionmessage` Declare a register for the attribution message:

```
\metadef{attributionmessage}
```

licenseurlmessage — Message to declare URL at which the relevant license or further details can be found (translated):

```
% \metaset{licenseurlmessage}{%
%   To view a copy of this license, visit: \metapick[#1]{licenseurl}.}
```

licenseurl — In `print` version, pass plain `licenseurl` through `\url`:

```
\metaset[print]{licenseurl}{\url{\metaget[]{licenseurl}}}
```

licenselogo
licenselogomessage — Display license logo, by default align centrally; abuse the variant argument for passing the file name argument to `\includegraphics`:

```
\metaset[print]{licenselogomessage}{%
  \centerline{\metapick[#1]{licenselogo}}}
\metaset[cmd]{licenselogo}{\includegraphics{#1}}
\metaset[print]{licenselogo}{%
  \IfFileExists{\metaget[]{licenselogo}.pdf}%
    {\mstr@getbare[cmd]{licenselogo}{\metaget[]{licenselogo}}}%
    {\mstr@getbare[cmd]{licenselogo}{\metaget[nocurr]{licenselogo}}}}
```

Write to PDF.

\metawritepdfrights — Write rights information (rights text, alternative language representations, license url) to PDF via hyperxmp:

```
\newcommand{\metawritepdfrights}{\ifdefined\hypersetup\ifdefined\xmptilde
    \metaifpick[\mstr@lang@meta]{rightstext}
      {\hypersetup{pdfcopyright=
        {\metapick[\mstr@lang@meta]{rightstext}}}}{}%
    \metaif[altlang]{rightstext}
      {\@for\mstr@tmp:=\mstr@data@rightstext@altlang{}\do
        {\XMPLangAlt{\mstr@tmp}{pdfcopyright=
          {\metapick[\mstr@tmp]{rightstext}}}}{}
    \metaifpick[\mstr@lang@meta]{licenseurl}
      {\hypersetup{pdflicenseurl=
        {\metapick[\mstr@lang@meta]{licenseurl}}}}{}%
\fi\fi}
```

Copyright Presets.

copyright@... — Declare some copyright presets:

```
\metadef{copyright@plain}
\metadef{copyright@parts}
\metadef{copyright@doc}
\metadef{copyright@doc-parts}
\metadef{copyright@reserved}
\metadef{copyright@publicdomain}
```

plain
parts
doc
doc-parts
reserved — Some plain copyright messages (translated):

```
% \metaset{copyright@plain}{%
%   This work is protected by copyright.}
% \metaset{copyright@parts}{%
%   This work as well as its parts is protected by copyright.}
% \metaset{copyright@doc}{%
%   This document is protected by copyright.}
```

```
% \metaset{copyright@doc-parts}{%
%   This document as well as its parts is protected by copyright.}
% \metaset{copyright@reserved}{All rights reserved.}
```

`publicdomain` A public domain declaration (translated):

```
% \metaset{copyright@publicdomain}
%   {This work is dedicated to the public domain.}
```

License Presets.

`license@...` Declare some license presets

```
\metadef{license@consent}
\metadef{license@consent-noncom}
\metadef{license@lppl}
```

`consent` A license to reproduce with prior written consent (translated):

```
% \metaset{license@consent}{%
%   Reproduction of any part of this work in any form
%   without prior written consent
%   \metacompose[#1]{licenseprovider}{}{}{of the author}
%   is not permissible.}
```

`consent-noncom` A license to reproduce for private, scientific and non-commercial purposes or with prior written consent (translated):

```
% \metaset{license@consent-noncom}{%
%   Reproduction of any part of this work in any form
%   without prior written consent
%   \metapick[#1]{licenseprovider}{}{}{of the author}
%   is permissible only for private, scientific and non-commercial use.}
```

`lppl` LaTeX project public license (translated):

```
\metaset[url]{license@lppl}{http://www.latex-project.org/lppl.txt}
% \metaset{license@lppl}{%
%   This work may be distributed and/or modified under the
%   conditions of the LaTeX Project Public License, either version
%   \metaif[]{licenseversion}{\metaget[]{licenseversion}}{1.3}
%   of this license or (at your option) any later version.}
```

Selection Code.

`\metacopyright` `\metalicense` Set a copyright or license message:

```
\newcommand{\metacopyright}[1]{%
  \metaset{copyrightmessage}{\metapick[##1]{copyright@#1}}}
\newcommand{\metalicense}[1]{%
  \metaset{licensemessage}{\metapick[##1]{license@#1}}%
  \metaif[url]{license@#1}{%
    \metaset{licenseurl}{\metaget[url]{license@#1}}}{}}
```

B.5 Creative Commons

The following implements the scheme of Creative Commons licenses.

Declarations.

cc@type
cc@class

`cc@type` stores the selected CC license type; `cc@class` is '`@zero`' for the CC0 public domain dedication and empty otherwise:

```
\metadef{cc@type}
\metadef{cc@class}
```

Text Components and Internationalisation. CC license declarations are composed from several elements which can be conveniently internationalised.

The following registers store various terms used in CC licenses:

```
\metasetterm{cc@sep}{-}
\metasetterm{cc@quotel}{\textquotedblleft}
\metasetterm{cc@quoter}{\textquotedblright}
\metasetterm{cc@cc}{Creative Commons}
\metasetterm{cc@zero}{CC0}
\metasetterm{cc@by}{Attribution}
\metasetterm{cc@sa}{ShareAlike}
\metasetterm{cc@nd}{NoDerivatives}
\metasetterm{cc@nc}{NonCommercial}
\metasetterm{cc@unported}{Unported}
\metasetterm{cc@generic}{Generic}
\metasetterm{cc@intl}{International}
\metasetterm{cc@univ}{Universal}
\metasetterm{cc@pd}{Public Domain}
\metasetterm{cc@license}{License}
\metasetterm{cc@pddecl}{Public Domain Dedication}
```

cc@pd
cc@zero
cc@by...

The following template registers store the combinations for the various CC licenses:

```
\metadef{cc@pd}
\metadef{cc@zero}
\metadef{cc@by}
\metadef{cc@by-sa}
\metadef{cc@by-nd}
\metadef{cc@by-nc}
\metadef{cc@by-nc-sa}
\metadef{cc@by-nc-nd}
```

Fill the registers:

```
\metaset{cc@zero}{\metatranslate[#1]{cc@zero}}
\metaset{cc@by}{\metatranslate[#1]{cc@by}}
\metaset{cc@by-sa}{%
  \metatranslate[#1]{cc@by}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@sa}}
\metaset{cc@by-nd}{%
  \metatranslate[#1]{cc@by}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@nd}}
\metaset{cc@by-nc}{%
  \metatranslate[#1]{cc@by}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@nc}}
\metaset{cc@by-nc-sa}{%
  \metatranslate[#1]{cc@by}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@nc}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@sa}}
\metaset{cc@by-nc-nd}{%
```

```
  \metatranslate[#1]{cc@by}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@nc}\metatranslate[#1]{cc@sep}%
  \metatranslate[#1]{cc@nd}}
```

cc@n.n
cc@n.n@zero

The following registers store the various versions for CC licenses:

```
\metadef{cc@1.0@zero}
\metadef{cc@1.0}
\metadef{cc@2.0}
\metadef{cc@2.5}
\metadef{cc@3.0}
\metadef{cc@4.0}
```

Fill the registers:

```
\metaset{cc@1.0@zero}{\metatranslate[#1]{cc@univ}}
\metaset{cc@1.0}{\metatranslate[#1]{cc@generic}}
\metaset{cc@2.0}{\metatranslate[#1]{cc@generic}}
\metaset{cc@2.5}{\metatranslate[#1]{cc@generic}}
\metaset{cc@3.0}{\metatranslate[#1]{cc@unported}}
\metaset{cc@4.0}{\metatranslate[#1]{cc@intl}}
```

cc@license
cc@license@zero
cc@license@pd

The following registers store the term "CC license":

```
\metadef{cc@license}
\metadef{cc@license@zero}
\metadef{cc@license@pd}
```

Fill the registers (translated):

```
% \metaset{cc@license}{%
%   \metatranslate[#1]{cc@cc} \metatranslate[#1]{cc@license}}
% \metaset{cc@license@zero}{%
%   \metatranslate[#1]{cc@cc} \metatranslate[#1]{cc@pddecl}}
\metaset{cc@license@pd}{\metatranslate[#1]{cc@pddecl}}
```

cc@message
cc@message@zero

The following registers contain presets for the CC license messages:

```
\metadef{cc@message}
\metadef{cc@message@zero}
```

Fill the registers (translated):

```
% \metaset{cc@message}{%
%   This work is licensed under the
%   \metapick[#1]{licenseccfull} (\metapick[short]{licenseccfull}).}
% \metaset{cc@message@zero}{%
%   This work is dedicated to the public domain by means of the
%   \metapick[#1]{licenseccfull} (\metapick[short]{licenseccfull}).}
```

licensecc
licenseccver
licenseccfull

licensecc and licenseccver represent the name and version of the selected CC license; licenseccfull contains a full representation of the selected CC license:

```
\metadef{licensecc}
\metadef{licenseccver}
\metadef{licenseccfull}
```

Fill the registers:

```
\metaset{licensecc}{%
  \metapick[#1]{cc@\metaget[]{cc@type}}}
```

```
\metaset{licenseccver}{%
  \metaget[]{licenseversion}
  \metapick[#1]{cc@\metaget[]{licenseversion}\metaget[]{cc@class}}}
\metaset{licenseccfull}{%
  \metapick[#1]{cc@license\metaget[]{cc@class}}
  \metatranslate[#1]{cc@quotel}%
  \metapick[#1]{licensecc}
  \metapick[#1]{licenseccver}%
  \metatranslate[#1]{cc@quoter}}
```

License Identifier.

`ident` Compose the license identifier by `ident` variant:

```
\metaset[ident]{licenseccver}{\metaget[]{licenseversion}}
\metaset[ident]{licenseccfull}{%
  \metapick[ident]{licensecc} \metaget[ident]{licenseccver}}
\metasetterm[ident]{cc@sep}{-}
\metasetterm[ident]{cc@cc}{CC}
\metasetterm[ident]{cc@by}{BY}
\metasetterm[ident]{cc@sa}{SA}
\metasetterm[ident]{cc@nd}{ND}
\metasetterm[ident]{cc@nc}{NC}
\metasetterm[ident]{cc@zero}{CC0}
```

`short` Compose the short license identifier by `short` variant:

```
\metaset[short]{licensecc}{%
  \metaget[short]{cc@license\metaget[]{cc@class}}%
  \metapick[short]{cc@\metaget[]{cc@type}}}
\metaset[short]{licenseccver}{\metaget[]{licenseversion}}
\metaset[short]{licenseccfull}{%
  \metapick[short]{licensecc} \metaget[short]{licenseccver}}
\metasetterm[short]{cc@sep}{-}
\metasetterm[short]{cc@cc}{CC}
\metasetterm[short]{cc@by}{BY}
\metasetterm[short]{cc@sa}{SA}
\metasetterm[short]{cc@nd}{ND}
\metasetterm[short]{cc@nc}{NC}
\metasetterm[short]{cc@zero}{CC0}
\metaset[short]{cc@license}{CC }
\metaset[short]{cc@license@zero}{}
```

License Logo.

`logo` `logo` variants used for the license logo provided by the doclicense package:

```
\metaset[logo]{cc@pd}{doclicense-CC-pd}
\metaset[logo]{cc@zero}{doclicense-CC-zero}
\metaset[logo]{cc@by}{doclicense-CC-by}
\metaset[logo]{cc@by-sa}{doclicense-CC-by-sa}
\metaset[logo]{cc@by-nd}{doclicense-CC-by-nd}
\metaset[logo]{cc@by-nc}{doclicense-CC-by-nc}
\metaset[logo]{cc@by-nc-sa}{doclicense-CC-by-nc-sa}
\metaset[logo]{cc@by-nc-nd}{doclicense-CC-by-nc-nd}
```

Select currency versions:

```
\mstr@iftext\mstr@opt@cclogocurr{dollar}{\metaset[curr]{licenselogo}{}}
\mstr@iftext\mstr@opt@cclogocurr{euro}{\metaset[curr]{licenselogo}{-eu}}
\mstr@iftext\mstr@opt@cclogocurr{yen}{\metaset[curr]{licenselogo}{-jp}}
```

Select shape versions:

```
\mstr@iftext\mstr@opt@cclogoshape{box}{\metaset[shape]{licenselogo}{-88x31}}
\mstr@iftext\mstr@opt@cclogoshape{slim}{\metaset[shape]{licenselogo}{-80x15}}
```

`\mstr@setcclogo` Use the Creative Commons logos included in the doclicense package:

```
\newcommand{\mstr@setcclogo}{%
  \ifmstr@opt@cclogo
    \IfFileExists{doclicense.sty}{%
      \ifdefined\includegraphics
      \IfFileExists{doclicense-CC-by-88x31.pdf}
        {\metaset[size]{licenselogo}{-88x31}}{}
      \metaset{licenselogo}{%
        \metapick[logo]{licensecc}\metaget[curr]{licenselogo}%
        \metaget[shape]{licenselogo}}%
      \metaset[nocurr]{licenselogo}{%
        \metapick[logo]{licensecc}\metaget[shape]{licenselogo}}%
      \fi}{\GenericWarning{please install package 'doclicense'}}%
  \fi}
```

`icon` `icon` variants used for the license icons provided by the ccicons package:

```
\metaset[icon]{licensecc}{%
  \metaget[icon]{cc@license\metaget[]{cc@class}}%
  \metapick[icon]{cc@\metaget[]{cc@type}}}
\metaset[icon]{licenseccver}{\metaget[]{licenseversion}}
\metaset[icon]{licenseccfull}{%
  \metaget[icon]{licensecc}
  \metaget[icon]{licenseccver}}
\metasetterm[icon]{cc@sep}{}
\metasetterm[icon]{cc@cc}{\ccLogo}
\metasetterm[icon]{cc@pd}{\ccPublicDomain}
\metasetterm[icon]{cc@zero}{\ccZero}
\metasetterm[icon]{cc@by}{\ccAttribution}
\metasetterm[icon]{cc@sa}{\ccShareAlike}
\metasetterm[icon]{cc@nd}{\ccNoDerivatives}
\metasetterm[icon]{cc@nc}{\ccNonCommercial}
\metaset[icon]{cc@license}{\metatranslate[#1]{cc@cc}}
\metaset[icon]{cc@license@zero}{\metatranslate[#1]{cc@cc}}
\metaset[icon]{cc@license@pd}{\metatranslate[#1]{cc@pd}}
\metaset[icon]{copyrightmark}{\ccCopy}
```

Use euro or yen sign versions:

```
\mstr@iftext\mstr@opt@cclogocurr{euro}{
  \metasetterm[icon]{cc@nc}{\ccNonCommercialEU}}
\mstr@iftext\mstr@opt@cclogocurr{yen}{
  \metasetterm[icon]{cc@nc}{\ccNonCommercialJP}}
```

License URL.

`url` The `url` variant of compounds are used for the license URL.

```
\metaset[url]{licenseccver}{\metaget[]{licenseversion}}
```

```
\metaset[url]{licenseccfull}{%
  \metaget[url]{cc@license\metaget[]{cc@class}}/%
  \metapick[url]{licensecc}/%
  \metaget[url]{licenseccver}/}
\metaset[url]{cc@license}{https://creativecommons.org/licenses}
\metaset[url]{cc@license@zero}{https://creativecommons.org/publicdomain}
\metaset[url]{cc@zero}{zero}
\metaset[url]{cc@by}{by}
\metaset[url]{cc@by-sa}{by-sa}
\metaset[url]{cc@by-nd}{by-nd}
\metaset[url]{cc@by-nc}{by-nc}
\metaset[url]{cc@by-nc-sa}{by-nc-sa}
\metaset[url]{cc@by-nc-nd}{by-nc-nd}
```

Declare registers for internationalisation of deed URL:

```
\metadef{cc@url}
\metadef{cc@url@deed}
```

Fill registers:

```
\metaset{cc@url@deed}{}
\metaset{cc@url}{\metapick[url]{licenseccfull}\metapick[#1]{cc@url@deed}}
```

License Preset.

`\metalicensecc` Set the CC license of type '`#1`':

```
\newcommand{\metalicensecc}[1]{%
  \def\mstr@tmpl{#1}%
  \def\mstr@tmp{pd}%
  \ifx\mstr@tmpl\mstr@tmp
    \metaset{cc@class}{@pd}%
    \metacopyright{publicdomain}%
    \metaset{cc@type}{pd}%
  \else
    \def\mstr@tmp{zero}%
    \ifx\mstr@tmpl\mstr@tmp
      \metaset{cc@class}{@zero}%
      \metaif[]{licenseversion}{}{\metaset{licenseversion}{1.0}}%
    \else
      \metaset{cc@class}{}%
      \metaif[]{licenseversion}{}{\metaset{licenseversion}{4.0}}%
    \fi
    \metaset{cc@type}{#1}%
    \metaset{licenseurl}{\metapick[##1]{cc@url}}%
    \metaset{licensemessage}
      {\metapick[##1]{cc@message\metaget[]{cc@class}}}%
  \fi
  \mstr@setcclogo}
```

B.6 Contact Information

The following describes an interface to store and write contact information.

Declarations.

contact... Contact register declarations:

```
\metadef{contactaddress}
\metadef{contactpostcode}
\metadef{contactcity}
\metadef{contactregion}
\metadef{contactcountry}
\metadef{contactemail}
\metadef{contacturl}
```

Write to PDF.

metawritepdfcontact Write contact information to PDF via hyperxmp:

```
\newcommand{\metawritepdfcontact}{\ifdefined\hypersetup\ifdefined\xmptilde
  \metaifpick[\mstr@lang@meta]{contactaddress}{%
    \hypersetup{pdfcontactaddress=
      {\metapick[\mstr@lang@meta]{contactaddress}}}}{}%
  \metaifpick[\mstr@lang@meta]{contactpostcode}{%
    \hypersetup{pdfcontactpostcode=
      {\metapick[\mstr@lang@meta]{contactpostcode}}}}{}%
  \metaifpick[\mstr@lang@meta]{contactcity}{%
    \hypersetup{pdfcontactcity=
      {\metapick[\mstr@lang@meta]{contactcity}}}}{}%
  \metaifpick[\mstr@lang@meta]{contactregion}{%
    \hypersetup{pdfcontactregion=
      {\metapick[\mstr@lang@meta]{contactregion}}}}{}%
  \metaifpick[\mstr@lang@meta]{contactcountry}{%
    \hypersetup{pdfcontactcountry=
      {\metapick[\mstr@lang@meta]{contactcountry}}}}{}%
  \metaifpick[\mstr@lang@meta]{contactemail}{%
    \hypersetup{pdfcontactemail=
      {\metapick[\mstr@lang@meta]{contactemail}}}}{}%
  \metaifpick[\mstr@lang@meta]{contacturl}{%
    \hypersetup{pdfcontacturl=
      {\metapick[\mstr@lang@meta]{contacturl}}}}{}%
\fi\fi}
```

B.7 Extras

The following defines some extras to be activated by package options.

Course Metadata. Include structures for course materials:

```
\ifmstr@opt@course
```

institution
instructor
course
material
period

Declare course structures:

```
\metadef{institution}
\metadef{instructor}
\metadef{course}
\metadef{material}
\metadef{period}
```

Preset formatting styles:

```
\metaset[style]{course}{\LARGE\bfseries}
\metaset[style]{material}{\large}
```

```
\metaset[style]{institution}{\large}
\metaset[style]{period}{\large}
\metaset[style]{instructor}{\scshape\Large}
\metaset[skip]{material}{\vspace{2ex}}
\metaset[skip]{institution}{\vspace{4ex}}
\metaset[skip]{period}{\vspace{4ex}}
\metaset[skip]{instructor}{\vspace{6ex}}
\metaset[sep]{period}{, }
```

Fill `titletext` in `course` variant to display relevant title data for the course material:

```
\metaset[course]{titlematter}{%
  \metapick[course]{titletext}%
  \metapick[course]{datatext}%
  \metapick[course]{authortext}%
  \metaget[skip]{titlematter}}
\metaset[course]{titletext}{%
  \metatitleline[print]{course}%
  \metatitlelinetwo[print]{material}[print]{draft}}
\metaset[course]{authortext}{%
  \metatitleline[print]{instructor}}
\metaset[course]{datatext}{%
  \metatitlelinetwo[print]{institution}[print]{period}}
```

Legacy presets:

```
\ifmstr@opt@titlematter\else
\metaset[course]{titlematter}{%
  \metapick[course]{titletext}%
  \metaget[skip]{titlematter}}
\metaset[course]{titletext}{%
  \metatitleline[print]{course}%
  \metatitlelinetwo[print]{material}[print]{draft}%
  \metapick[course]{datatext}%
  \metapick[course]{authortext}}
\fi
```

Inherit title, subtitle, author and date:

```
\metaset{title}{\metapick[#1]{course}}
\metaset{subtitle}{\metapick[#1]{material}}
\metaset{author}{\metapick[#1]{instructor}}
\metaset{date}{\metapick[#1]{period}}
\metaset{location}{\metapick[#1]{institution}}

\fi
```

B.8 Translations

Determine all desired international versions to be loaded; use first one as fallback language:

```
\def\mstr@loadlangloop#1|#2&{%
  \mstr@csdo\let{mstr@lang@#1}\relax%
  \ifx\mstr@lang@fallback\@empty\def\mstr@lang@fallback{#1}\fi%
  \if @#2@\else\mstr@loadlangloop#2&\fi}
\expandafter\mstr@loadlangloop\mstr@opt@loadlang|&
```

English. Check whether to load English strings:

```
\ifdefined\mstr@lang@en
```

Terms:

```
\metasetterm[en]{title}{Title}
\metasetterm[en]{abstract}{Abstract}
\metasetterm[en]{copyright}{Copyright}
\metasetterm[en]{preface}{Preface}
\metasetterm[en]{part}{Part}
\metasetterm[en]{chapter}{Chapter}
\metasetterm[en]{section}{Section}
\metasetterm[en]{subsection}{Subsection}
\metasetterm[en]{paragraph}{Paragraph}
\metasetterm[en]{appendix}{Appendix}
\metasetterm[en]{page}{Page}
\metasetterm[en]{figure}{Figure}
\metasetterm[en]{table}{Table}
\metasetterm[en]{contents}{Contents}
\metasetterm[en]{listfigure}{List of Figures}
\metasetterm[en]{listtable}{List of Tables}
\metasetterm[en]{references}{References}
\metasetterm[en]{index}{Index}
\metasetterm[en]{draft}{DRAFT}
```

General purpose messages:

```
\metaset[en]{urlmessage}{%
  The current version of this work can be found at: \metapick[#1]{url}.}
\metaset[en]{partofmessage}{%
  This document is part of the work: \metapick[#1]{partof}.}
\metaset[en]{licenseurlmessage}{%
  To view a copy of this license, visit: \metapick[#1]{licenseurl}.}
```

Copyright statements:

```
\metaset[en]{copyright@plain}{%
  This work is protected by copyright.}
\metaset[en]{copyright@parts}{%
  This work as well as its parts is protected by copyright.}
\metaset[en]{copyright@doc}{%
  This document is protected by copyright.}
\metaset[en]{copyright@doc-parts}{%
  This document as well as its parts is protected by copyright.}
\metaset[en]{copyright@reserved}{All rights reserved.}
\metaset[en]{copyright@publicdomain}
  {This work is dedicated to the public domain.}
```

License statements:

```
\metaset[en]{license@consent}{%
  Reproduction of any part of this work in any form
  without prior written consent
  \metacompose[#1]{licenseprovider}{}{}{of the author}
  is not permissible.}
\metaset[en]{license@consent-noncom}{%
  Reproduction of any part of this work in any form
  without prior written consent
  \metacompose[#1]{licenseprovider}{}{}{of the author}
  is permissible only for private, scientific and non-commercial use.}
\metaset[en]{license@lppl}{%
  This work may be distributed and/or modified under the
  conditions of the LaTeX Project Public License, either version
  \metaif[]{licenseversion}{\metaget[]{licenseversion}}{1.3}
```

```
  of this license or (at your option) any later version.}
```

Creative Commons license composition:

```
\metaset[en]{cc@url@deed}{deed.en}
\metaset[en]{cc@message}{%
  This work is licensed under the
  \metapick[en]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[en]{cc@message@zero}{%
  This work is dedicated to the public domain by means of the
  \metapick[#1]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[en]{cc@license}{%
  \metatranslate[#1]{cc@cc} \metatranslate[#1]{cc@license}}
\metaset[en]{cc@license@zero}{%
  \metatranslate[#1]{cc@cc} \metatranslate[#1]{cc@pddecl}}
\metasetterm[en]{cc@sep}{-}
\metasetterm[en]{cc@quotel}{\textquotedblleft}
\metasetterm[en]{cc@quoter}{\textquotedblright}
\metasetterm[en]{cc@by}{Attribution}
\metasetterm[en]{cc@sa}{ShareAlike}
\metasetterm[en]{cc@nd}{NoDerivatives}
\metasetterm[en]{cc@nc}{NonCommercial}
\metasetterm[en]{cc@unported}{Unported}
\metasetterm[en]{cc@generic}{Generic}
\metasetterm[en]{cc@intl}{International}
\metasetterm[en]{cc@univ}{Universal}
\metasetterm[en]{cc@license}{License}
\metasetterm[en]{cc@pd}{Public Domain}
\metasetterm[en]{cc@pddecl}{Public Domain Dedication}

\fi
```

German. Check whether to load German strings:

```
\ifdefined\mstr@lang@de
```

Terms:

```
\metasetterm[de]{title}{Titel}
\metasetterm[de]{abstract}{Zusammenfassung}
\metasetterm[de]{copyright}{Urheberrechte}
\metasetterm[de]{preface}{Vorwort}
\metasetterm[de]{part}{Teil}
\metasetterm[de]{chapter}{Kapitel}
\metasetterm[de]{section}{Abschnitt}
\metasetterm[de]{subsection}{Unterabschnitt}
\metasetterm[de]{paragraph}{Absatz}
\metasetterm[de]{appendix}{Anhang}
\metasetterm[de]{page}{Seite}
\metasetterm[de]{figure}{Abbildung}
\metasetterm[de]{table}{Tabelle}
\metasetterm[de]{contents}{Inhaltsverzeichnis}
\metasetterm[de]{listfigure}{Abbildungsverzeichnis}
\metasetterm[de]{listtable}{Tabellenverzeichnis}
\metasetterm[de]{references}{Literatur}
\metasetterm[de]{index}{Index}
\metasetterm[de]{draft}{ENTWURF}
```

General purpose messages:

```
\metaset[de]{urlmessage}{%
  Die aktuelle Version dieses Werks befindet sich unter:
  \metapick[#1]{url}.}
\metaset[de]{partofmessage}{%
  Dieses Dokument ist Teil des Werks: \metapick[#1]{partof}.}
\metaset[de]{licenseurlmessage}{%
  Die Lizenz kann eingesehen werden unter:
  \metapick[#1]{licenseurl}.}
```

Copyright statements:

```
\metaset[de]{copyright@plain}{%
  Dieses Werk ist urheberrechtlich gesch\"utzt.}
\metaset[de]{copyright@parts}{%
  Dieses Werk sowie seine Teile sind urheberrechtlich gesch\"utzt.}
\metaset[de]{copyright@doc}{%
  Dieses Dokument ist urheberrechtlich gesch\"utzt.}
\metaset[de]{copyright@doc-parts}{%
  Dieses Dokument sowie seine Teile sind urheberrechtlich gesch\"utzt.}
\metaset[de]{copyright@reserved}{Alle Rechte vorbehalten.}
\metaset[de]{copyright@publicdomain}
  {Dieses Werk ist gemeinfrei.}
```

License statements:

```
\metaset[de]{license@consent}{%
  Reproduktion eines Teils dieses Werks in beliebiger Form
  ohne vorg\"angige schriftliche Erlaubnis
  \metacompose[#1]{licenseprovider}{}{}{des Verfassers}
  ist nicht gestattet.}
\metaset[de]{license@consent-noncom}{%
  Reproduktion eines Teils dieses Werks in beliebiger Form
  ohne vorg\"angige schriftliche Erlaubnis
  \metacompose[#1]{licenseprovider}{}{}{des Verfassers}
  ist nur zum privaten, wissenschaftlichen
  und nicht-gewerblichen Gebrauch gestattet.}
\metaset[de]{license@lppl}{%
  Dieses Werk darf nach den Bedingungen der LaTeX Project Public Lizenz,
  entweder Version
  \metaif[]{licenseversion}{\metaget[]{licenseversion}}{1.3}
  oder (nach Ihrer Wahl) jede sp\"atere Version,
  verteilt und/oder ver\"andert werden.}
```

Creative Commons license composition:

```
\metaset[de]{cc@url@deed}{deed.de}
\metaset[de]{cc@message}{%
  Dieses Werk ist lizensiert unter der
  \metapick[de]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[de]{cc@message@zero}{%
  Dieses Werk ist gemeinfrei deklariert mittels der
  \metapick[de]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[de]{cc@license}{%
  \metatranslate[#1]{cc@cc} \metatranslate[#1]{cc@license}}
\metaset[de]{cc@license@zero}{%
  \metatranslate[#1]{cc@cc} \metatranslate[#1]{cc@pddecl}}
\metasetterm[de]{cc@sep}{ -- }
\metasetterm[de]{cc@quotel}{\quotedblbase}
\metasetterm[de]{cc@quoter}{\textquotedblleft}
\metasetterm[de]{cc@by}{Namensnennung}
\metasetterm[de]{cc@sa}{Weitergabe unter gleichen Bedingungen}
```

```
\metasetterm[de]{cc@nd}{Keine Bearbeitungen}
\metasetterm[de]{cc@nc}{Nicht kommerziell}
\metasetterm[de]{cc@unported}{Unportiert}
\metasetterm[de]{cc@generic}{Generisch}
\metasetterm[de]{cc@intl}{International}
\metasetterm[de]{cc@univ}{Universell}
\metasetterm[de]{cc@license}{Lizenz}
\metasetterm[de]{cc@pd}{Gemeinfrei}
\metasetterm[de]{cc@pddecl}{Gemeinfrei Deklaration}

\fi
```

French. Disclaimer: professional assistance with translations needed.

Check whether to load French strings:

```
\ifdefined\mstr@lang@fr
```

Terms:

```
\metasetterm[fr]{title}{Titre}
\metasetterm[fr]{abstract}{R\'esum\'e}
\metasetterm[fr]{copyright}{Droits d'Auteur}
\metasetterm[fr]{preface}{Pr\'eface}
\metasetterm[fr]{part}{Partie}
\metasetterm[fr]{chapter}{Chapitre}
\metasetterm[fr]{section}{Section}
\metasetterm[fr]{subsection}{Sous-Section}
\metasetterm[fr]{paragraph}{Paragraphe}
\metasetterm[fr]{appendix}{Annexe}
\metasetterm[fr]{page}{Page}
\metasetterm[fr]{figure}{Figure}
\metasetterm[fr]{table}{Table}
\metasetterm[fr]{contents}{Table des Mati\‘eres}
\metasetterm[fr]{listfigure}{Table des Figures}
\metasetterm[fr]{listtable}{Liste des Tableaux}
\metasetterm[fr]{references}{R\'ef\'erences}
\metasetterm[fr]{index}{Index}
\metasetterm[fr]{draft}{BROUILLON}
```

General purpose messages:

```
\metaset[fr]{urlmessage}{%
  La version actuelle de cet \oe uvre se trouve \‘a l'adresse:
  \metapick[#1]{url}.}
\metaset[fr]{partofmessage}{%
  Ce document fait partie de la \oe uvre: \metapick[#1]{partof}.}
\metaset[fr]{licenseurlmessage}{%
  Pour voir une copie de cette licence, visitez:
  \metapick[#1]{licenseurl}.}
```

Copyright statements:

```
\metaset[fr]{copyright@plain}{%
  Cette \oe uvre est prot\'eg\'ee par le droit d'auteur.}
\metaset[fr]{copyright@parts}{%
  Cette \oe uvre ainsi que ses parties
  sont prot\'eg\'ees par le droit d'auteur.}
\metaset[fr]{copyright@doc}{%
  Ce document est prot\'eg\'es par le droit d'auteur.}
\metaset[fr]{copyright@doc-parts}{%
```

```
  Ce document ainsi que ses parties
  sont prot\'eg\'es par le droit d'auteur.}
\metaset[fr]{copyright@reserved}{Tous les droits sont r\'eserv\'es.}
\metaset[fr]{copyright@publicdomain}
  {Cette \oe uvre est du domaine public.}
```

Creative Commons license composition:

```
\metaset[fr]{cc@url@deed}{deed.fr}
\metaset[fr]{cc@message}{%
  Cette \oe uvre est mise \`a disposition selon les termes de la
  \metapick[fr]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[fr]{cc@message@zero}{%
  Cette \oe uvre est d\'eclar\'ee du domaine public par le
  \metapick[fr]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[fr]{cc@license}{%
  \metatranslate[#1]{cc@license} \metatranslate[#1]{cc@cc}}
\metaset[fr]{cc@license@zero}{%
  \metatranslate[#1]{cc@pddecl} \metatranslate[#1]{cc@cc}}
\metasetterm[fr]{cc@sep}{ -- }
\metasetterm[fr]{cc@quotel}{\guillemotleft}
\metasetterm[fr]{cc@quoter}{\guillemotright}
\metasetterm[fr]{cc@by}{Attribution}
\metasetterm[fr]{cc@sa}{Partage dans les M\^emes Conditions}
\metasetterm[fr]{cc@nd}{Pas de Modification}
\metasetterm[fr]{cc@nc}{Pas d'Utilisation Commerciale}
\metasetterm[fr]{cc@unported}{Non Transpos\'e}
\metasetterm[fr]{cc@generic}{G\'en\'erique}
\metasetterm[fr]{cc@intl}{International}
\metasetterm[fr]{cc@univ}{Universel}
\metasetterm[fr]{cc@license}{Licence}
\metasetterm[fr]{cc@pd}{Domaine Public}
\metasetterm[fr]{cc@pddecl}{Transfert dans le Domaine Public}

\fi
```

Spanish. Disclaimer: professional assistance with translations needed.

Check whether to load Spanish strings:

```
\ifdefined\mstr@lang@es
```

Terms:

```
\metasetterm[es]{chapter}{Cap\'itulo}
\metasetterm[es]{section}{Secci\'on}
\metasetterm[es]{subsection}{Subsecci\'on}
\metasetterm[es]{paragraph}{P\'arrafo}
\metasetterm[es]{title}{T\'itulo}
\metasetterm[es]{abstract}{Resumen}
\metasetterm[es]{copyright}{Derechos de Autor}
\metasetterm[es]{preface}{Prefacio}
\metasetterm[es]{part}{Parte}
\metasetterm[es]{appendix}{Ap\'endice}
\metasetterm[es]{page}{P\'agina}
\metasetterm[es]{figure}{Figura}
\metasetterm[es]{table}{Cuadro}
\metasetterm[es]{contents}{\'Indice}
\metasetterm[es]{listfigure}{\'Indice de Figuras}
\metasetterm[es]{listtable}{\'Indice de Cuadros}
```

```
\metasetterm[es]{references}{Referencias}
\metasetterm[es]{index}{\'Indice Alfab\'etico}
\metasetterm[es]{draft}{BORRADOR}
```

General purpose messages:

```
\metaset[es]{urlmessage}{%
  La versi\'on actual de esta obra se puede encontrar en:
  \metapick[#1]{url}.}
\metaset[es]{partofmessage}{%
  Este documento es parte de la obra: \metapick[#1]{partof}.}
\metaset[es]{licenseurlmessage}{%
  Para ver una copia de esta licencia, visite:
  \metapick[#1]{licenseurl}.}
```

Copyright statements:

```
\metaset[es]{copyright@plain}{%
  Esta obra est\'a protegida por derechos de autor.}
\metaset[es]{copyright@parts}{%
  Esta obra y sus partes est\'an protegidas por derechos de autor.}
\metaset[es]{copyright@doc}{%
  Este documento est\'a protegido por derechos de autor.}
\metaset[es]{copyright@doc-parts}{%
  Este documento y sus partes est\'an protegidos por derechos de autor.}
\metaset[es]{copyright@reserved}{Todos los derechos reservados.}
\metaset[es]{copyright@publicdomain}
  {Esta obra es de dominio p\'ublico.}
```

Creative Commons license composition:

```
\metaset[es]{cc@url@deed}{deed.es}
\metaset[es]{cc@message}{%
  Esta obra est\'a bajo la
  \metapick[es]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[es]{cc@message@zero}{Esta obra est\'a dedicada
  al dominio p\'ublico por la
  \metapick[es]{licenseccfull} (\metapick[short]{licenseccfull}).}
\metaset[es]{cc@license}{%
  \metatranslate[#1]{cc@license} \metatranslate[#1]{cc@cc}}
\metaset[es]{cc@license@zero}{%
  \metatranslate[#1]{cc@pddecl} \metatranslate[#1]{cc@cc}}
\metasetterm[es]{cc@sep}{-}
\metasetterm[es]{cc@quotel}{\textquotedblleft}
\metasetterm[es]{cc@quoter}{\textquotedblright}
\metasetterm[es]{cc@by}{Atribuci\'on}
\metasetterm[es]{cc@sa}{CompartirIgual}
\metasetterm[es]{cc@nd}{SinDerivadas}
\metasetterm[es]{cc@nc}{NoComercial}
\metasetterm[es]{cc@unported}{No Portada}
\metasetterm[es]{cc@generic}{Gen\'erica}
\metasetterm[es]{cc@intl}{Internacional}
\metasetterm[es]{cc@univ}{Universal}
\metasetterm[es]{cc@license}{Licencia}
\metasetterm[es]{cc@pd}{Dominio P\'ublico}
\metasetterm[es]{cc@pddecl}{Dedicaci\'on de Dominio P\'ublico}
```

```
\fi
```