The `metacapture` LaTeX package v0.9.1*
Structured metadata from authors

Joppe W. Bos
NXP Semiconductors
Belgium
joppe.bos@nxp.com

Kevin S. McCurley†
Unaffiliated
United States
latex@digicrime.com

February 13, 2026

Abstract. This document describes the `metacapture` LaTeX package that can be used to capture metadata during the compilation of a LaTeX document. This package is intended for use by document class designers as part of a journal publishing workflow. Authors provide their title, author information, affiliation, license, etc in macros that are used to produce the final document as well as a machine-parseable external text file. This external text file can then be used in a publishing workflow to provide HTML pages, JATS, and registration with indexing agencies like crossref. This packages comes with several implementations of a default `\maketitle` command that can be invoked from the package at load time, or the document class designer can design their own using documented internal variables of the package.

1 Motivation

The original goal of TeX was focused on typesetting and the appearance of the output on paper. With the later invention of LaTeX, Lamport advised authors that

> As you are writing your document, you should be concerned with its logical structure, not its visual appearance. The LaTeX approach to typesetting can therefore be characterized as *logical design*. [5, §1.4]

Users were encouraged to use high-level macros like `\section`, and leave the decisions like how much space to put before or after a section to the style that is

*Footnotes on titles do not use `\thanks`
†Authors are allowed to have footnotes on their name.

used. This separation between structure and appearance is an example of a more general concept from computer science known as "separation of concerns". The goal of the `metacapture` package is to extend this concept to metadata about a publication. Authors provide their metadata (e.g., title, subtitle, keywords, license, author names, emails, ORCID, funding, affiliations, etc) without any styling, and the display of this is left completely to the document class.

We should mention that we insist on metadata consisting of only text elements, and not LaTeX macros. We allow some simple things like accents `\"u` to slip through because they are easily converted to just text in a post-processing step. We also allow mathematics inside titles and abstracts. One of the purposes for the `metacapture` package is to check that authors comply with the restriction to avoid user-defined macros in their metadata. Authors are still able to use macros and stylized text in their titles, subtitles, and abstracts, but we require authors to also supply "plain text" versions of all metadata.

1.1 Previous approaches

The `\author` macro represents a fundamental limitation of the original LaTeX `article` class, because authors are asked to include formatting as part of their author list using newline codes and macros such as `\and` and `\thanks`. If you peek under the covers, the default implementation of the `\and` macro used to separate authors in the `\author` macro is given in `latex.ltx` as

```
\DeclareRobustCommand\and{%   % \begin{tabular}
  \end{tabular}%
  \hskip 1em \@plus.17fil%
  \begin{tabular}[t]{c}}%     % \end{tabular}
```

The `\and` macro therefore ends up serving two purposes, namely as a delimiter between author markup blobs and as a spacing instruction. This clearly violates the separation of concerns principle because it mixes structure and appearance. Some document classes such as `acmart` and `llncs` redefine the `\and` macro for other purposes.

In almost all cases, authors need to associate other metadata elements with their name, such as email, affiliations, ORCID, funding, etc. Depending on which document class they use, authors have various choices for this such as the `\thanks` macro to create footnotes, or things like the `\orcidlink` macro of the `orcidlink` package. Both of these are implemented as visual display macros, which again violates the separation of concerns principle.

More modern document classes have recognized the need for metadata to be associated with articles and authors, and each of them has invented their own way to encode this data. The `llncs` class extends `article` and still uses a single `\author` macro with authors separated by `\and`, but intersperses other macros like `\orcidID` and `\inst` inside the `\author` macro to annotate the individual authors. The implementation of the `\inst` and `\orcidID` macros are still based on layout rather than structure. The `IEEEtran` class also takes this approach.

The `acmart`, `amsart`, and `revtex4-2` document classes all use a sequence of `\author` commands for each author, with intervening macros such as `\orcid`, `\affiliation`, `\email`, etc. to describe the metadata for each author. The `acmart` package also defines a `\additionalaffiliation` macro in case the layout of affiliations takes too much space, but this places the burden of layout back on the author instead of the document class. The `elsarticle` document class also uses a sequence of `\author` macros for the authors, but the main argument contains embedded footnote marks. Email addresses and home page links are inserted by intervening `\ead` macros. Affiliations are specified with a main argument that consists of a set of key-value pairs. This bears some resemblance to our approach, but the style of metadata entry determines the styling of frontmatter, which once again violates the separation of concerns principle.

There have been several packages such as `titling`[1] and `authblk`[2] that offer some flexibility in how authors provide their metadata, but none of them are sufficiently detailed for modern metadata requirements.

2 Standard metadata schemas in publishing

There has unfortunately been no effort among LaTeX document classes to standardize the syntax for entering metadata, or even which fields to associate with an author.[3] This is annoying for authors who try to adapt their LaTeX from one publisher format to another. Moreover, the metadata associated with authors and articles has become increasingly complicated over the years, with new requirements to identify authors by a unique ID (ORCID), as well as the need to identify institutions by their ROR ID[4] and a need to identify funding sources with standard identifiers like the funder ID.[5]

In the world of scholarly journal publishing, there have been several efforts to standarize schemas for metadata. One of the best examples of this is the schema used by `crossref.org` for requests to register a DOI.[6] Another well-designed schema is described in the Journal Article Tag Standard (JATS)[7] that is used as a structured document format by many publishers. We took our guidance from these two schemas in how to represent metadata in a LaTeX document. In fact, our workflow creates both the crossref format and the `<front>` and `<back>` sections of a JATS document. This package does not attempt to cover all possible metadata associated with an article, but see section 8. Authors and affiliations are listed independently, with an `inst` argument for an author to indicate which affiliation is associated to an author. Funding is associated to the document itself rather than the author, in keeping with the schemas provided by `crossref.org` and JATS.

[1] See https://ctan.org/pkg/titling
[2] See https://ctan.org/pkg/authblk
[3] For example, the `llncs` class associates emails with affiliations instead of authors.
[4] See https://ror.org/
[5] See https://www.crossref.org/services/funder-registry/
[6] See https://data.crossref.org/reports/help/schema_doc/5.4.0/index.html
[7] See https://jats.nlm.nih.gov/publishing/tag-library/1.4/

3 Our solution

The processing of metadata really has four parts to it:

1. Authors use LaTeX macros to supply their metadata in a well-structured format.

2. When compiled, the metadata is used to perform visual markup of the front matter (e.g., title, authors, affiliations, keywords, etc). This is completely under the control of the document class, but this package supplies multiple versions of the `\maketitle` macro to assist in the process.

3. When the article is published, the metadata is *extracted* from the author-supplied document and used in the publishing workflow. This author-supplied metadata is combined with publisher-supplied metadata such as volume number, issue number, dates, etc. All of the metadata can then be registered with indexing agencies and used to supply structured data for the journal web pages and later harvesting agents.

4. The metadata may be embedded into the output PDF or HTML.

This package addresses all four steps, and they are addressed in the subsections that follow.

3.1 Author-supplied metadata

The primary macros used by authors are `\title`, `\subtitle`, `\addauthor`, `\addaffiliation`, `\addfunding`, `\license`, and `\addkeywords`. A complete description of these is in Section 5. The author enters only the data with these macros, omitting all formatting of how authors are to be displayed in the front matter. Macros other than accents are forbidden in the primary argument to `\author`, and in particular `\thanks` is disabled. This is so that the package can clearly identify the name of the author. Any attributes to the author such as email are added as optional key-value pairs (`\thanks` is replaced by a `footnote` attribute to `\addauthor`).

In our first implementation of metadata capture [2], the metadata extraction was intertwined with the document class `iacrcc` [1]. In this `metacapture` package we have separated out the macros to capture the metadata from the formatting of metadata. This completes the separation of metadata capture from document formatting, and allows document classes to style their documents however they like.

3.1.1 Abstracts

There is a bit of ambiguity in what constitutes "metadata" about an article. While we have attempted to cover the most important elements, we also list some additional elements in Section 8. One element that is problematic is abstracts. These are supported metadata in the Crossref schema, and they have encouraged

publishers to submit them as part of the Initiative for Open Abstracts (I4OA). While abstracts can be useful for summarization and discovery, there are a few problems associated with treating abstracts as metadata. For one thing, some journals treat them as copyrighted material, whereas many institutions like the Research Library Association argue that metadata should be made available under a CC0 license. Quite a few publishers incuding ACM, Elsevier, and Springer were withholding abstracts from their metadata in 2020.

Another complication that arises with abstracts is in formatting. The crossref schema an abstract to be formatted in JATS format, and the conversion from LaTeX to JATS can be problematic. Some authors treat their abstracts as mini-articles and use all sorts of formatting including displayed equations, bibliographic references, tables, bulleted lists, etc. They also often use user-defined macros in their abstracts. For this reason, it can be compilicated to encode abstracts as metadata.

Due to the complications associated with abstracts, we have decided to pursue a middle ground between trying to restrict author content in abstracts and successfully capturing an abstract that can easily be encoded as metadata. We do not modify the `abstract` environment, but instead have a load-time option to require a `textabstract` environment that will result in the contents being captured to an external file.

3.2 Display of metadata

When it is displayed, the metadata of an article is called the "front matter", and there are many different styles for this to be displayed, often with a custom `\maketitle` macro. Despite the name, the `\maketitle` macro is often responsible for display of author information, and sometimes also responsible for display of abstract, keywords, and license. The display of front matter can be quite complicated, with authors having multiple affiliations, authors sharing affiliations, footnotes attached to titles and authors, etc. For example, `https://arxiv.org/pdf/2210.03375` has hundreds of authors, 75 affiliations, and 12 footnotes on author names (not surprisingly, they omit email addresses).

The author metadata is inherently *relational*, with authors related to their affiliations, and other attributes. These relationships are often represented visually with footnote structures.[8] There are numerous common styles for displaying this information, including listing author affiliations under each author's name (repeating the information), or using footnotes to show affiliations for authors, or grouping authors together for a given institute, or authors ordered in some way (e.g., alphabetically or randomly). The `amsart` class places the affiliations *after* the body of the article as endnotes, and so does `OUP-EJ` for *The Economic Journal*.

[8] One problem with this is that the standard `\footnote` macro does not work inside boxes that may be used to construct the front matter. This shows up in some two-column formats because the title and author names are typically displayed in a block across both columns. We use the `footnotehyper` package to overcome this.

This package supplies several ways to display the front matter of the document. This is done by having various implementations of `\maketitle` that can be selected at load time. This particular document is typeset with the standard `article` document class, for which default values of `\@title` and `\@author` are supplied to just work out of the box with the existing `\maketitle`. Document classes are free to use one of the built-in implementations of `\maketitle`, but they can also provide their own.

At present, the styles consist of the following (visual appearance of each is displayed in Appendix A):

`iacrj` Author names are strung together in a list, with optional ORCID icons after their names, and footnotes to indicate which affiliations they belong to. Affiliations are listed individually under the block of author names. This is the official version used by the `iacrj.cls` document class for IACR journals. It is similar to the first style of `elsarticle`. See page 24.

`acmsmall` This is similar to the `acmsmall` style of `acmart.cls`, with one author per line in a vertical list, with author names in small caps followed by their affiliations and countries. See page 25.

`acmconf` This is similar to the conference proceedings style of `acmart.cls`. Each author is listed in a block with their email and affiliations underneath their name. Shared affiliations are repeated under each author's name, and links to home page and ORCID are omitted. See page 26.

`jems` Modeled after the style used for the *Journal of the European Mathematical Society*, in which author names appear before the title, keywords after the abstract, and each author has an unnumbered footnote that includes the affiliation, email, and URL. See page 27.

`inv` A left-aligned style inspired in part by the style of *Inventiones mathematicae* in that it uses blocks of text to display emails. See page 28.

`lipics` This is modeled after the style of the Dagstuhl `lipics-v2021` document class. It shows icons for the author email, homepage, and orcid. See page 29.

`ams` This is similar to what is used in `amsart`, namely a title and author names in small caps, with affiliations listed after the references. For some reason this style has author footnotes with no footnote mark, so the footnote has to mention the author to give context in the footnote. See page 30.

The visual appearance of these styles can be seen in Appendix A at the end of this document. There is also a `sample.tex` file supplied with this package that can be used to test the combination of these `\maketitle` styles with various document classes.

With the exception of the `iacrj` style, none of these represent the official styles of their respective publishers. These styles are included to allow authors to choose a preferred style, but also to demonstrate the flexibility of the schema

and to provide useful examples for document class designers who wish to to implement their own `\maketitle` using the internal variables documented in Section 7. We believe that this should simplify the construction of a `\maketitle` macro, since the variables hold only metadata without formatting.

3.2.1 Abstracts

In the original LaTeX document classes, the abstract was considered merely as a preliminary section of the document with special styling, and it would appear after the `\maketitle` macro. Some document classes have started treating the abstract as part of the frontmatter, and delegate the display of it to `\maketitle`. As a result, some document classes like `amsart`, `acmart`, `elsarticle`, and `REVTeX` now require the `abstract` environment to appear before the `\maketitle` macro. Our implementations of `\maketitle` can adapt to the `amsart`, `acmart`, and `elsarticle` document classes by invoking their internal commands to display the abstract when `\maketitle` is invoked.

There are other metadata elements that may need to be displayed, such as license, keywords, abstract, etc. The display of these is up to the document class. Our document class `iacrj` has implementations for visual display of license, abstract, and keywords, but also things like a volume number, issue number, DOI, Crossmark, etc. A document class can implement these elements in any manner they wish using the internal variables from this package that are defined in Section 7.

3.3 Capture of metadata

When a document that uses the `metacapture` package is compiled, the author-supplied metadata is extracted from the LaTeX and written into a `.meta` file that is machine-parseable. The extraction of metadata in a machine-readable format during compilation makes it easy to build publishing workflow systems around LaTeX, and this was a big part of the original motivation for this package. An example of this was used by the journal *IACR Communications in Cryptology*[9] and the publishing pipeline system for this is available as open source.[10] One part of that system is a python parser for the file containing extracted metadata that is written by the package, but it should be easy to write another parser, because the extracted metadata has a simplified yaml-like structure. The structure of this file is described in Section 6 and a sample is given in Figure 2. For more information on this workflow system, the reader is referred to [3].

Most journal production workflows are proprietary and opaque, but it appears that some use parsing tools to extract the metadata directly from the LaTeX source. Examples of this include the ACM workflow[11] and the Dagstuhl

[9] See https://cic.iacr.org/

[10] Source code available at https://github.com/IACR/latex-submit and a demo is at https://publishtest.iacr.org/.

[11] Extraction tools are mentioned in https://mirror.math.princeton.edu/pub/CTAN/macros/latex/contrib/acmart/acmart.pdf

LaTeX project.[12]. This approach can be difficult because LaTeX is a full programming language, and things like `\ifx` conditionals make it difficult to reliably parse LaTeX. This is one reason why we decided to use LaTeX itself to produce the metadata in an external file. The only real parser for TeX is the TeX binary itself, but our approach avoids the problem. It appears that the `aomart.cls` document class used for the Annals of Mathematics also follows the approach of writing metadata to an external file.

3.4 Embedding metadata in PDF

There have been multiple attempts to provide packages for embedding metadata into PDF. These include the `hyperxmp`, `pdfx`, and `xmpincl` packages. The LaTeX team is working on providing XMP metadata in the PDFs as part of their accessibility initiative [4], and we expect this to be the eventual solution. We plan to support this as part of `metacapture` when the API for the `l3pdfmeta` module becomes stable. Similar solutions should exist to inject the structured metadata into other output formats such as HTML or EPUB.

We don't require the `hyperref` package to be loaded unless the `maketitle` package option is used or the `\license` macro is used. If the `hyperref` package is loaded, then the `metacapture` package will set the PDF metadata for `pdftitle`, and `pdfkeywords`. If the `anonymous` option is not used, then it will also set `pdfauthor`. If `hyperref` is loaded, it should not be loaded with the `pdfusetitle` option.

4 Options for loading

The `metacapture` package may be loaded by the document class but may also be loaded by the author. In any event, the `metacapture` package must be loaded before the author specifies author, title, etc. `metacapture` may be loaded with various options:

`maketitle=<style>` If this is used, then the package provides a `\maketitle`. The `style` can be any of the styles listed in Section 3.2. If this is not chosen, then the class must define its own `\maketitle` that makes reference to internal variables of the package. Note that the `\maketitle` macro from `article.cls` will work out of the box, because under the covers we implement the `\@title` and `\@author` macros. This document is typeset using those values. See Section 3.2 and Appendix A.

`anonymous` If chosen, then the implementations of `\maketitle` that may be invoked with the textttmaketitle option will not disclose author names or affiliations in the PDF. A document class should load with this option if it is intending to format for a blind peer review system.

[12] See https://github.com/dagstuhl-publishing/latex

`licensereq` This required the document to specify a license with the `\license` macro. At present we only support a few licenses (see section 5.6) If a document class wishes to further restrict which license is acceptable, they can check the `\METAC@license` variable at the end of the preamble.

`countryrequired` if chosen, then every affiliation is required to declare a `country` attribute.

`cityrequired` if chosen, then every affiliation is required to declare both a `city` and a `country` attribute.

`textabstract` if chosen, then the document must specify a separate "text-only" abstract that is free of macros other than mathematics in a `textabstract` environment that contains no user-defined macros. This abstract is in addition to the ordinary `abstract` environment, and results in a file named `\jobname.abstract` that contains the abstract when the paper is compiled. We ask for such an abstract from authors so that we can capture an abstract that is suitable for indexing and HTML pages.

`emailreq` this takes one of three possible options `none,one,all` that indicates whether no emails are required for authors, at least one email is required for some author, or all authors must supply an email. This option might be used by a document class that wishes to require a corresponding author.

`orcidreq` whether each author must have an ORCID. Keep in mind that some authors may refuse to use an ORCID. The ORCID of an author should probably only be included if it is supplied by the author themself.

`notitlefootnote` when selected, the `\footnote` macro is disabled inside the main argument of `\title`

`footnotesymbols` Some of the options for `maketitle` use a different style of footnote marker for affiliations from the rest of the footnotes. For example, in the `iacrj` style the footnotes on title and authors would ordinarily be labeled as a,b,c, but they are labeled as symbols $*$, †, ‡, etc if the `footnotesymbols` option is also used. Note that this option should be used with caution, because at most 10 authors can have footnotes with this option.

5 Usage by authors

The main macros for authors that are provided by this package are `\title`, `\subtitle`, `\license`, `\addauthor`, `\addaffiliation`, `\addfunding`, and `\addkeywords`. These can only be used in the preamble before `\begin{document}`. There is also a `textabstract` environment to capture text-only versions of the abstract.

5.1 Title

A title is added using the `\title` macro, which has a number of optional arguments:

`running`	The running title intended for display in the headers.
`plaintext`	A text version of the title (mandatory if macros are used in the title).

An example using all the optional arguments is shown below.

```
\title[running   = {The iacrcc class},
       plaintext = {How to use the iacrcc LaTeX class},
      ]{How to use the \texttt{iacrcc} \LaTeX\
      class\footnote{A revision of an earlier paper on arxiv.org}}
```

The `plaintext` option is only required if you use macros in your title (it is required in the example). Inline mathematics and accents like `\"u` are allowed in the main argument to `\title`, and so are newlines `\\`. Note that LaTeX has defaulted to UTF-8 input since 2019, so just ü is preferred to `\"u`. Note also that `\thanks` is disabled inside `\title`, and `\footnote` can optionally be disabled by loading `metacapture` with the option `notitlefootnote`. See Section 5.5 for information about footnotes.

In our previous implementation from `iacrcc.cls`, we had a `subtitle` attribute, but that has now been moved into a separate `\subtitle` macro in order to support a plain text version.

5.1.1 Subtitle

An author is always allowed to have a two-line title by inserting a newline `\\` into the main argument of `\title`, but a subtitle would often be typeset in a smaller font. The semantics of a subtitle are always a little unclear, but the most common definition is for a "subordinate or explanatory title".[13] If an author wishes to have a subtitle, they use the `\subtitle` macro, which also requires an optional `plaintext` attribute if the main argument to `\subtitle` contains any macros. A full example could be:

```
\subtitle[plaintext={A LaTeX tutorial}]{%
     A \LaTeX\ tutorial\protect\footnote{Thanks to Leslie Lamport}}
```

Note that footnotes need to be protected inside a subtitle. The `notitlefootnote` option also prevents `\footnote` from being used inside `\subtitle`. A document class is free to treat subtitles in any way they see fit, but if the `\title` macro is used with the `running` attribute, then the subtitle should probably not be added to a running title.

[13] The JATS standard states that "The <subtitle> is a subordinate or auxiliary title that adds information to the full title or modifies the full title."

`inst`	A numerical list of 1-based indices specifying an institution in the affiliation array (see below).
`orcid`	The ORCID of the author, specified using the 19-character format `xxxx-xxxx-xxxx-xxxx`.
`footnote`	Create an author-specific footnote.
`surname`	Indicate the surname of the author for indexing purposes.
`onclick`	Provide a URL for the author, e.g., a home page.
`email`	Define the e-mail address of this author. Note that the load option `emailreq` may place restrictions on whether an author needs to supply an e-mail address.

Figure 1: Arguments to `\addauthor`

5.2 Authors

Author information is entered using the `\addauthor`, `\addaffiliation`, and `\addfunding` macros. Authors are asked to enter this information in a structured way so that we can provide it to indexing agencies. The `\author` macro is disabled.

Authors are listed individually using repeated calls to the `\addauthor` command, and these must appear before `\begin{document}`. The `\addauthor` macro has a number of optional arguments shown in Figure 1.

The display of these elements by a document class may be customized in any way the document designer sees fit. In some of the `\maketitle` implementations provided by `metacapture`, the presence of the `orcid` attribute creates a small clickable orcid logo next to the authors name looking like  that is a hyperlink to the authors ORCID home page. This is the authenticated logo for ORCID, but the unauthenticated version  is also bundled into this package if your journal workflow requires it. Similarly, some of our implementations of `\maketitle` display the `onclick` attribute with an icon like  or displayed next to the author's name that is an active link to the URL.

It's not obvious how to interpret the omission of the `inst` argument from `\addauthor`. It's possible that the author has no affiliation, but it's also possible that the author is affiliated with all listed affiliations. That is a matter of policy for the document class. In order to eliminate this ambiguity, the document class may choose to require the `inst` argument for every author, and use an empty `inst` argument in case the author has no affiliation. In the `\maketitle` implementations supplied in this package, we have chosen to omit the footnotes on author names for affiliations in the following cases:

- if `\addauthor` omits the `inst` attribute or it is empty,
- if there is only a single author,

- if there is only a single affiliation

In the last two cases we also omit the numbers on the affiliations. The `inst` array serves two purposes, namely for appearance to link authors to affiliations, and for metadata processing in a journal workflow where author affiliations are reported. In the latter case the indices must be validated to make sure that they refer to actual entries in the affiliation array.

Some downstream processors like crossref request author names to be broken into `given-name,surname` but this is in conflict with many existing cultural norms for author names (see [6]). `crossref` has a required element for surname, which is why we include this.

When the URL provided to the `onclick` option contain characters with a "special" meaning in LaTeX they might render incorrectly. For example, the URL

```
https://web.com/~foo/the best/#zoo
```

contains a tilde, a space, and a pound symbol #. It would be encoded as

```
onclick = {https://web.com/\%7Efoo/the\%20best\#zoo}
```

An example using all the optional arguments is given below. In this case the author has `inst={1,2}` to indicate that they are affiliated with the first and second affiliations that are entered with `\addaffiliation`:

```
\addauthor[orcid    = {0000-0000-0000-0000},
           inst     = {1,2},
           footnote = {Thanks to my supervisor for the support.},
           onclick  = {https://www.mypersonalwebpage.com},
           email    = {alice@accomplished.com},
           surname  = {Accomplished},
          ]{Alice Accomplished}
```

The `\thanks` macro is disabled inside `\addauthor`, so use the `footnote` option on `\addauthor` instead. In fact, if an author attempts to use any non-accent macros inside the primary argument to `\addauthor` it generates an error.

5.3 Affiliations

Affiliations are listed individually using the `\addaffiliation` command *after* the last author has been added using `\addauthor`. It can only be used before `\begin{document}`, and has several optional arguments:

`ror`	The Research Organization Registry (ROR) indentifier for this affiliation. This is the equivalent of ORCID for organizations. See https://ror.org/.
`department`	Department or suborganization name.
`street`	Street address.
`city`	City name.
`state`	State or province name.
`postcode`	Zip or postal code.
`country`	Country name. This is strongly recommended.
`countrycode`	ISO-3166 Alpha-2 identifier for country. This is strongly recommended, and it eliminates ambiguity in country name (e.g., Österreich vs Austria). If `country` is omitted, this can be used to fill it in. A list of these can be found at https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2.

There is an online tool at publish.iacr.org/funding to help you find ROR identifiers, and authors are strongly urged to include these. It is up to the implementation of `maketitle` to decide whether to show all attributes on an affiliation. Most implementations will use the name, `city` and `country` arguments. All arguments can be used to provide metadata to indexing agencies.

A full invocation of `\addaffiliation` would look like:

```
\addaffiliation[ror        = {05f950310},
                department = {Computer Security and Industrial Cryptography},
                street     = {Kasteelpark Arenberg 10, box 2452},
                city       = {Leuven},
                state      = {Vlaams-Brabant},
                postcode   = {3001},
                country    = {Belgium},
                countrycode = {BE}
               ]{KU Leuven}
```

5.4 Funding information

Authors should use the `\addfunding` macro to make sure that funding agencies can find articles published under their sponsorship. An example is:

```
\addfunding[fundref = {100000001},
            grantid = {CNS-1237235},
            country = {United States}]{National Science Foundation}
\addfunding[ror     = {00pn5a327},
            country = {United States}]{Rambus}
```

In this example, the author acknowledges a grant from the National Science Foundation and support from Rambus (with no `grantid`). The inclusion of

funding from an agency without a `grantid` might be appropriate if the author simply received support for a visit.

The complete list of optional arguments for `\addfunding` is:

`fundref`	An identifier from the Crossref funder registry.
`ror`	An identifier from the Research Organization Registry (ROR).
`country`	The country of the funding agency.
`countrycode`	ISO-3166 Alpha-2 identifier for country.
`grantid`	The identifier of the grant that is assigned by the agency who provided it.

You can use the online tool at `publish.iacr.org/funding` to help you find `fundref` and `ror` identifiers.

Note that `\addfunding` **does not** automatically create footnotes or an acknowledgements section to identify funding - it only collects the metadata for indexing. If you wish to include such visible annotations, you can use the `footnote` option on `\addauthor` or add a separate acknowledgements section. Some funding agencies have specific requirements for how they want to be acknowledged in the article.

5.5 Footnotes

Authors may be accustomed to using `\thanks` for footnotes indicating affiliation, email, or funding, but the `\thanks` macro is disabled and authors should use the methods described in this document. We provide the `footnote` attribute on authors so that they can add an arbitrary footnote to their name. This can be used for indicating that the author's affiliation for the work was different than their current affiliation, or to indicate contact address, or a previous name, etc. Some of the implementations of `\maketitle` use footnotes to connect authors to their affiliations. Document designers often have specific requirements on footnotes, and one such requirement is supported by the `notitlefootnote` option of this package in case footnotes are not allowed on titles.

It should be noted that footnotes are specifically tied to paper-oriented layouts, and can be problematic in HTML output.

5.6 License

When the `licensereq` option is used upon load, the author needs to provide a supported license. At present the only acceptable licenses are the following creative commons licenses: `CC-BY-4.0`, `CC-BY-NC-4.0`, `CC-BY-NC-ND-4.0`, `CC-BY-NC-SA-4.0`, `CC-BY-ND-4.0`, and `CC0-1.0`. An example would look like:

```
\license{CC-BY-4.0}
```

5.7 Keywords

Use `\addkeywords`{keyword1, keyword2} to give a list of keywords or key phrases. This is an optional macro that should appear before the abstract. Individual keywords should be separated by commas. If the keywords contains math or macros, then you must supply an additional set of text-only keywords; for example:

```
\addkeywords[rings, arithmetic on Z]{
   rings, arithmetic on $\mathbb{Z}$}
```

5.8 Abstract

A document class that loads the `metacapture` package may format the abstract however it is desired, but `metacapture` also provides a mechanism for extracting a "text-only" abstract. If the author provides such an abstract within the `textabstract` environment, it will create a file named `\jobname.abstract` that contains the contents. The purpose of the text-only abstract is to provide for indexing and production of HTML pages to describe the paper. As such, it is just as important as the classical `abstract` of a paper because it contains a textual summary that readers will use to decide if the paper is worth reading. The only difference is that the contents of the `textabstract` is constrained on what it may contain.

Note that the contents of the `textabstract` will not be displayed in the final PDF except as metadata. Note also that `\begin{textabstract}` must appear on a line by itself.

6 Format of the `.meta` file

The `metacapture-doc.meta` file that is created when a LaTeX document is compiled is similar to yaml. An example is shown in Figure 2. While this looks like yaml, it's not quite the same. The reader might wonder why we don't write yaml, and the real reason is that yaml requires enclosing strings inside double quotes if they contain any of the characters `{}[]&*#?|-<>=!\%@:`, and those characters would need to be escaped. This would be a pain to implement in LaTeX, and we don't need the full generality of yaml. The syntax of the `.meta` file is simplified by the fact that every value is on a single line.

Note that the output format may contain macros in math mode, and also a few simple macros such as `\'e`, The complete list of macros is defined in `IsMacroAllowed{}`.

7 Internal variables

For those seeking to implement their own document class based on this, you should make use of some internal variables. If a document class wishes to

```
schema:0.9.1
title: The metacapture LaTeX package
  subtitle: A demo with different styles and classes
author:
  name:Paul Erdős
  orcid:0000-1111-2222-3333
  inst:1,2
  footnote:Paul has a footnote
  email:erdos@att.com
  surname:Erdős
author:
  name:P\'al Tur\'an
  orcid:0000-0001-7890-5430
  inst:3
  footnote:Another remarkable Hungarian mathematician
  email:latex@digicrime.com
  surname:Tur\'an
affiliation:
  name:University of California, San Diego
  ror:0168r3w48
  department:Computer Science Department
  country:United States
affiliation:
  name:Mega Corporation
  department:Department of Redundancy Department
  city:Sunnydale
  state:California
  country:Elbonia
affiliation:
  name:Faber College
  country:Absurdistan
  department:Department of Unfundable Research
  city:Gottaknow
keywords: Metadata, publishing, LaTeX
license: CC-BY-4.0
```

Figure 2: Sample `.meta` file that is described in Section 6. The `schema` attribute indicates a version of `metacapture` that was used to create the file. The resst of the format should be fairly clear.

provide additional restrictions on the metadata that is provided, then they can implement additional checks on these variables at the end of the preamble. An example might be to check that every author supplied a surname, or that every author supplied an affiliation.

The most important internal variables are listed in Table 7. We believe that these are sufficient to construct any form of front matter that is desired, and we provide several implementations of a `\maketitle` command that can be accessed through the load option `maketitle=<style>`.

The first version of this package was written in LaTeX2ϵ syntax, but that made it complicated to store a list of authors, affiliations, or funding agencies. The `metacapture` package is now implemented using functionality from the LaTeX3 programming layer.[14] In particular, this means that some variable names follow the general pattern of
`\<scope>_<module>_<name>_<type>`, where

- `<scope>` is either `g` or `l` for global or local variables,
- `<module>` is the string `metac`, which we use to denote the module,
- `<name>` is a variable name,
- `<type>` is a data type.

The two most important data types from the LaTeX3 programming layer are the `seq` and `prop` data structures. The `prop` data structure is a property list, and is much like a dictionary that holds key-value pairs. This is a natural match for storing each author, which is itself a set of key-value pairs. The same goes for each affiliation and each funder. The other important data structure is `seq`, which is a sequence. We use the variable `\g_metac_author_seq` to hold the sequence of authors. Due to a limitation of the `seq` and `prop` objects, the sequences hold only serialized versions of the author `prop` rather than the `prop` object itself.[15] Finally, there is an additional datastructure called a `clist` for comma-separated list that is useful for holding the lists of keywords.

If any of the LaTeX3 variables are used in a document class, then the code has to be enclosed inside `\ExplSyntaxOn`...`\ExplSyntaxOff` groups. This is not a serious limitation, since it's much like the restriction to access variables that contain the `@` character inside `\makeatletter`...`\makeatother` blocks.

A complete tutorial on the use of `expl3` is beyond the scope of this article, but we hope that the source code of the package contains sufficiently many examples of how to use the variables.

[14] For those who are unfamiliar with this, we recommend reading https://ctan.math.washington.edu/tex-archive/macros/latex/required/l3kernel/expl3.pdf and the reference manual https://ctan.math.washington.edu/tex-archive/macros/latex/required/l3kernel/interface3.pdf.

[15] Apparently the entry of a `seq` variable can only be "balanced text" as defined in the TEX book. See https://tex.stackexchange.com/questions/115700/can-i-store-sequences-in-sequences-with-expl3 and https://github.com/latex3/latex3/issues/500 where the LaTeX team discussed such nested data structures and decided not to support them.

`\g_metac_author_seq`
: the list of authors, each of which is a serialized key-value `prop`

`\g_metac_affil_seq`
: the list of affiliations

`\g_metac_funders_seq`
: the list of funders

`\g_metac_keywords_raw_clist`
: The list of raw encoded keywords (may contain macros)

`\g_metac_keywords_plaintext_clist`
: The list of plaintext keywords

`\METAC@license`
: When `\license` is called, this is set to the license identifier. This is an SPDX identifier because of our dependence on the `doclicense` package. An example is `CC-BY-4.0`.

`\if@metacapture@anonymous`
: Set if the anonymous option is used to load it.

`\g_metac_display_emails_tl`
: This is a comma-delimited list of `email,(name)` values that were constructed from calls to the `\addauthor` macro.

`\@title`
: The formatted title supplied by the author as argument `#2` of `\title`. This does not include anything from `\subtitle`.

`\g_metac_titleraw_tl`
: The raw title supplied as the main argument to `\title`.

`\g_metac_titlerunning_tl`
: Optional running title supplied by the author.

`\g_metac_titleplain_tl`
: Optional plain text title.

`\g_metac_subtitleraw_tl`
: Optional subtitle.

`\g_metac_subtitleplain_tl`
: Optional plaintext version of subtitle.

`\METAC@listofauthors`
: A list of author names separated by ', '

`METAC@author@cnt`
: A counter for the number of authors. It is incremented each time `\addauthor` is called.

`METAC@email@cnt`
: A counter for the number of authors with email.

`METAC@affil@cnt`
: A counter for the number of affiliations. It is incremented each time `\addaffiliation` is called.

Table 1: Internal variables that are set by calls to `\addauthor`, `\addaffiliation`, `\addfunder`, `\addkeywords`, `\title`, `\subtitle`, and `\license`. Some of these are LaTeX3-specific, as indicated by the name used for them. All of these are available at the end of the preamble, because the commands to set them may only be used in the preamble.

8 What's missing

The purpose of this package is to capture author-supplied metadata rather than publisher-supplied metadata such as a DOI or page numbers. Such publisher-supplied metadata is often encoded into the PDF of a publication, e.g. as a hyperlink to the DOI. We leave the handling of publisher-supplied metadata to the document class, but the `iacrj.cls` and our open-source workflow may prove useful as an example.

The breadth of metadata for a publication has been growing in recent years. We have attempted to include only the minimal metadata elements that have clear definitions, are reported to Crossref, and are currently required in all disciplines. We expect that others may be needed in the future. This list is not complete, but some things include:

Licenses We currently only support a limited selection of licenses (e.g., we omit copyleft). It's possible that someone may wish to place different licenses on media embedded in the document. It's also possible that someone may wish to place different licenses on the LaTeX source than the final document intended for readers. We do not cover these cases.

Copyright The `acmart` document class provides the `\setcopyright` macro to stipulate addtional copyright conditions such as `usgovmixed` to stipulate that some authors are employees of the US government. Authors may also wish to declare copyright limitations on selected portions of the document. Both JATS and the crossref schema currently supports the elements `<copyright-holder>`, `<copyright-statement>`, and `<copyright-year>` that contain structured data. Both schemas allow them to be applied to subsections of the document so that a document may recognize copyright of a third party for embedded elements.

Languages We have no way for an author to express which languages are used in the document, or to provide language-specific versions of title, keywords, affiliation name, abstract, etc.

Article categories Some journals tag an article as a type, e.g, "Commentary", "Research article", "Book Review", or "Survey". These appear as `<article-categories>` in JATS.

Affiliations There are a number of other elements that might be associated with an affiliation, including address lines for a postal address, phone number, a URL, or other identifiers such as Grid, Ringgold, Scopus, etc.

XMP XMP stands for "eXtensible Metadata Platform", and is an XML standard for embedding metadata into PDF as well as other document formats. Unfortunately the schema lags badly behind other standards (it doesn't even have support for ORCID without resorting to non-standard extension schema). See Section 3.4.

Contributor roles There have been various attempts to define a taxonomy of roles played by authors. The `amsart.cls` document class allows specifying a *contributor* with `\contrib` and a *role* argument to say things like "with an appendix by N. Bourbaki" after the list of authors. They do not appear to report this information to crossref. Perhaps the best known definition of contributor roles is CRediT, which stands for Contributor Role Taxonomy, and has now become an ISO standard.[16] Crossref has announced that they will support something like this in version 5.5 of their schema. There are several things that remain to be determined, like the role of AI agents in authorship, the degree of a role, whether "translator" is a recognized role, etc.

Author bio IEEE and other publishers may collect an author bio, and JATS also supports this. The model in JATS is pretty complex and supports titled sections.

Other author IDs ORCID is pretty common now, but some authors may not have them (e.g., a deceased author) and a publisher may wish to use their own namespace (e.g., SCOPUS or MathSciNet Author ID).

Author notes Sometimes a particular author will receive a designation (e.g., a "contact author", or the author responsible for supplying data). This is in the `<author-notes>` element of JATS, and may have multiple authors referencing a single note.

Bibliographic references Since most users of LaTeX use `bibtex` or `biblatex`, it is natural to think of exporting bibliographic references as a structured part of the metadata for the article. There are several problems with this, including the fact that the fields for a BibTeX entry are not well defined and the format has failed to evolve.[17] For example, authors may add things like a URL as part of the `url` field, or a `note` field, or a `howpublished` field. Moreover, packages like `biblatex` have added additional entry types and fields.

Given the weaknesses of the BibTeX format, we might consider an alternative export format. There are several such bibliographic database formats, but they are seldom used with LaTeX, and they all suffer from deficiencies. These include RIS,[18], Endnote, Zotero,[19], citeproc JSON,[20] and JATS.[21]

In our first effort at metadata extraction [2], we used a custom BibTeX style to export the bibliography in a structured format, but that introduced additional problems because we wanted to follow the separation of

[16] See https://credit.niso.org/

[17] The original BibTeX documentation says "Don't take the field names too seriously".

[18] See https://en.wikipedia.org/wiki/RIS_(file_format)

[19] See https://gist.github.com/pchemguy/19fa69fb4e74ef0cca0026aa0dbf5f42

[20] See https://github.com/citation-style-language/schema

[21] See https://jats.nlm.nih.gov/publishing/tag-library/1.4/element/element-citation.html

concerns principle. In the end we decided that exporting bibliographic references is a big complicated mess that is better left to a high-level language. In our companion workflow software,[22] we use python to invoke `bibexport` to find the cited references, and then parse the bibtex files directly. This was complicated by the fact that we wanted to support both `biblatex` and `bibtex`.

Name parts Some agencies like crossref are attempting to gather names of authors in two parts, namely first and last (or given and family name). We have attempted to comply by allowing an optional surname field on author names, but this approach is flawed since names cannot be assumed to have the same structure across all cultures. See [6]. We also do not support alternate names for authors, and we do not support author-supplied `name-style` attribute that crossref supports for an author to report that they have only a given name.

Funding text Some funding agencies have specific text that they want to be displayed to acknowledge them. Ideally this would appear in both the document itself but also as part of the metadata on an HTML landing page. We could address this by including an optional `text` attribute on `\addfunder`

Funding groups We currently support the name, identifier, and award number for a funder, but we may wish to provide further information like the name of a PI or the program within a larger funding organization. This would be driven by downstream requirements.

Shared footnotes We don't support shared footnotes for authors. These might be useful to for a single statement that they contributed equally, or to identify all corresponding authors. We also don't support multiple footnotes on an author or a title.

Multiple departments Consider the case where author_1 is in the mathematics department of UCSB, and author_2 wishes to list both the mathematics and computer science departments in their affiliation. In this case it's not clear how the affiliations should be listed. One choice is to list UCSB twice, with author_1 specifying the mathematics department and author_2 specifying both departments in the `department` attribute. Alternatively, the UCSB affiliation would be listed once, but footnotes used on the author to indicate which department. The `acmart` document class has some support for this.

Discipline-specific data Some disciplines use additional metadata such as clinical trials that are registered with a International Standard Randomized Controlled Trial Number (ISRCTN), or the `ClinicalTrials.gov` number. We don't understand them well enough to include them here, but they seem like natural extensions.

[22] See https://github.com/IACR/latex-submit

External documents Some journal articles are explicitly linked to other documents or media. This could include supplementary material, former versions of the document, translations, related media, data, code, clinical information, etc.

Keywords and taxonomies Some disciplines also use specific taxonomies or keyword vocabularies (e.g., ACM Computing Classification System, AMS Mathematics Subject Classification, or the JEL classification system in economics). At present, we regard these as too publisher-specific be included in this general package. A document class can always provide support for them.

Both JATS and Crossref have support for keywords and/or subject classifications in their schemas. In both cases there is support for multiple classifications, with multiple vocabularies or assigning authorities.

9 Package dependencies

This package depends directly on several other packages, including the following:

`xstring` This is used for `\IfSubStr`.

`footnote` Authors are allowed to have footnotes attached to them, and these may be contained inside boxes in the `\maketitle` implementations that the package provides. For this we use the `footnote` package for footnotes inside of boxes. We tried using the `footnotehyper` package but that package is too restrictive in how footnotes are defined.

`alphalph` Footnote labels may be alphabetic, depending on the load options.

`tokcycle` This is used to perform checks on metadata arguments to make sure that they contain "only text" that can safely be written to a plain text file.

`listofitems` This is used to process a list of macros that are allowed to appear in "text-only" arguments to macros. We use `\readlist` from `listofitems` to read that list. We might be able to switch to native `clist` from `expl3` instead.

`doclicense` This is used to identify creative commons licenses in the `\license` macro.

`hyperref` This is used to provide hyperlinks on footnotes, ORCID links, and because `doclicense` requires it. We try to delay loading this as late as possible so as not to collide with any options from other packages or the document class.

`fancyvrb` This is used to write out the `textabstract` environment to a file.

`xpatch` This is used to patch an output macro from `fancyvrb`.

`tikz` This is used with the `svg.path` library to draw some icons like the home link and the ORCID link.

10 Feedback

Use the `metacapture` github project to report bugs and submit feature requests.[23] If your feature is only relevant to a specific discipline, then perhaps the natural thing to do is to extend the `metacapture` package and add additional fields. Adding too many fields and too much complexity can make the documentation hard to digest.

References

[1] Joppe W. Bos and Kevin S. McCurley. How to use the IACR Communications in Cryptology class, 2023. URL: https://publish.iacr.org/iacrcc.

[2] Joppe W. Bos and Kevin S. McCurley. Metadata in journal publishing. *TUGBoat*, 44(1):71–76, 2023. doi:10.47397/tb/44-1/tb136bos-metadata.

[3] Joppe W. Bos and Kevin S. McCurley. Lowering the cost of diamond open access journals, 2025. doi:10.48550/arXiv.2504.10424.

[4] Ulrike Fischer and Frank Mittelbach. Adding XMP metadata in latex. *TUGBoat*, 43(3):263–267, 2022. URL: https://www.tug.org/TUGboat/tb43-3/tb135fischer-xmp.pdf, doi:10.47397/tb/43-3/tb135fischer-xmp.

[5] Leslie Lamport. *LATEX : A Document Preparation System*. Addison-Wesley Publishing Company, first edition edition, 1985.

[6] Patrick McKenzie. Falsehoods programmers believe about names, 2010. URL: https://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/.

A Appendix: Example styles for `\maketitle`

This document was typeset with `article` class and the default `\@title` and `\@author` (i.e., using `maketitle=none`). In subsequent pages we show the appearance of the different styles for `\maketitle`. A class designer can of course make their own `\maketitle` to suit their own needs, and hopefully these examples will be useful.

[23] See https://github.com/IACR/latex/tree/main/metacapture

The `metacapture` LaTeX package v0.9.1[a]

Structured metadata from authors

Joppe W. Bos[1] and Kevin S. McCurley[b,2]

[1] NXP Semiconductors, Leuven, Belgium
[2] Department of Redundancy Department, Unaffiliated, San Jose, United States

This uses `maketitle=iacrj`. Footnotes for affiliations are numbered, but footnote symbols on title footnotes and author footnotes are alphabetic (they can also be symbols). The icon for a home page is different than what is used in `\@author`.

[a] Footnotes on titles do not use `\thanks`
[b] Authors are allowed to have footnotes on their name.

The `metacapture` LaTeX package v0.9.1[1]

Structured metadata from authors

Joppe W. Bos, NXP Semiconductors, Belgium
Kevin S. McCurley[2], Unaffiliated, United States

This uses `maketitle=acmsmall`. Author names are in small caps.

[1] Footnotes on titles do not use `\thanks`
[2] Authors are allowed to have footnotes on their name.

The metacapture LaTeX package v0.9.1[1]

Structured metadata from authors

Joppe W. Bos
joppe.bos@nxp.com
NXP Semiconductors
Belgium

Kevin S. McCurley[2]
latex@digicrime.com
Unaffiliated
United States

Keywords
Metadata, publishing, LaTeX

This uses `maketitle=acmconf`. Each author is displayed in a block with repeated affiliations. It appears similar to the default `\@author`, but the spacing is better for more than a couple of authors and links for ORCID and author home pages are omitted.

[1] Footnotes on titles do not use `\thanks`
[2] Authors are allowed to have footnotes on their name.

Joppe W. Bos · Kevin S. McCurley

The `metacapture` LaTeX package v0.9.1[1]

Structured metadata from authors

This uses `maketitle=jems`. Author names appear above the title, and each author has an unnumbered footnote with their information. It's not clear what to do with footnotes on author names, and the journal class appears not to support them.

Joppe W. Bos: NXP Semiconductors, Leuven, Belgium; joppe.bos@nxp.com; https://www.joppebos.com

Kevin S. McCurley: Department of Redundancy Department, Unaffiliated, San Jose, United States; latex@digicrime.com

[1] Footnotes on titles do not use `\thanks`

The `metacapture` LaTeX package v0.9.1[a]

Structured metadata from authors

Joppe W. Bos[1] and Kevin S. McCurley[2,b]

[1] NXP Semiconductors, Leuven, Belgium
[2] Department of Redundancy Department, Unaffiliated, San Jose, United States

Joppe W. Bos
joppe.bos@nxp.com

Kevin S. McCurley
latex@digicrime.com

This uses `maketitle=inv`. Affiliations are listed below each author name, and are repeated for shared affiliations. Emails are listed after affiliations in a block.

[a] Footnotes on titles do not use `\thanks`

[b] Authors are allowed to have footnotes on their name.

The metacapture LaTeX package v0.9.1[1]

Structured metadata from authors

Joppe W. Bos
NXP Semiconductors, Leuven, Belgium

Kevin S. McCurley[2]
Department of Redundancy Department, Unaffiliated, San Jose, California, United States

This uses `maketitle=lipics`. Author names have icons for email, home page, and ORCID. Affiliations are listed below each author name, and are repeated for shared affiliations.

[1] Footnotes on titles do not use `\thanks`

[2] Authors are allowed to have footnotes on their name.

The metacapture LaTeX package v0.9.1
Structured metadata from authors

Joppe W. Bos and Kevin S. McCurley

This uses `maketitle=ams`. Title and author names are in small caps. Author footnotes are unnumbered (for some reason this is the style for `amsart`). Each author's affiliation is listed at the end of the document as below.

NXP Semiconductors, Leuven, Belgium
Email address: joppe.bos@nxp.com
URL: https://www.joppebos.com

Department of Redundancy Department, Unaffiliated, San Jose, California, United States
Email address: latex@digicrime.com

Key words and phrases. Metadata, publishing, and LaTeX

Footnotes on titles do not use `\thanks`

Authors are allowed to have footnotes on their name.