BIBLIOGRAPHIES
THE CONTEXT WAY

Hans Hagen and Alan Braslau

Contents

Introduction

How Hans got involved

This manual describes how MkIV handles bibliographies. Support in ConTEXt started in MkII for bibTEX, using a module written by Taco Hoekwater. Later his code was adapted to MkIV, but because users demanded more, I decided that reimplementing made more sense than patching. In particular, through the use of Lua, the bibTEX data files can be easily directly parsed, thus liberating ConTEXt from the dependency on an external bibTEX executable. The CritEd project (by Thomas Schmitz, Alan Braslau, Luigi Scarso and myself) was a good reason to undertake this rewrite. As part that project users were invited to come up with ideas about extensions. Not all of them are (yet) honored, but the rewrite makes more functionality possible.

This manual is dedicated to Taco Hoekwater who in a previous century implemented the first bibTEX module and saw it morph into a TEX–Lua hybrid in this century. The fact that there was support for bibliographies made it possible for users to use ConTEXt in an academic environment, dominated by bibliographic databases encoded in the bibTEX format.

The subsystem described here is one of the most complex and messy of all ConTEXt subsystems. This has to do with the fact that it combines (multiple) lists and (multiple) forward and backward references, all kind of rendering of the citation as well as the entry in the list, rather complex interactivity, multiple databases, datasets and renderings and of course combinations of this. The implementation uses a mix of TEX and Lua code with so called setups as rendering specifications. At the cost of complexity (and some runtime penalty) this provides a lot of freedom and flexibility.

How Alan got involved

Bibliographies and citations are of utmost importance in any scholarly work. Nevertheless, the production of bibliography lists and the insertion of citations, just like the production of an index, is a task that is often postponed to a later stage in the writing of an article, a book, or a manual. Perhaps this is because it can be more important to create than to refer, but perhaps this is because the tools that one uses are insufficient or unnatural.

A computerized typesetting system should help an author produce a text, not impose any preset format or unnecessary constraint. In a referenced work, a bibliography system should be flexible enough to adapt to very different styles and practices. Creating such a system is quite a challenge.

ConTEXt MkII implemented a system that was based on a use of bibTEX, an external program that built upon basic bibliographic macros introduced in LATEX. ConTEXt MkIV moved away from this dependency, opening up many possibilities for new functionality and, we hoped, providing more natural and flexible tools for authors. For my own use, the most important of which is a very powerful search and match mechanism that has been made possible though the use of Lua.

I had started by asking simple questions on details of the workings of this system and making "wouldn't it be nice to" requests for functionality that I knew was somehow buried

in the inner workings of ConTEXt. As a result of these inquiries, I got drawn into the project to make this new system a reality.

How you can be involved

Bibliography management is indeed one of the most complex subsystems in ConTEXt, and many, many design decisions had to be made during its development. Experimental features were added, some of which were later abandoned as being inappropriate or else superseded by some better mechanism. The effort (and time) that we spent in reimplementing the treatment of bibliographies was much greater than any of us had anticipated when we undertook this project. Hopefully, now the system is stable enough to be more widely used and this manual is an attempt to make it accessible to all users.

There are ConTEXt users who just use whatever the bibliograpy modules provide by default. Often apa is good enough. But sometimes on the mailinglist users start asking questions that are not answered in this manual. Feel free to come up with additional examples that can be added to the test suite, or when we consider them to be of general use, to this manual.

1 Quick start, for the impatient

Without *any* explanation:

\usebtxdataset[mkiv-publications.bib**]**

\starttext

\startbodymatter
a citation: **\cite[**article,book**]**
\stopbodymatter

\startbackmatter
    \startchapter[title=Bibliography**]**
        \placelistofpublications
    \stopchapter
\stopbackmatter

\stoptext

1

a citation: [1,2]

2

Bibliography

1   F. Last, F.M. Last, F.-C. Last et al., "Article title", *Journal name* volume(number), ff–tt, YYYYb.

2   F. BookAuthorLastnameA, F.M. BookAuthorLastnameB and F.-C. BookAuthorLastnameC, *Book title* (edition ed., Vol. volume, no. number), Publisher, Address, YYYY.
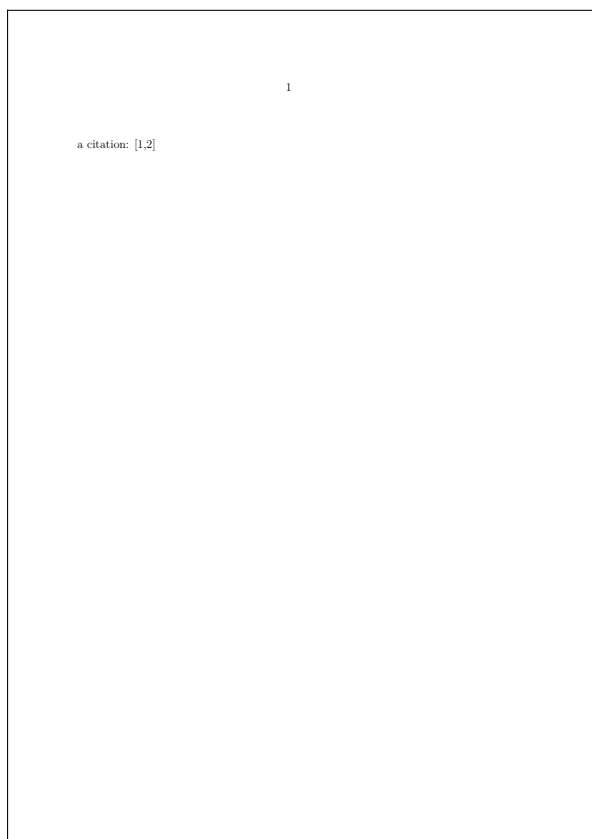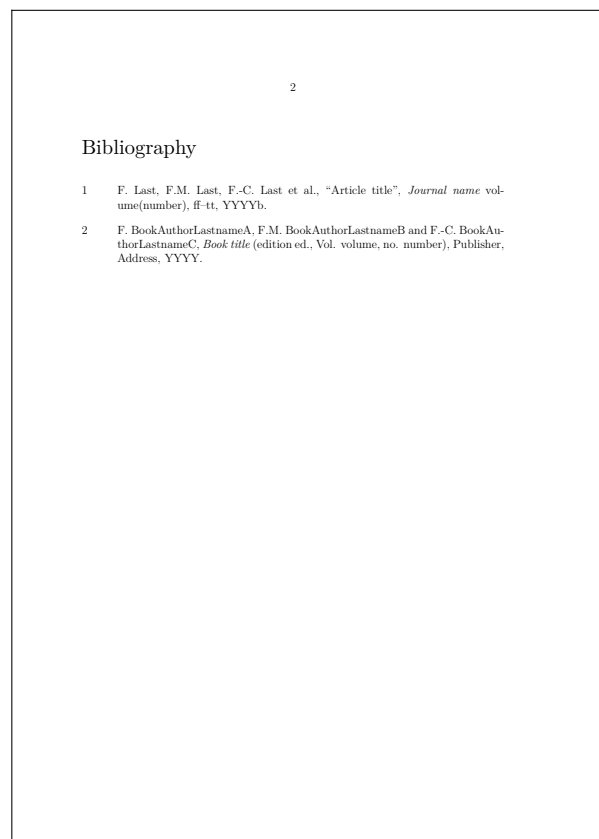
bodymatter   backmatter

This example demonstrates that the basic usage in the production of bibliography citations and lists is rather simple. However, once the user starts asking for particular customizations, then things can get quickly quite complicated.

2 The database

The bibliography subsystem uses a database (or a set of databases) to construct a list of citations to be used in a scholarly work. However, it will be shown later that the database system can be used (and abused) to many ends having little or nothing at all to do with citations and bibliographies. Nevertheless, at first we shall remain focused on the use of bibliography databases.

The data to be used must have a source and a structure. In the next sections we describe the possible input.

2.1 BIBTEX

The bibTEX format is rather popular in the TEX community and even with its shortcomings it will stay around for a while. Many publication websites can export and many tools are available to work with this database format. It is rather simple and looks a bit like Lua tables.[1]

Unfortunately the content can be (and usually is) polluted with non-standardized TEX commands which complicates pre- or post-processing outside TEX. In that sense a bibTEX database is often not coded neutrally. Some limitations, like the use of commands to encode accented characters root in the ascii world and can be bypassed by using utf instead (as handled somewhat in LATEX through extensions such as `bibtex8`).

The normal way to deal with a bibliography is to refer to entries using a unique tag or key. When a text containing a list of entries is typeset, this reference can be used for linking purposes. The list can be processed and sorted using the `bibtex` program that converts the database into something more TEX friendly (a `.bbl` file).

In ConTEXt we no longer use the (external) `bibtex` program at all: we simply parse the database files and deal with the necessary manipulations directly in ConTEXt.[2] One or more such databases can be used and combined with additional entries defined within the document. We can have several such datasets active at the same time.

On the name btx: many of the ConTEXt commands that will be used in the following contain the label `btx` in their name. This identifier was retained despite the fact that ConTEXt MkIV is now completely independent of bibTEX; it reflects the role still played by bibTEX data as a preferred source format and serves as a handy, unique identifier, both internally in the programming as well as for the user. This three-letter label is systematically used in commands that otherwise attempt to avoid cryptic-styled names.

A bibTEX file entry looks like this:

```
@Article {sometag,
    author  = "An Author and Another One",
    title   = "A hopefully meaningful title",
    journal = maps,
    volume  = "25",
    number  = "2",
```

```
    pages    = "5--9",
    month    = mar,
    year     = "2013",
    ISSN     = "1234-5678",
}
```

Entries are of the form: `@category{...}`[3][4][5] The field names are all cast to lowercase so capitalization is irrelevant; Spacing is not important and should be used advantageously for readability. The leading tag (`sometag` above) cannot contain spaces and *must* be followed by a comma.[6]

Normally a value is given between quotes (or curly brackets) but single words are also valid (as there is no real benefit in not using quotes or curly brackets, we advise to always use them). The order of the fields in an entry is inconsequential and there can be many more fields than those shown above. Instead of string values one can also use predefined shortcuts. The title for example might quite often contain TeX macros. Some fields, like `pages` have funny characters such as the endash (typically entered as `--`) so we have a mixture of data and typesetting directives. Furthermore, if you are covering non–English references, you often need characters that are not in the ascii subset. Note that ConTeXt is quite happy with utf, but if your database file uses old-fashioned TeX accent combinations then these will be internally converted automatically to utf.

Commands (macros) found in a database file are converted to an indirect call, which is quite robust. The use of commands in the database file is described in **section 2.2**.

The `author` (and `editor`) fields are parsed separating multiple authors identified by the conjunction 'and'. Each name is assumed to be in the form:

Firstname(s) Lastname

where Lastname may include an optional (nobility) particle: lower-case word(s) such as "von", "de", "de la", etc.) *unless* specifically in the two- or three-token form:

Lastname(s), Firstname(s)

Lastnames(s), Suffix(es), Firstname(s)

separated explicitly using comma(s).

bibTeX also (obscurely) supports the syntax:

Firstname(s) {Lastname(s), Suffix(es)}

we may (or may not) support this in the future. Don't use this!

We extend bibTeX by optionally parsing each name in terms of four or five tokens:

Particule(s), Lastname(s), Suffix(es), Firstname(s)

Particule(s), Lastname(s), Suffix(es), Firstname(s), Initial(s)

in order to allow a free form for the particules, irrespective of capitalization, thus avoiding the need to resort to any sort of TeX trickery [6,7]. In fact, an optional sixth token is parsed whose meaning is presently reserved for future directives describing how the name is to be interpreted:

Particule(s), Lastname(s), Suffix(es), Firstname(s), Initial(s), directives

bibTeX additionally accepts the special token `others` to be used (sparingly) to indicate an incomplete author list.

Note that most style specifications will handle the truncation of long author lists in a systematic fashion. The `and others` construction finds its use when the complete author list is not well known or ill-defined.

Fields other than `author` and `editor`, for example `artist` or `director` if one desires, can be declared to be of type 'author' and thus interpreted as names, but this is a subject for specialists.

The `keywords` field can also be split into tokens separated by semicolons (keyword; keyword; ...). This can be useful, as will be seen later, in the creation of keyword indexes, for example.

Other string values such as `title` are kept literally (except for an internal automatic conversion to utf of certain TeX strings such as accent combinations, endash, quotations, etc.). Note that the bibliography rendering style (see below) might specify a capitalization (using the ConTeXt commands `\Word` or `\Words`, for example). Capitalized Names and acronyms are respected, removing a need for the bibTeX practice of 'protecting' such words or letters with surrounding curly brackets (which here are simply stripped off).[7] As some styles might not specify the capitalization of words in the title whereas other styles might, it is recommended that strings be written in lower case except where upper case is explicitly required so as to be compatible with all such capitalization styles.[8]

String values can be enclosed indifferently between matching curly brackets: `{}` or pairs of quotation marks: `""`.

Multiple string values can be concatenated using the operator `\#`,[9] as will be illustrated in **table 3.1**.

Everything outside of a valid entry is ignored and treated as a comment. Syntactic errors (such as a missing comma or some unbalanced quotes or parenthesis) are also skipped over, i.e. ignored. This is to attempt to continue on to valid data but may lead to unexpected results. It is therefore the user's responsibility to insure the correctness of the data files. Whereas some checks and warnings are issued, the system is purposefully not too verbose.

Data is handled on a 'first come, first served' basis: duplicate *fields* in an entry are ignored[10] though duplicate *entries* (having the same tag) are retained, but the subsequent identical tags will be modified by adding a suffix $-n$ for the nth duplicate.[11] The presence of duplicate fields or tags will be flagged as such with warnings in the log file.

A special provision has been made to declare author synonyms, that is names that might occur with a variation of spellings or aliases. This shall be discussed later.

We have attempted to remain compatible with the bibTEX format, and any new bibliography extensions that we introduce here were designed in a way to remain compatible with bibTEX, being simply ignored rather than potentially generating a bibTEX error.

The bibTEX files are loaded in memory as Lua table but can be converted to xml so that we can access them in a more flexible way, but that is another subject for specialists.

2.2 Commands in entries

One unfortunate aspect commonly found in bibTEX files is that they may contain TEX commands. Even worse is that there is no standard on what these commands can be and what they mean, at least not formally, as bibTEX is a program intended to be used with many variants of TEX style: plain, LATEX, and others. This means that we need to define our use of these typesetting commands.[12] However, in most cases, they are just abbreviations or font switches and these are often well known. Therefore, ConTEXt will try to resolve them before reporting an issue. The log file will announce the commands that have been seen in the loaded databases. For instance, loading `tugboat.bib`[13] gives a long list of commands of which we show a small set of the five most frequently encountered ones here:[14]

```
publications > tugboat  tt     134 known
publications > tugboat  Dash   136 unknown
publications > tugboat  acro   137 known
publications > tugboat  LaTeX  209 known
publications > tugboat  TeX    856 known
```

Some are flagged as known and others as unknown. You can define unknown commands, or overload existing definitions in the standard way (*e.g.* \def\Dash{–}) or, alternatively, in the following way:

```
\definebtxcommand\TUB {TUGboat}
\definebtxcommand\MP  {METAPOST}
\definebtxcommand\sltt{\tt}
\definebtxcommand\<#1>{\type{#1}}
```

Custom commands created using \definebtxcommand have the advantage of using a separate name space thus allowing isolation from other ConTEXt commands.[15] Unknown commands do not stall processing, but their names are then typeset in a mono-spaced font so they probably stand out for proofreading. You can access the commands using \btxcommand{...},[16] as in:

```
commands like \btxcommand{MySpecialCommand} are handled in an indirect way
```

As this is an undefined command we get: "commands like MySpecialCommand are handled in an indirect way". Often, these embedded TEX commands are present in these files in

order to trick bibTeX into certain behavior. Since this will generally not be necessary here, we strongly encourage users to clean-up such unnecessary extras. Indeed, the idea is to keep the data clean, using styles and parameter settings instead to handle rendering issues. Indeed, we don't see it as challenge nor as a duty to support all kinds of messy definitions. Of course, we try to be somewhat tolerant, but you will be sure to get better results if you use nicely setup, consistent databases.

Finally, the bibTeX entry `@String{}` is preprocessed as expected.[17]

```
@String{j-TUGboat = "TUGboat"}
```

2.3 MKII definitions

In the old MkII setup we have two kinds of entries: the ones that come from the bibTeX run and additional user supplied ones. We no longer rely on bibTeX output but we do still support the user supplied definitions. These were in fact prepared in a way that suits the processing of bibTeX generated entries. The next variant reflects the ConTeXt recoding of the old bibTeX output. For this reason, some users refer to this as `.bbl` format.

```
\startpublication[k=Hagen:Second,t=article,a={Hans Hagen},y=2013,s=HH01]
    \artauthor[]{Hans}[H.]{}{Hagen}
    \arttitle{Who knows more?}
    \journal{MyJournal}
    \pubyear{2013}
    \month{8}
    \volume{1}
    \issue{3}
    \issn{1234-5678}
    \pages{123--126}
\stoppublication
```

The split `\artauthor` fields are collapsed into a single `author` field as we handle the splitting later when it gets parsed in Lua. The `\artauthor` syntax is only kept around for backward compatibility with the previous use of bibTeX.

In the new setup we support these variants:

```
\startpublication[k=Hagen:Third,t=article]
    \author{Hans Hagen}
    \title {Who knows who?}
    ...
\stoppublication
```

as well as

```
\startpublication[tag=Hagen:Third,category=article]
    \author{Hans Hagen}
```

```
    \title {Who knows who?}
    ...
\stoppublication
```

and

```
\startpublication
    \tag      {Hagen:Third}
    \category{article}
    \author  {Hans Hagen}
    \title   {Who knows who?}
    ...
\stoppublication
```

It will be illustrated later a means to export the database which may be of great use in converting collections of MkII bibliography files.

2.4 LUA tables

Because internally the entries are Lua tables, we also support the loading of Lua based definitions:

```
return {
    ["Hagen:First"] = {
        author   = "Hans Hagen",
        category = "article",
        issn     = "1234-5678",
        issue    = "3",
        journal  = "MyJournal",
        month    = "8",
        pages    = "123--126",
        tag      = "Hagen:First",
        title    = "Who knows nothing?",
        volume   = "1",
        year     = "2013",
    },
}
```

Notice that the tag is redundantly specified; it is 'pushed' into the table so that one can access it without knowing the tag of the original table.

2.5 XML

The following xml input is rather close in structure, and is also accepted as input.

```
<?xml version="2.0" standalone="yes" ?>
<bibtex>
```

10

```
<entry tag="Hagen:First" category="article">
    <field name="author">Hans Hagen</field>
    <field name="category">article</field>
    <field name="issn">1234-5678</field>
    <field name="issue">3</field>
    <field name="journal">MyJournal</field>
    <field name="month">8</field>
    <field name="pages">123--126</field>
    <field name="tag">Hagen:First</field>
    <field name="title">Who knows nothing?</field>
    <field name="volume">1</field>
    <field name="year">2013</field>
</entry>
</bibtex>
```

We shall focus on the use of bibTeX `.bib` files as the input data format of reference. Keep in mind, however, that the Lua table format and the xml format might prove to be more flexible for future expansion of functionality.

2.6 Other formats

Various bibliographic data file formats are in common use, such as:

savedrecs.txt	Institute of Scientific Information (ISI) tagged format (e.g. Thomson Reuters™ Web of Science™),
filename.enw	Thomson Reuters™ Endnote™ export format (there is also an Endnote `.xml` export),
filename.ris	Research Information Systems, Incorporated, now Thomson Reuters™ Reference Manager™, and
pubmed_result.txt	The National Library of Medicine® (NLM®) MEDLINE®/PubMed® data format

just to name a few (amongst many more). Filters can be easily written in Lua to read these and other bibliography data formats, although no such filters are provided. This is because the user has a choice of a certain number of bibliography database management programs that can easily convert from these to the bibTeX format.[18] Indeed, it is not the vocation of the present ConTeXt bibliography subsystem to fully manage the bibliography data sources, only to be able to use such data in the production of documents.

A note on database management programs: these are very valuable tools for the manipulation of bibliography database information, which is why the bibTeX format has so much importance for us here. However, one must be aware that these programs are not standards and many of them may introduce invalid extensions that might not even be handled correctly by bibTeX itself.

Footnotes

[1] Indeed, it is said that the bibTeX format was one of the inspirations for the constructor syntax in Lua [8, Chapter 12.].

[2] As will be seen, we attempt to use all the power of ConTeXt and Lua, thus making unnecessary much (most?) of the TeX-like encoding of the data. We encourage users to clean-up their `.bib` database files as much as possible.

[3] Anything outside of a valid `@category{...}` construction is ignored and is taken to be a comment. Within an entry, there are to be no comments but one can prefix field names, for example, to have them ignored.

[4] There is a special entry type named `@comment{...}`. The main use of such an entry type is to comment a large part of the bibliography easily, since anything outside an entry is already a comment, and commenting out one entry may be achieved by just removing its initial @. — This is perhaps of some use, although not very elegant! As one can input multiple bibliography data files, as will be seen below, it is much better practice to split datafiles for optional loading.

[5] Many bibTeX data management tools such as `jabref` (see below) will ignore and then throw-away all such handily-crafted comments and data entries turned into comments. So one must beware!

[6] The entry tag (`@category{sometag,...}`) is not to be confused with the optional field `key = sortkey,` that may also be present.

[7] Since ConTeXt uses utf, it does not suffer from all of the complicated sorting issues that plague bibTeX/LaTeX.

[8] Some bibliographic database sources can be quite sloppy and return strings (titles and even authors) in all capitals, for example. We have made the design choice *not* to follow the bibTeX practice/feature of managing all string values as we did not want to require the protection through enclosing curly brackets that would have been a necessary consequence. Thus, some cleaning of these database files might be needed.

[9] The bibTeX string concatenation operator `\#` is not to be confused with the Lua operator `..`, nor with the MetaPost operator `\&`!

[10] Some bibTeX practice allows for the concatenation of duplicate name fields (i.e. `author` and `editor`) through `and`, but (silently) ignores duplicate other fields. We choose to have a consistant behavior and disallow duplicate field occurrences.

[11] Duplicate entries using different tags will be also be maintained.

[12] In particular, one might need to redefine those that are too LaTeX-centric.

[13] `tugboat.bib` is distributed with TeXLive. Note that this file is typical of many bibTeX files that can either be downloaded or generated. It contains a certain number of TeX constructions that can cause problems, to say the least!

[14] The illustration here is taken from the log file.

[15] The isolation of `\btxcommand` allows the `.bib` files to safely contain TeX and LaTeX idiosyncrasies that might conflict with proper ConTeXt syntax.

[16] `\btxcommand` is a synonym for the internal, more cryptic `\btxcmd`.

[17] Notice that `tugboat.bib` also contains:
`@Preamble{"\input tugboat.def"} @Preamble{"\input path.sty"}`
These are silently ignored as many such commands are most likely not to be compatible with ConTeXt, indeed, the examples show here are not!

[18] Notable, open source examples are **jabref** and **zotero**.

3 Datasets

Normally in a document you will use only one bibliographic database, whether or not distributed over multiple files. Nevertheless, we support multiple database formats as well which is why we talk of datasets instead. The use of multiple datasets allows the isolation of different bibliographies (a single bibliography can nevertheless be rendered by structure element: section, chapter, part, etc. as we shall see later). A good example of the use of multiple datasets would be for a proper bibliography in addition to a reference catalog (of equipment, suppliers, software, patents, legal jurisprudence, music, ...). Indeed, datasets can be used to hold both bibliographic and non-bibliographic information.

A dataset is initiated with the `\definebtxdataset` command.[1]

\definebtxdataset[default**]**

Like other commands in ConTEXt, the dataset options can be setup using the command `\setupbtxdataset`.

```
\setupbtxdataset [...¹.] [.²=.]
                       OPT      OPT
1   IDENTIFIER

2   ...
```

```
\definebtxdataset [...¹.] [.²=.]
                                OPT
1   IDENTIFIER

2   inherits from \setupbtxdataset
```

A dataset is loaded from some source through the use of the `\usebtxdataset` command.

```
\usebtxdataset [...¹.] [...².]

1   IDENTIFIER

2   FILE
```

Here are some examples:

\usebtxdataset[tugboat**][**tugboat.bib**]**
\usebtxdataset[default**][**mtx**-**bibtex**-**output.xml**]**
\usebtxdataset[default**][**test**-**001**-**btx**-**standard.lua**]**
\usebtxdataset[default**][**mkii**-**publications.bbl**]**
\usebtxdataset[default**][**named.buffer**]**

These four suffixes are understood by the loader. Here the dataset (other than the first) has the name `default` and the four database files are merged.[2] Definitions in the document source (coded in TEX speak) are also added, and they are saved for successive runs. This means that if you load and define entries, they will be known at a next run beforehand, so that references to them are independent of when loading and definitions take place. This

is convenient to eventually break-up the dataset loading calls to relevant sections of the document structure.

In this document we use some example databases, so let's load one of them now:[3]

\usebtxdataset[example**][**mkiv**-**publications.bib**]**

The beginning of the file `mkiv-publications.bib` is shown in **table 3.1**.

Table 3.1 The example database file `mkiv-publications.bib`. This bibliography database test file contains one entry of each type or category, with the tag set to the entry type name. This entry shown here illustrates many features explained elsewhere in the text.

```
@Article{article,
  Title                 = {Article title},
  Title:fr              = {Titre de l'article},
  Author                = {First Last}
                      # { and First Middle Last}
                      # { and Last, First-Combined}
                      # { and First van Last}
                      # { and van Last, First Middle}
                      # { and Last, Junior, First Middle}
                      # { and van Last, Junior III, First Middle}
                      # { and De La, Last, Junior III, First Middle},
  Journal               = {Journal name},
  Year                  = {YYYY},

  Note                  = {note...},
  Number                = {number},
  Pages                 = {ff--tt},
  Subtitle              = {Subtitle},
  Type                  = {Special Issue},
  Volume                = {volume},

  Month                 = {MM-ignored!},
  Day                   = {DD-ignored!},
  Abstract              = {Abstract...},
  Warning               = {Notice that Comment= also exists},
  Comment               = {Comment...},
  XCrossref             = {crossref},
  Doi                   = {doi-specification},
  Keywords              = {keyword1; keyword2; keyword3},
  Language              = {english},
  Review                = {Review...},
  Url                   = {url}
}
```

This test file was constructed to illustrate various features of the bibTeX format and contains some fields that might appear somewhat curious.

3.1 Dataset coverage

You can load much more data than you actually need. Usually only those entries that are referred to explicitly will be shown in lists; commands used to select these dataset entries will described in **section 5**. A single bibliography list can span groups of datasets; also multiple datasets can loaded from the same source, for example, one per chapter, in order to achieve a complete isolation of bibliographies with respect to numbering and references.

As this concept is not obvious but can be quite useful, we will repeat this last point: multiple datasets can be loaded from the same source, i.e. contain the same data, to be used in parallel, independently. There is little penalty in keeping even very large datasets as multiple copies in memory.

The current active dataset to be used by default can be set with

\setupbtx[dataset=example**]**

However, most publication-related commands accept optional arguments that denote the dataset and references to entries can be prefixed with a dataset identifier. More about that later.

3.2 Specification

The content of a dataset can be anything: entries of type or categories of all sorts containing arbitrary fields. The use to be made of the data can vary greatly since the system is not limited to the production of bibliography lists in particular. The intended use is reflected through a set of specifications, specific to each bibliography (or non-bibliography) style. These specifications affect the interpretation of dataset categories and fields as well as their rendering. They will also affect the rendering of citations or the invocation of individual data entries.

The `default` bibliography specification is very simple: only the categories `book` and `article` are explicitly defined. These were shown along with their default rendering in the quick-start example on **page 4**. The notion of categories and the fields that they might contain and their interpretation depend on a particular specification, although the dataset *content* is independent of all eventual rendering specifications.

An alternative set of specifications can be selected using, for example

\usebtxdefinitions[apa**]**

Alternately, the set of specifications can be loaded and (later) activated using

\loadbtxdefinitionfile[apa**]**
...
\setupbtx[specification=apa**]**

but it is safer to use the `\use` rather than `\load` form, in particular with specifications that may themselves have several variants. Also, it is way too easy to later forget to set the `specification` parameter.

We wish to insist on the fact that each specification defines the categories of entries and the interpretation or use of the fields that they contain. It also defines *setups* that control the rendering of lists as well as citations (to be described below). Additionally, it creates a namespace with settings for particular *parameters* controlling the formatting of names, for example, punctuation as well as other stylistic features. The user can tune or overload these settings as needed.

A specification need not be activated before loading a dataset; indeed the contents of a dataset are stored independent of the specification, and multiple specifications can be applied to the same dataset, although this will not usually be the case.

Multiple specification files can be loaded as they reside in separate namespaces, but only one specification can be selected at a time. We introduce these commands here in the context of datasets as the labeling of categories and of field use can change depending on the specification. Indeed, some specifications might ignore certain fields that could be used with other specifications. The details of how this is programmed will be explained in **Chapter 6**.

So a specification is both a definition of how a dataset is to be interpreted as well as stylistic tuning of how it is to be rendered.

3.3 Dataset diagnostics

You can ask for an overview of entries present in a dataset with:

\showbtxdatasetfields[example**]**

The listing that this produces is shown in **Appendix A**.

Sometimes you might want to check a database, listing all of its entries in detail. This can be particularly useful when in doubt concerning the correctness or the completeness of the data source, remembering that invalid entries and some syntax errors are simply skipped over. One way of examining the loaded dataset in detail is the following:

\showbtxdatasetcompleteness[example**]**

The diagnostic listing which can be rather long is shown in **Appendix B**.

The dataset contains many entries and each entry is assigned to a category. It must be stressed, so we repeat ourselves here, that these 'categories' can be of any sort whatsoever, the meaning of which resides in the rendering style that is chosen. The entries contain fields, and these too can be of any sort; their use also depends on the rendering style and the category in which they belong. BibTEX has conventionally defined a number of standard

categories, each making use of a number of fields considered either required, optional or ignored. However, different traditional bibTeX rendering styles can make inconsistant use of the standard categories and fields. To make matters worse, different `.bib` database handling programs might use (and impose) differing 'standards' as well, as mentioned above.[4] This situation arises from the complexity of handling bibliographic data of all sorts.

You can see all (currently known) categories and fields with:

\showbtxfields[rotation**=...]**

The result is shown **table 3.2**.

Note that other, possibly non-bibliographic use of the present dataset system might define entirely different categories and field types, possibly having nothing at all to do with the names shown here. An example of such use is given in chapter ??.

Just as a database can be much larger than needed for a document, the same is true for the fields that make up an entry; not all entry fields will be necessarily used. This idea will be developed in the next section describing the rendering of bibliography lists.

Table 3.2 `\showbtxfields[rotation=90]` The entry category and field names (and how they are used) are defined by both the rendering style as well as by the contents of the dataset. 'Required' fields are indicated in green. All unmarked fields are normally ignored in the rendering.

	article	book	booklet	conference	electronic	film	inbook	incollection	inproceedings	literal	magazine	manual	mastersthesis	misc	music	newspaper	other	patent	periodical	phdthesis	proceedings	standard	techreport	thesis	unpublished
address		*	*	*	*	*	*	*	*			*	*		*			*		*	*			*	*
album															*										
artist															*										
assignee																		*							
author	*	*	*	*	*		*	*	*		*	*	*	*	*	*		*		*		*	*	*	*
booktitle				*			*	*	*																
composer															*										
day		*									*					*		*							
dayfiled																		*							
director						*																			
doi	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
edition		*	*				*	*	*			*									*				
editor	*	*	*				*	*	*		*					*		*			*				
file	*	*	*	*	*		*	*	*		*	*	*	*	*	*	*	*	*	*			*	*	*
howpublished			*		*									*											
institution																							*	*	
journal	*										*					*									
key										*															
month		*	*	*	*		*	*	*		*	*	*	*		*		*	*	*	*		*	*	*
monthfiled																		*							
nationality																		*							
note	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
number	*	*					*	*			*	*				*		*	*		*		*		
organization				*	*				*			*						*			*	*			
pages	*	*					*	*			*										*		*		
producer						*																			
publisher		*	*			*	*	*	*			*			*			*	*		*		*		
school													*							*				*	
series		*	*				*	*	*										*		*				
subtitle	*	*	*	*	*		*	*	*		*	*	*	*		*	*	*	*	*	*	*	*	*	*
text										*															
title	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
type	*	*	*	*	*	*	*	*	*		*		*	*		*		*		*				*	*
url	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
volume	*	*					*	*			*								*				*		
year	*	*	*	*	*	*	*	*	*		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
yearfiled																		*							

4 Renderings

4.1 Generating lists of publications

A list of publications can be rendered at any place in the document, and multiple renderings can appear under certain limitations (according to scope). The specification introduced previously defines the style of the rendering as well as what data it will contain.

If you want to see what publications are in the database, the easiest way is to ask for a complete list:

```
\usebtxdataset
  [example]
  [mkiv-publications.bib]


\definebtxrendering
  [example]                  % named rendering
  [apa]                      % parent namespace
  [dataset=example,
   group=examples]           % group will be presented later


\placelistofpublications % aka \placebtxrendering
  [example]                  % rendering defined above
  [method=dataset]           % i.e. all entries
```

The commands \placelistofpublications (that is just a synonym for \placebtxrendering) refer to a named rendering and accept a list of options (inherited from \setupbtxrendering). When applied to a named dataset (other than default), a named rendering needs to be associated through \definebtxrendering as is done here.

Note that the define shown above explicitly inherits from a rendering named apa that was itself defined when loading the specification file.[1] Had this inheritance not been specified, the new rendering would inherit default, minimal settings. This notion of inheritance will be further illustrated later.

The bibliography list rendering of our example dataset, following the APA style, is:

BookAuthorLastname, F. (YYYY). *Book title*. Address: Author.

BookAuthorLastnameA, F., BookAuthorLastnameB, F.M., & BookAuthorLastnameC, F.-C. (YYYY). *Book title*. (Vol. volume, no. number, edition ed.). Address: Publisher. doi:doi-specification (note...)

BookAuthorLastname2, F. (YYYY). *Book title*. Address: Author.

BookletAuthorLastname, F. (YYYY). *Booklet title*. Address: howpublished. doi:doi -specification (note...)

ComposerLastName, F. (YYYY). *Music title*. [Recorded by FirstnameArtistLastname]. In Album title. address: label. Available from url (note...)

ConferenceAuthorLastname, F. (YYYY). Conference title. In EditorLastname, F. (Ed.). Booktitle. <Symposium conducted at the meeting of> Organization, Address: publisher. doi:doi-specification (note...)

EditorLastname, F. (Ed.). (YYYY). *Editorial title*. [Editorial]. *Newspaper name*, (number). Available from url (note...)

———— (YYYY). *Periodical title*. Available from url (note...)

———— (YYYY). *Proceedings title*. (Vol. volume, no. number). Address: Publisher. doi:doi-specification (note...)

EditorLastnameA, F. & EditorLastnameB, F. (Eds.). (YYYY). *Book title*. Address: Publisher.

ElectronicAuthorLastname, F. (YYYY). *Electronic title*. Organization. howpublished. doi:doi-specification (note...)

InBookAuthorLastname, F. (YYYY). *InBook title*. (Vol. volume, no. number, edition ed., pp. ff–tt). Address: publisher. doi:doi-specification (note...)

InCollectionAuthorLastname, F. (YYYY). InCollection title. In EditorLastname, F. (Ed.). Booktitle. (Vol. volume, no. number, edition ed., pp. ff–tt). Address: publisher. doi:doi-specification (note...)

InProceedingsAuthorLastname, F. (YYYY). InProceedings title. In EditorLastname, F. (Ed.). Booktitle. <Symposium conducted at the meeting of> Organization, Address: publisher. doi:doi-specification (note...)

Last, F., Last, F.M., Last, F.-C., van Last, F., van Last, F.M., Last Junior, F.M., ... De La Last Junior III, F.M. (YYYY). Article title: Subtitle. [Special Issue]. *Journal name*, *volume*(number), ff–tt. doi:doi-specification (note...)

———— (YYYY). Article title: Subtitle. [Special Issue]. *Journal name*, *volume*(number), ff–tt. doi:doi-specification (note...)

LastnameA, F., LastnameB, F., LastnameC, F., LastnameD, F., LastnameE, F., LastnameF, F., & LastnameG, F. (n.d.). Advanced article title. *Journal name*. Advanced online publication. Available from url

MagazineLastname, F. (YYYY). *Magazine article title*. *Magazine name*, (number). Available from url (note...)

ManualLastname, F. (YYYY). *Manual title*. (Edition ed.). Address: Organization. doi:doi-specification (note...)

MastersThesisAuthorLastname, F. (YYYY). *MastersThesis title*. (Type). school, Address. doi:doi-specification (note...)

MiscAuthorLastname, F. (YYYY). *Misc title*. howpublished. doi:doi-specification (note...)

NewspaperLastname, F. (YYYY). *Article title*. *Newspaper name*, (number). Available from url (note...)

Organization (YYYY). *Proceedings title*. (Vol. volume, no. number). Address: Publisher. doi:doi-specification (note...)

OtherAuthorLastname, F. (YYYY). *Other title*. Author. doi:doi-specification (note...)

PatentAuthorLastname, F. (YYYY). *Patent title*. *nationality Patent no. number*. Author. Available from url (note...)

PhDThesisAuthorLastname, F. (YYYY). *PhDThesis title*. (Doctoral dissertation). school. doi:doi-specification (note...)

ProducerLastName, F. (Producer), & FirstnameDirectorLastname (Director). (YYYY). *Film title*. [Motion picture]. address: studio. Available from url (note...)

Publisher (YYYY). *Book title*. Address: Author.

StandardAuthorLastname, F. (YYYY). *Standard title*. doi:doi-specification (note...)

TechReportAuthorLastname, F. (YYYY). *TechReport title*. (Type, no. number). Address: institution. doi:doi-specification (note...)

ThesisAuthorLastname, F. (YYYY). *Thesis title*. (Type of thesis). school, Address. doi:doi-specification (note...)

UnpublishedAuthorLastname, F. (YYYY). *Unpublished title*. Author. doi:doi-specification (note...)

(literal) Copied literally to the bibliography

The rendering is a list whose appearance can be tuned, as any list in ConTeXt. It is somewhat more complex to manage, though, because we can have not only many different standards but also many fields that can be set up. This means that there are several commands involved. As we saw demonstrated above, a rendering is defined and setup using the commands:

```
\definebtxrendering [...] [...] [...]
                       1     2     3
                            OPT   OPT
1   IDENTIFIER

2   IDENTIFIER

3   inherits from \setupbtxrendering
```

```
\setupbtxrendering [..¹..] [..²..]
                         OPT
1   IDENTIFIER

2   alternative  =  TEXT
    dataset      =  TEXT = default
    setups       =  TEXT
    method       =  local global none force = global
    sorttype     =  short reference dataset default = default
    criterium    =  TEXT = text
    refcommand   =  TEXT = authoryears
    numbering    =  yes cite = yes
    width        =  DIMENSION auto = auto
    distance     =  DIMENSION = 1.5\emwidth
    continue     =  yes no = no
    repeat       =  yes no = no
```

A rendering then placed using \placebtxrendering, a ConTEXt command that accepts the same arguments as for its setup. Using method=dataset, as above, one renders the entire contents of the dataset. Normally, however, one would not use this method and only a selection of entries (publication references) that are to be used in the document will be placed, whereas the dataset may very well contain many other, unused references. The method=global or method=local option can be used to specify if this use is to be global for the entire document or else local to a structure element such as a part or a chapter.

The rendering is a list and this reference list is numbered. These numbers can be displayed or not, depending on the bibliography style specification, through the use of the numbering=yes or no parameter values. The APA style, illustrated above, normally does not number the bibliography list.

The reference list is also sorted, controlled by the parameter sorttype; here sorttype=authoryear, appropriate for the APA style, sorting first by author list, then by publication year, then by title, finally by page. One can sort the list in many ways: for example sorttype=index will render the list in the order in which it was loaded into the dataset (see **table 4.1**).

As a concrete example, the rendering named example above inherits from the rendering instance named apa that is defined as:

```
\definebtxrendering
  [apa]
  [specification=apa,
   sorttype=authoryear,
   numbering=no]
```

in the specification file.

Table 4.1 `sorttype=`

`default`
`none`
`cite`
`list`
	render the list in the order in which it was built, that is, in the order that references were selected from the dataset for inclusion in the list.

`dataset` `index`	sort the list according to the dataset index, that is, in the order in that references were added to the dataset.
`reference`	sort the list in alphabetical order of the citation tags of the dataset. Note that all entries missing tags get assigned the dataset index as their tag.
`key`	sort the list in alphabetical order of the entry field `key` (traditionally used in bibTEX as an alternate sorting key). Falls back on the dataset index if no such field is present.
`short`	sort the list in alphabetical order of the short tag (first three letters of the author name or first letter of the first three authors followed by the last two digits of the year).
`authoryear`	sort the list in alphabetical order of the authors (or editors or publisher), then by publication year, then by title (or by journal and volume) and finally by page.

A more subtle characteristic of renderings is that one generally would not want a bibliography list to appear redundantly in a document, unless of course it is desired that elements of the list reappear in later lists, for example when placing partial bibliographies at the end of each chapter and a complete bibliography list at the end of a book. One must specify `repeat=yes` in order to get multiple renderings of a bibliography list; otherwise, as they appear, entries get marked as placed and will be inhibited from being placed again elsewhere.

All renderings of bibliography lists such as the one shown above also depend on a set of general list parameters that apply to each individual entry (publication), like for any list item in ConTEXt. These can be adjusted through the command \setupbtxlist (see also **section 6.2** below). As an example, the `apa` specification file includes:

```
\setupbtxlist
  [apa]
  [alternative=paragraph,
   width=fit,
   distance=.5em,
   margin=3em]
```

Such settings (yielding a hanging list that would be inappropriate with a numbered bibliography list) get inherited from the `apa` namespace in the rendering that we named `example` above.

Let's try another example (taken from the **TUG bibliography archive**):[2]

```
\definebtxdataset[template]
\usebtxdataset   [template][template2.bib]

\loadbtxdefinitionfile[aps]
\definebtxrendering
  [template]
  [aps]
  [dataset=template,
   group=examples]

\placelistofpublications
  [template]
  [method=dataset]
```

Here, the new rendering is defined to inherit explicitly from a rendering named aps (loaded, but not activated here). Notice, in particular, compared to the previous rendering example, that the list is numbered and that there is no hanging margin (it uses alternative=b).

[34] P. Chen and M.A. Harrison, Index Preparation and Processing, *Software—Practice and Experience* **19**(9), 897–915 (1988). (The LaTeX text of this paper is included in the makeindex software distribution.)

[35] D. Kahaner, C. Moler, and S. Nash, *Numerical Methods and Software*, p. xii + 495 (Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1989).

[36] R. Fletcher, *Practical Methods of Optimization*, second ed., pp. 183–188 (John Wiley & Sons, New York, 1987).

[37] S.C. Johnson and M.E. Lesk, Language Development Tools, In UNIX System Readings and Applications, pp. 245–265 (Prentice-Hall, Englewood Cliffs, NJ 07632, USA, 1987). (Reprinted from The Bell System Technical Journal, 1978)

[38] R.W. Gray, γ-GLA—A Generator for Lexical Analyzers that Programmers Can Use, In Summer USENIX '88 (1988).

[39] Adobe Systems Incorporated, *Adobe Type 1 Font Format* (Adobe Systems Incorporated, 1585 Charleston Road, P. O. Box 7900, Mountain View, CA 94039-7900, USA, Tel: (415) 961-4400, 1990).

[40] D.A. Adams, *A Dialogue of Forms: Letters and Digital Font Design*, (Master's thesis). Massachusetts Institute of Technology, MIT Microreproduction Lab, 77 Massachusetts Avenue, room 014-0551, Cambridge, MA 02139. $46.35 (1986).

[41] Abraxas Software, Inc., *PCYACC 2.0*, (1987). (7033 SW Macadam Ave., Portland, OR 97219.)

[42] Current Science, *AIDS*, (1987–date). (Current Science publishes more than 80 medical periodicals using an in-house markup language for which TEX is the back-end typesetting system. All submitted articles (whether on paper or on magnetic media) are rekeyed by clerical staff into the markup system.)

[43] A. Naiman, *The Use of Grayscale for Improved Character Presentation*, (Doctoral dissertation). University of Toronto, ATTN: Technical Reports, Computer Systems Research Institute, University of Toronto, 6 King's College Road, Toronto, Ontario M5S 1A1, CANADA, Tel: (416) 978-8751. The cost is CDN\$16 if inside Canada, US\$16 if outside. Avi's e-mail address is pathavi@cvs.rochester.edu. (1991). (Technical report CSRI-253)

[44] J. André and R. Hersch, *Raster Imaging and Digital Topography* (Cambridge University Press, 1989).

[45] P. Chen and M.A. Harrison, *Automating Index Preparation*, Technical Report, no. 87/347 (Computer Science Division, University of California, Berkeley, CA, USA, 1987). (This is an expanded version of [`<Chen:SPE-19-9-897>`].)

[46] P.M. English, *Using METAFONT for Original Font Design* (1987).

The numbering of the references are unique and pick-up from where the previous numbering stops. This behavior can be `method=global` for the entire document, or else `method=local`, limited to a structural element such as a chapter or a part. Furthermore, one can associate renderings into particular number groups, effectively isolating them from any other renderings. For example, as seen above in the rendering definitions, the present manual uses:

```
\setupbtxrendering[example] [group=examples]
\setupbtxrendering[template][group=examples]
```

setting the numbering of the group `examples` apart from the numbering of the dataset and its named rendering `default` that contains this manual's bibliography (that will be placed later, on **page 87**).

4.2 Rendering styles, or more on specifications

The default rendering style implemented in the MkII module was loosely based on the APA standard. We made a design choice in the present MkIV system to provide a very minimal `default` style and one should not expect much by default. A user requiring a more complete rendering will want to explicitly load another style file. One such complete specification, shown above and appropriately named `apa`, is described in the *Publication Manual of the American Psychological Association* [9].

A note on the APA style: We get the strong impression that the APA bibliography style standard was made with the implicit assumption that manual intervention would be involved in the editing and production process; It has been an arduous task to create a system capable of fully conforming to these specifications.

Furthermore, we note that it has sometimes been argued that a numbered citation system and bibliography list is supposedly superior to the author-year scheme, as is employed in the APA style (as well as others); indeed, it is certainly much easier from the point of view of the programmer of an automated typesetting system such as TEX. Yet many find that the longer author-year citations can be of great use to the reader (as well as the writer) so we take no stand in this debate and provide both possibilities.

We have made (and continue to make) a great effort to scrupulously respect the APA style in the rendering. Yet be warned that very few editors and publishers in fact follow this style exactly so some customization will always be required.

There are many prescribed styles to render bibliographic descriptions that can be programmed as standards. Alternatives to `specification=apa` might be `mla`, `chicago`, `harvard`, `ieee`, `aps` (commonly used in the physical sciences), `vancouver` (used in the biological sciences), or many others. At this time, we only provide two description files, with `aps` being an example[3] of a number-based rather than an authoryear-based scheme; more style schemes may be added in the future and the customization of a rendering style will be described in a later chapter.

The rendering style usually also implies a particular bibliography list sorting scheme as well as the use of a particular citation style. Indeed, the rendering of bibliography lists and references to it are intimately coupled. This question will be explained a bit later. The APA style, for example, specifies that the bibliography list be sorted by author and then by year, then by title. The list is not numbered (references are cited by author and year). Note, however, as can be seen from the two examples shown above that the references were assigned numbers even though they are not displayed in the `apa` rendering (the user can easily choose to display these numbers).

4.3 Bibliography list scope

A single dataset can by used with multiple renderings. Although they may illustrate different styles (as here for the purpose of demonstration in a manual on bibliographies), this would not be a coherent choice for a document that would normally employ a single bibliography style.

The most obvious use of multiple renderings (employing a single specification) is the placement of bibliography lists localized by structure elements: parts, chapters in a book, sections, etc. through the option `criterium=chapter`, for example. This can be setup using:

```
\setupbtxrendering
    [default]
    [repeat=yes,
     continue=yes,
     method=global]
```

followed at the end of each chapter by

```
\placelistofpublications
    [criterium=chapter]
```

The numbering might alternately be made local through the option method=local. If desired, the reference numbers can also get prefixed:

```
\setupbtxlist
    [default]
    [prefix=yes,
     width=1cm]
```

A bibliography list rendering placed at a single location, at the end of a book, for example, can also be easily split into structural parts. The following code (given without any explanation) illustrates this nicely:

```
\startbackmatter
    \startchapter [title=Bibliography,number=no,incrementnumber=no]
        \startsubject [title=Introduction]
            \placelistofpublications
                [criterium=reference,reference=Introduction]
        \stopsubject
        \dorecurse{12} {
            \startsubject [title={Chapter #1}]
                \placelistofpublications
                    [criterium=bodypart:chapter,reference=#1]
            \stopsubject
        }
    \stopchapter
\stopbackmatter
```

Note, as the above illustrates, that the list of bibliography citations for *any* referenced structural element can be placed *anywhere* in the document, not simply within the structural element.

4.4 Language

Bibliography lists (and citations in the text, see below) are rendered in the language of the document (\mainlanguage). However, a bibliography entry can contain a language= field and this will be used (if present), depending on the specification, in the rendering and hyphenation of the title, for example.

One might choose to override this behavior and impose a single language for all bibliography entries (as well as for all other ConTEXt constructs) using \setupdelimited-text[language=global].

Table 4.2 \setupdelimitedtext[language=…]

language=	
<blank>	(the default) use the current active language;
local	respect local language directives (such as the entry's language={…}, field, if present and managed by the rendering specification)
global	impose the main document language;
en, de, nl, ...	impose the specified language.

Since this directive is general for all delimited text in ConTEXt and is not specific to bibliographies, one can apply it to force a particular language within a unique section, as in:

```
\startsection[title=Bibliografía]
    \setupdelimitedtext[language=es]
    \placelistofpublications
\stopsection
```

The language setting also influences the sorting of utf strings, in particular authors and titles.

Translated titles

Going beyond this handling of punctuation, labels and sorting, we have also introduced an entirely new feature, allowing for dataset entries to contain fields describing translated titles, for example. This is particularly useful when including citations to references published in languages other than the document language.

Below is an example including *three* languages:[4]

```
\startbuffer[leibniz]
@INCOLLECTION{Leibniz1885,
  author       = {Leibniz, G. W.},
  title        = {Principes de la nature et de la grâce fondés en raison, 1714},
  title:en     = {Principles of Nature and Grace Founded in Reason},
  booktitle    = {\de Die Philosophischen Schriften von Gottfried Wilhelm Leibniz},
  booktitle:en = {The Philosophical Writings of Gottfried Wilhelm Leibniz},
  editor       = {Gerhardt, C. G.},
  publisher    = {Weidmann},
  year         = {1885},
  volume       = {6},
  chapter      = {8},
  pages        = {598–606},
  address      = {Berlin},
  language     = {french},
}
\stopbuffer
```

with

```
\usebtxdataset      [leibniz][leibniz.buffer]
\definebtxrendering[leibniz][apa][dataset=leibniz]
\placebtxrendering [leibniz][group=examples,method=dataset]
```

that gets rendered as:

> Leibniz, G.W. (1885). Principes de la nature et de la grâce fondés en raison, 1714 [Principles of Nature and Grace Founded in Reason]. In Gerhardt, C.G. (Ed.). Die Philosophischen Schriften von Gottfried Wilhelm Leibniz [The Philosophical Writings of Gottfried Wilhelm Leibniz]. (Vol. 6, p. 598–606). Berlin: Weidmann.

In the APA bibliography style, the original reference title is listed followed by the translated title (within square brackets), the translation chosen being that of the \mainlanguage of the document (if present in the entry, of course). In other bibliography styles, this feature might not be implemented or ignored.

Multilingual bibliographies

The present handling of languages is only the beginning, and these features will be developed further in the future: multilingual typesetting being one of the great strengths behind ConTEXt.

4.5 Page index

The list renderings can also include a page index of citations to each entry. This can be enabled using the parameter pagestate:

\setupbtxrendering **[**pagestate**=**start**]**

As this is more of a subject related to the citation mechanism, it will be described in the following chapter (see **section 5.6**, **p. 43**).

Footnotes

[1] A default database, default, is predefined, yet we recommend defining it explicitly because in the future we may provide more options.

[2] The fourth example shows that a named buffer can also be employed to add dataset entries (in bibTEX format). This may be useful for small additions or examples, but it is generally a good idea (for convenience of management of data) to place them in files separate from the document source code.

[3] This code snippet demonstrates that \usebtxdataset will implicitly declare an undefined dataset name, although this practice is to be discouraged. Similarly, omitting to specify the dataset name [default] in the examples given earlier would fall-back correctly, but this, too, is to be discouraged as being error-prone.

Renderings

[4] For example, `jabref`, in addition to discarding all comments contained in the database file, will convert all unrecognized, preciously named categories to `@Other`! Of course, `jabref` is flexible enough to be configured with new categories and additional fields, so users of `jabref` with ConTEXt will probably want to use an extended, custom configuration.

[1] The rendering parent `[apa]` sets the parameters: `specification=apa, sorttype=authoryear,` and `numbering=no,` described below. It also sets various list parameters in a protected `apa` namespace to produce a 'hanging' bibliography list.

[2] The file `template2.bib` is simply `template.bib` that has been cleaned-up to remove the empty entries.

[3] The APS style attempts to be very compact, providing complete yet minimal information, in contrast to the APA style which is rather verbose.

[4] In this reference, the `title` and the `booktitle` are in different languages and the `language = {french},` field refers to the cited text. Both titles would have been rendered using French hyphenation rules had this not been overridden by including the switch `\de` that otherwise should be avoided in a database. This complicated situation is not very common.

5 Citations

The APA Style Guide as well as good practice demand that *all* references appearing in the bibliography be cited at least once in the text (and, of course, all citations must have a corresponding bibliographical reference). Other publishing styles, textbooks in particular, might additionally include lists of general references for 'further reading' (and these lists might sometimes be split into sections according to subject). An author may, in contrast, choose not to interrupt a text with many citations, nevertheless, including a list of references. In contrast, one might refer in the text to certain works that need not necessarily be accompanied by a full bibliography listing (for example, Darwin's *Origin*).[1] Thus, a system providing tools to handle bibliographies needs to be flexible.

A good, general reference on bibliography practice (in the English language), independent of any particular specification, can be found in *<vanLeunen1992>*. (<vanLeunen1992>) Note that rules and traditions may differ slightly in other languages and cultures.

The examples of bibliography listing of the previous chapter were simplified by the fact that the entire bibliographical dataset was rendered. In practice, the same dataset source(s) could be used over many documents, and it might contain many more references than are used in any one document. That is, the data source might be more complete than the final rendered bibliography list or lists. The mechanism of citation allows you to select references from the dataset(s) to build the list rendering as well as to place a indicator of the reference (or not) in the text of the document corresponding to a list entry (publication). These citation renderings can be of many forms.

A citation is normally pretty short as its main purpose is to refer uniquely to a more detailed description. But, there are several ways to refer, which is why the citation subsystem is configurable and extensible. Just look at the following commands:

\cite[num**]**	**[**article**]**	[15]
\cite[textnum**]**	**[**article**]**	Ref. 15
\cite[authornum**]**	**[**article**]**	(Last, et al. [15])
\cite[authoryear**]**	**[**article**]**	(Last, et al., YYYYb)
\cite[authoryears**]**	**[**article**]**	Last, et al. (YYYYb)
\cite[short**]**	**[**article**]**	[LLL+99b]
\cite[tag**]**	**[**article**]**	[article]
\cite[index**]**	**[**article**]**	[1]
\cite[category**]**	**[**article**]**	article
\cite[author**]**	**[**article**]**	Last, et al.
\cite[year**]**	**[**article**]**	YYYY
\cite[title**]**	**[**article**]**	*Article title*
\cite[keywords**]**	**[**article**]**	keyword1; keyword2; keyword3
\cite[none**]**	**[**article**]**	
\cite[X**]**		(<X>)
\cite[]		()
\cite[template::Chen:1988:IPP**]**		(Chen and Harrison, 1988)
\cite[entry**]**	**[**article**]**	Last, F., Last, F.M., Last, F.-C., van Last, F., van Last, F.M., Last Junior, F.M., ... De La Last Junior III, F.M. (YYYY). Article title:

Subtitle. [Special Issue]. *Journal name, volume*(number), ff–tt. doi:doi-specification (note...)

```
\cite [...¹.] [...².]
          OPT
1   IDENTIFIER

2   IDENTIFIER
```

The first argument is optional and if omitted, the default citation rendering (for example, `num` or `authoryear`, depending on the specification) will be selected.[2]

\cite[article]

(Last, et al., YYYYb)

The default citation alternative is defined via

\setupbtx[default:cite]**[alternative=**num**]**
\setupbtx   **[**apa:cite]**[alternative=**authoryear**]**

The `\citation` command is synonymous to `\cite`.

> Note that the MkII module based on `bibtex` allowed the use of curly brackets enclosing the tag (for reasons of backward compatibility with traditional LaTeX practice). As a consequence, this made it a bit picky about spaces between its arguments, the first of which is optional. We have chosen to remove this restriction through the use of standard ConTeXt syntax using square brackets (reserving curly brackets to usually be used to enclose text that is to be typeset). However, although the practice is to be strongly discouraged, the system will tolerate the depreciated syntax `\cite{tag}`, but this cannot be mixed with any other options (see as follows).

The most commonly used citation variants (or alternatives) are `num` and `authoryear` introduced above. The first is typically employed in conjunction with a numbered bibliography list, usually sorted in the citation order in the text; The second is typically used in conjunction with a bibliography list sorted by author and year of publication.

Other citation variants may be quite useful, even when used in the context of the above standard schemes. One such example would be `\cite[title][tag]` that can be used to include the title of the work in the running text (as can be seen earlier in **section 4.1**; another one is `\cite[year][tag]` that can be used to include the publication date or `\cite[author][tag]` that can be used to extract the authors' names. These are examples of a general rule where `\cite[field][tag]` will return the contents of `field` for an entry, if it contains such a field.

The variants `textnum` and `authoryears`[3] are intended to be used typically in the running text when the reference becomes part of the syntax of the sentence.

Notice that in the next to last example shown above, the `tag` was prefixed with an alternate dataset name (`template::`) rather than to the default `dataset=example` that was specified in the `\setupbtx` command earlier. The reason behind the double-colon syntax should be made obvious through this example (where the tag itself uses single colons).

In the last example shown above, the full rendered bibliographic entry gets placed.

Unless the placed rendering uses `method=dataset`, only publications that are explicitly cited will end up in the lists. You can force a citation into a list using `\usecitation`, for example:

\usecitation[patent**]**

This command has two synonyms: `\nocite` and `\nocitation` so you can choose whatever fits you best.

```
\nocite [...]
                *

*    IDENTIFIER
```

The purpose and utility of these commands (and their synonyms) is not only to draw a citation from the dataset for inclusion in the bibliography, but also to mark the place in the text where the citation is relevant. Normally, one might claim that this should be done through one of the forms of the `\cite` (`\citation`) command, as all references appearing in the bibliography are to be cited at least once in the text. However, one might wish attach the citation to the reference in the text of a floating object such as a table or a figure, thus establishing a proper order in the numbering since the explicit citation rendering might occur within the table or the figure caption that might get placed on a much later page. Consider the following schematic illustration:

(see **\nocite [**MyReference**]\in {**table**} [**tab:MyTable**]).**

\startplacetable[reference=tab:MyTable**]**
    \unknown\ \cite[MyReference**]**
\stopplacetable

The citation rendering and the bibliographic list rendering are intimately coupled reciprocally and cannot be dissociated. This coupling can be through the reference number for example, but unnumbered reference lists also contain interacting hyperlinks. A failure to take into account this interdependence can lead to fundamental misunderstandings in use.

Both the citation and the list must be rendered. For example, a common error would be to omit (or comment-out) the list rendering during the writing stage of a document. Such an error will cause the citations to fail to render properly.

Whereas this might seem to be an unnecessary limitation, it results from a specific design choice and from the possibility of placing multiple renderings freely, anywhere in a document. It is preferable not to render citations at all than to render these citations possibly incorrectly.

5.1 Combining citations

A single citation might refer to several sources, obtained through the use of a comma separated list of tags, for example: [2,4,15]

`\cite[num][article,book,booklet]`

A comma separated list of three (or more) consecutive numbers will get collapsed or compressed into a range of numbers.

The order in which the citation tags appear in the list may or may not be important, depending upon the citation variant and on the style specification. Consider the following examples:

```
\cite[num]        [article,book]
\cite[textnum]    [article,book]
\cite[authoryear] [article,book]
\cite[authoryears][article,book]
\cite[short]      [article,book]
\cite[author]     [article,book]
\cite[year]       [article,book]
\cite[title]      [article,book]
```

[2,15]
Ref. 2 and 15
(BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC, YYYY; Last, et al., YYYYb)
BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC (YYYY); Last, et al. (YYYYb)
[BBB99; LLL+99b]
BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC; Last, et al.
YYYY and YYYY
Article title and *Book title*

```
\cite[num]        [book,article]
\cite[textnum]    [book,article]
\cite[authoryear] [book,article]
\cite[authoryears][book,article]
\cite[short]      [book,article]
\cite[author]     [book,article]
\cite[year]       [book,article]
\cite[title]      [book,article]
```

[2,15]

Ref. 2 and 15

(BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC, YYYY; Last, et al., YYYYb)

BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC (YYYY); Last, et al. (YYYYb)

[BBB99; LLL+99b]

BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC; Last, et al. YYYY and YYYY

Book title and *Article title*

Note that the numbered citation reference is always rendered in numerical order, where the numbers correspond to the order in which the entries appear in the bibliography list. This order depends on the sorting of the list rendering (that may be in some styles in the order in which the references are cited) and is controlled by the [list] sorttype parameter. In the APA style, presently active, the citations are always sorted according to author and year. Thus, "BookAuthorLastname" appears before "Last" in our example here, regardless of the order in which the references appear in the \cite command ([article,book] or [book,article]).

The user can control the state of sorting of the various \cite variants through the [specification:cite:variant] parameter sorttype=normal; other choices are reverse and none.

Of course, a user can always obtain any desired order through individual calls to the \cite command.

Single list item containing multiple publication entries

Some bibliography styles admit the combination of several bibliographical sources into a single list item having a unique reference number.[4] To achieve this, one combines tags using the addition operator symbol (+), best illustrated though an example:

```
\METAFUN\ began as an expression of love and appreciation for \METAPOST.
\citation [num] [tugboat::Berdnikov:TB21-2-129+Hobby:TB21-2-131]
```

```
\definebtxrendering[tugboat][aps][dataset=tugboat,group=examples]
\placebtxrendering [tugboat][criterium=section]
```

MetaFun began as an expression of love and appreciation for MetaPost. [48]

> [48]   A. Berdnikov, H. Hagen, T. Hoekwater, and B. Jackowski, Even more MetaFun with METAPOST: A request for permission, *TUGboat* **21**(2), 129–130 (2000).; J.D. Hobby, Extending METAPOST: Response to "Even more MetaFun", *TUGboat* **21**(2), 131–132 (2000).

Combined entries are joined using a separator that can be specified, as in:

```
\setupbtxrendering[tugboat][separator={\emdash}]
```

or suppressed (using separator=,); By default separator={;\space}.

Dataset entries that are combined cannot also appear apart (nor does this really make any logical sense). All of the following:

```
\cite[num][tugboat::Berdnikov:TB21-2-129]
\cite[num][tugboat::Berdnikov:TB21-2-129,Hobby:TB21-2-131]
\cite[num][tugboat::Hobby:TB21-2-131]
```

[48] [48] [48]

will point to the combined reference (as they should). However, beware that attempting to include any of these references in a *different* combined list item is undefined, meaning unsupported.

Combining list entries is another instance showing that citations and the rendered bibliography list interact and cannot be separated.

5.2 Additional text

Additional text in a citation reference

Sometimes one would like to include additional text in a citation, for example a specific commentary or page number reference.

```
\cite[righttext={ p.\nbsp xx}][article]
```

yielding:

(Last, et al., YYYYb p. xx)

The additional text can be either before (lefttext=) or after (righttext=)[5] each citation entry and are not to be confused with the delimiters left={(}, and right={)}, used to

surround the entire citation. The difference becomes important when referencing multiple citations.

The following examples further illustrate the syntax:

\cite[lefttext=**{**See **}**,righttext=**{** p.**\nbsp** yy**}**]**[**book**]**

(See BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC, YYYY p. yy)

\cite[alternative=num,righttext=**{{** p.**\nbsp** xx**}**,**{** p.**\nbsp** yy**}}**]
    [article,book,booklet**]**

[2 p. xx,4 p. yy,15]

\cite[lefttext=**{**In the article: **}**,righttext=**{**.**}**]**[**article**]**

(In the article: Last, et al., YYYYb.)

\cite[lefttext=**{{**In the article: **}**,**{** or in the book: **}}**,alternative=title**]**
    [article,book**]**

In the article: Article title and or in the book: Book title

\cite[righttext=**{{** p.**\nbsp** xx**}**,**}**]**[**article,book**]**

(BookAuthorLastnameA, BookAuthorLastnameB, and BookAuthorLastnameC, YYYY p. xx; Last, et al., YYYYb)

Note that since ConTEXt does not allow mixing key-value pair lists with single value keys, the keyword alternative= must be used, if needed, as shown in the examples above.

Additional text in a list entry

Additional text such as notes and page numbers can also get placed with the entry in the bibliography list. Of course, the bibliography data entry can contain a note= field that may or may not get rendered, according to the style specification.

It is also possible to specify notes or page references to be rendered before or after a bibliography entry through the citation call.

\cite
  [before=**{{**Introducing MetaPost: **}**,**}**,
   after=**{**,**{** See, also, the references therein, p.**\nbsp** 58.**}}**]**
  [tugboat::Hobby:TB10**-4-**505,Hobby:TB22**-1-**46**]**
\blank
\placebtxrendering **[**tugboat**][**criterium=section**]**

(Hobby, 1989; Hobby, 2001)

[49]   Introducing MetaPost: J.D. Hobby, A <u>MF</u>-like system with PostScript output, *TUGboat* **10**(4), 505–512 (1989).

[50]   J.D. Hobby, Drawing graphs with MetaPost, *TUGboat* **22**(1/2), 46–58 (2001). See, also, the references therein, p. 58.

Clearly, such additional text can be added to each entry only once, so the first such `\cite` call wins.

5.3 Placing a single citation

Sometimes, one would like to place a single citation somewhere in the text without necessarily adding it to a bibliography list. Take, for example,

obtained by using

`\placecitation[tugboat::Mahajan:TB31-1-88]`

Note that `\placecitation[tag]` is a synonym for `\citation[entry][tag]` (that was seen earlier).

As for other citation reference, this will fail if a bibliography list rendering is not placed somewhere in the document so let's do that here:

`%\placebtxrendering[tugboat][criterium=section]`

Whereas this might seem redundant above (placing the citation entry as well as its list rendering), this will only exceptionally be the case (as in this highly artificial example of this manual); Indeed, all cited work (such as Ref. , above) will logically always be placed in a list of references.

The placement of a single citation brings to light a subtle point: The `specification` that is used by citations is that set in the root namespace (through `\setupbtx[specification=xxx]` or possibly through `\usebtxdefinitions[xxx]`. This is *not* necessarily the same as that of the rendering (if a different rendering specification is declared).[6] Thus, the style of the citations can be made to differ from the style of their bibliography list, as here, but this is not a very good practice.

5.4 Placing citations as footnotes

A `\placecitation` command can occur inside a footnote or other floating object such as a figure or table caption. No specification or style that places its bibliography list renderings as footnotes has been implemented yet.

5.5 Searching

Finding the right tag in a database can be a pain. In particular, datasets having their origin in multiple source files may contain conflicting tags, though duplicate tags get suffixed automatically so this should not be a real problem.

On the other hand, asking for a wildcard also makes no sense.[7] Nevertheless, we provide a powerful mechanism for matching a query. Keep in mind that a tag is used as a quick lookup in a hashed table whereas a search will look through the entire dataset. If processing speed is critical, one should use the cite tag lookup; in practice, even on a big book project employing a very large dataset, the search is not a penalty.

Let's look in more detail at the `\cite` command. In order to distinguish efficiently between a normal reference and a more clever one, we use the `match()` function:

```
match(query)
dataset::match(query)
dataset :: match ( query )
```

The handler is rather tolerant for spaces (as well as capitalization) which is handy if you have long queries that wrap around in the source code. Of course the `dataset::` prefix is optional in which case the current dataset is taken. Such match queries can be mixed in a multiple reference citation indifferently with hashed cite tags so the system is really flexible.

To demonstrate the use a search query, we load a small bibliographic database:

```
\usebtxdataset[boekplan][boekplan.bib]
```

We could switch to this base using:

```
\setupbtx[dataset=boekplan]
```

but instead we shall use a prefix as seen previously. Consider the following:

```
everything that you might want to know about layouts
\cite[boekplan::match(author:Egger)]
```

We will get: "everything that you might want to know about layouts (Hoekwater, Hagen, and Egger, 2011)". Of course this assumes that we also typeset a list of references somewhere in our document, so let's do that here:

```
\definebtxrendering[boekplan][apa][dataset=boekplan,group=examples]
\placebtxrendering [boekplan][criterium=chapter]
```

Hagen, H. (2010). *MetaFun*. (2nd ed.). Hassalt, NL: Boekplan. Available from http://www
.h2o-books.com/catalog/context/metafun

———— (2010). *MKII - MKIV, the history of LuaTeX*. Hassalt, NL: Boekplan. Available from
http://www.h2o-books.com/catalog/context/mkii-mkiv

Hagen, H. & Braslau, A. (2015). *Bibliographies: The ConTEXt way*. Hassalt, NL: PRAGMA ADE & ConTEXt Development Team. (still incomplete)

Hoekwater, T. & Hagen, H. (2011). *Fonts in ConTeXt*. Hassalt, NL: Boekplan. Available from http://www.h2o-books.com/catalog/context/fonts

Hoekwater, T., Hagen, H., & Egger, W. (2011). *Layouts in ConTeXt*. Hassalt, NL: Boekplan. Available from http://www.h2o-books.com/catalog/context/layouts

```
\cite[authoryears][          match(title:Lua)]\\
\cite[authoryears][default:: match(title:Lua)]\\
\cite[authoryears][boekplan::match(title:Lua)]
```

```
< match(title:Lua)>
```
Ierusalimschy (2006)
Hagen (2010)

Notice that no match is found in the 'current' dataset (`example`).

A query eventually becomes a Lua expression so you can use helpers to achieve your goal. As a convenience there are some shortcuts to access fields. The following examples demonstrate this:

```
\cite
  [authoryears]
  [boekplan::match(author:hagen and year:2010-2011)]\\
\cite
  [authoryears]
  [boekplan::match(author:{Willi Egger})]\\
\cite
  [authoryears]
  [boekplan::match(author:Hans and (tonumber(field:year) or 0) > 2011)]
```

```
<boekplan::match(author:hagen and year:2010-2011)>
```
Hoekwater, Hagen, and Egger (2011)
```
<boekplan::match(author:Hans and (tonumber(field:year) or 0) > 2011)>
```

You can use grouping braces when spaces are involved. Ranges (of numbers) are recognized. Of course, you can use other characters that the basic alphabet.

```
\cite[authoryears][tugboat::match(author:{Bogusław Jackowski})]\\
```

Ryćko and Jackowski (1993); Jackowski (1999); Berdnikov, Hagen, Hoekwater, and Jackowski (2000); Jackowski and Nowacki (2003); Jackowski, Pianowski, and Strzelczyk (1998); Nowacki and Jackowski (1998); Jackowski, Pianowski, and Strzelczyk (1998); Jackowski (1995); Jackowski, Nowacki, and Strzelczyk (2003)

This demonstrates further the interest in converting classical TEX accent sequences into proper utf characters. The citations found above **must** correspond to some bibliography list entries, so we will place this list here:

\placebtxrendering [tugboat**][**criterium**=**section**]**

[56]   M. Ryćko and B. Jackowski, TEX from **indent** to **par**, *TUGboat* **14**(3), 171–176 (1993).

[57]   B. Jackowski, M̲F̲: Practical and impractical applications, *TUGboat* **20**(2), 104–119 (1999).

[58]   B. Jackowski and J.M. Nowacki, Latin Modern: Enhancing Computer Modern with accents, accents, accents, *TUGboat* **24**(1), 64–74 (2003).

[59]   B. Jackowski, P. Pianowski, and P. Strzelczyk, More TEX-PostScript links, *TUGboat* **19**(3), 272–275 (1998).

[60]   J.M. Nowacki and B. Jackowski, Antykwa Toruńska: An electronic replica of a Polish traditional type, *TUGboat* **19**(3), 242–243 (1998).

[61]   B. Jackowski, P. Pianowski, and P. Strzelczyk, Threshing EPS files, *TUGboat* **19**(3), 267–271 (1998).

[62]   B. Jackowski, A M̲F̲–EPS interface, *TUGboat* **16**(4), 388–395 (1995).

[63]   B. Jackowski, J. Nowacki, and P. Strzelczyk, Programming PostScript Type 1 fonts using MetaType1: Auditing, enhancing, creating, *TUGboat* **24**(3), 575–581 (2003).

String lookups are partial and case insensitive.

\cite [boekplan**::**match**(**author**:**e**)]**
\cite [boekplan**::**match**(**author**:**h**)]**

so one must take care in formulating cite queries as both lookups above will get all five entries: (Hagen, 2010; Hoekwater and Hagen, 2011; Hagen and Braslau, 2015; Hagen, 2010; Hoekwater, Hagen, and Egger, 2011)
(Hagen, 2010; Hoekwater and Hagen, 2011; Hagen and Braslau, 2015; Hagen, 2010; Hoekwater, Hagen, and Egger, 2011), whereas

\cite [boekplan**::**match**(**author**:**eg**)]**
\cite [boekplan**::**match**(**author**:**e*r**)]**

only finds (Hoekwater, Hagen, and Egger, 2011)
(Hagen and Braslau, 2015; Hoekwater and Hagen, 2011; Hoekwater, Hagen, and Egger, 2011).

Note also that the order of the match criteria is not significant.

The search mechanism is very powerful. However, a **major pitfall** or risk comes from the possibility to easily under-specify the match criteria. For example, `match(author:Hagen and author:Hoekwater)` will find *Fonts in ConTeXt*, but it will also find *Layouts in Con-TeXt*; Nor would adding the criterium `and year:2011` be of any help. The solution is

```
\cite
  [authoryear]
  [boekplan::match(author:Hagen and author:Hoekwater and title:fonts)]
```

(Hoekwater and Hagen, 2011)

Note that using `match(title:fonts)` alone would also work, that is until another reference work containing the word 'fonts' in its title gets added to the dataset![8] Thus, whereas it might appear that the match mechanism be a more robust way of identifying dataset entries then short cite tags, incomplete search queries might return unwanted, excess matches.

It is quite common in real bibliography databases for some author or authors to appear in many different references published in the same year, perhaps buried in a longer list of authors, so some care has to be taken to uniquely identify the desired work. Of course, this feature can also be a good shorthand as well to select several different matching works when that is the desired result.

5.6 Page index and interaction

Each citation in the text not only marks the dataset entry for inclusion in the bibliography list but also records the page number on which the citation occurs. Each named list rendering can be instructed to include these pages just like an index.

```
\setupbtxrendering
    [default]
    [pagestate=start]
```

The result can be observed in the "**Bibliography**" on **page 87**. If interaction is active (`\setupinteraction[state=start]`), then these page numbers will be hyperlinks back to the citations. If `numbering=yes`, then the numbered bibliography entries will also contain hyperlinks back to the first occurrence in the text where the entry is cited (which is the same as the first page indexed).

phasisThis is no longer working, it seems!

Some styles, such as `apa`*, will have other hyperlinks. The author list including the year will be active just like the numbers above (an* `authoryear` *list is usually not numbered). Furthermore, any doi or url that is included in the entry will be a hyperlink to the external resource. Finally, all titles will contain hyperlinks to the local downloaded file of the cited work if this exists and was specified (using the data field* `file={},`*).*

In addition to the index of pages of citations in the text associated with each entry in the bibliography list, one has the possibility of adding items to any standard index list. This can be illustrated through the creation of a list of authors or names:

```
\defineregister
  [indexofauthors]
```

```
\definebtxregister
  [authors]
  [field=author,
   register=indexofauthors,
   method=always,
   dataset=default,
   alternative=invertedshort]
```

placed in the preamble of the document and to be followed later, typically towards the end of the document, by

```
\startchapter[title=Index of Names]
    \placeregister[indexofauthors][compress=yes]
\stopchapter
```

as can be see in the present manual on **page 89**.

Any field can be indexed, as desired, for example `field=title`. One particularly useful field is `keywords={}`: a semi-colon separated list of keywords will be split into individual index entries for each cited work.[9]

5.7 Summary and explanation of the \cite mechanism

As the list of cited references is being built in memory, each used dataset entry is flagged. As the list gets placed, either partially or fully, eventual conflits in `short` and in `authoryear` tags are resolved through an assignment of suffixes. These are then fed-back to the citation references.

A dataset can have multiple renderings and multiple datasets and renderings can be grouped. The many citation variants can be freely used together and this may or may not make coherent sense.[10] With such flexibility comes complication.

Footnotes

[1] Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life*. London: John Murray.

[2] The cite variant num refers to a counter that is usually sequential in the order in which citations are called in the running text, with the rendered bibliography list sorted in the same order. It is not to be confused with number which corresponds to the bibliographic data field that is also available.

[3] The name `authoryears` (with an 's') is a shorthand that was used in MkII. Whereas its name is not immediately obvious at first view, in practice it is a quite convenient variant of `authoryear` that differs only in the style of punctuation, thus its place in a sentence structure.

[4] The combination of multiple bibliographic entries into as single bibliography list item is more compact and this practice is often encountered in short 'letter' type journal articles. Entries combined as such do not make any sense in an authoryear scheme such as APA.

[5] The MkII bib module used the keyword `extra=` in the place of `righttext=`. We chose not to support this as a synonym as the name is ambiguous. Furthermore, we seek to limit the number of keywords used in ConTeXt.

[6] Note that the named rendering used above (`[tugboat]`) was not declared a child of the rendering named `[apa]`, though the `specification=apa` is active. The rendering is thus numbered, but otherwise follows the apa style.

[7] Note that `\nocite{*}` is a valid LaTeX/bibTeX practice that is used to include all entries of a `.bib` file in the final bibliography. This result can be obtained in ConTeXt through the `method=dataset` list rendering parameter. Alternately, one can use the syntax `\cite[match(*:*)]` which is a shorthand to match all values of all fields.

[8] Multiple hits would have occurred in the examples above containing `title:lua` had only one dataset been used in the typesetting of the present manual.

[9] The handling of fields to be interpreted as names (as in `author`) or as keywords (split into fields separated by semi-colons), etc. depends on the specification and is declared in an associated lua file. This will be described in the next chapter.

[10] An `authoryear` sorted list is assigned numbers that are available for use as numbered citation references, but these numbers will follow the order of the list and not the order of citations. Conversely, a list that is sorted in the order of citations will still have `authoryear` and `short` citation tags, but these will be of less use for the reader in this case than the `authornum` variant, for example.

6 Custom renderings

The rendering of citations and bibliography lists is highly configurable and custom rendering schemes can be added. The details can get quite complicated so we will begin with a description of how citation variations can be used in the running text, followed by a description on how to control the rending of the associated bibliography list.

Loading the specification file defines a namespace, say `apa`, containing subspaces `apa:list` and `apa:cite` as well as variants, for example `apa:cite:authoryear`, allowing parameters to be tuned independently for each form. This insures independence but can get somewhat confusing. Most users will not encounter any problems as all of this setup work is taken care of in the specification file; only those needing to fine-tune specific settings need to better understand the underlying mechanism.

The rendering of the variants can be adjusted through `\setupbtx` in the appropriate namespace, for example modifying (or suppressing) parenthesis or activating or deactivating interaction hyperlinks. Beyond the settings of parameters in the appropriate namespace, setups are used to perform the actual renderings of citations as well as of lists. One must be aware of namespace isolation and inheritance as well as fall-back when attempting to make changes.

A common setup will enable or disable interaction:

`\setupbtx [interaction=start]`

Notice that this is applied to the root namespace.

Other root-namespace parameters that have been used are `specification`, `dataset` and `numbering`.

Associated with `specification` is an additional parameter `default=` that defines a fall-back specification to be used to complete a derived specification.[1]

*Further 'low-level' formatting parameters, mostly dealing with the formatting of names, are also defined by default in the root namespace. These are listed in **table 6.1**. Some might eventually be redefined by the specification in other namespaces, and may with good reason differ by context: between the [cite] and [list] namespaces, for example.*

In order to insure the independence of specifications, each defines its own root namespace, for example [default], [apa] or [aps], inheriting its settings from root, upon which are built the other namespaces.

Table 6.1 \setupbtx low-level formatting parameters. Some may be redefined differently in the cite and list namespaces.

specification	*=default*	*default style*
default	*=*	*fall-back style*
interaction	*=start*	*active hyperlinks*
separator:firstnames	*=\btxspace*	*between first names*
separator:juniors	*=\btxspace*	*before 'juniors'*
separator:vons	*=\btxspace*	*before 'vons'*
separator:initials	*=\btxspace*	*between initials*
stopper:initials	*=.*	*initialization truncation symbol*
separator:invertedinitials	*=\btxcomma*	*before initials, inverted form*
separator:invertedfirstnames	*=\btxcomma*	*before first names, inverted form*
authorconversion	*=normal*	*normal, normalshort, inverted, invertedshort, name (see **table 6.3**)*
monthconversion	*=number*	*month, month:mnen (see \convertnumber)*
journalconversion	*=normal*	*normal, abbreviated(short?)*
etallimit	*=3*	*length of author list*
etaldisplay	*=3*	*length of truncated author list*
otherstext	*=et al.*	*author list truncation text*
separator:names:2	*=\btxcomma*	*separates multiple names*
separator:names:3	*=\btxcomma*	*before last name in a list*
separator:names:4	*=\btxcomma*	*between only two names*
separator:2	*=\btxsemicolon*	*separates multiple objects*
separator:3	*=\btxsemicolon*	*before last object in a list*
separator:4	*=\btxsemicolon*	*between only two objects*

6.1 Custom citation renderings

Parameters

Citation variants are mostly controlled though adjustment of their parameters within the appropriate namespace. For example, the cite variant \cite[num] would be addressed through

\setupbtx*[default:cite:num]*

for the default specification, or alternately, through

\setupbtx*[apa:cite:num]*

for the apa specification. For example, superscript numbered citations in the aps specification could be obtained through

\setupbtx*[aps:cite:num][command=***\high***]*

*Typical parameters (in addition to those listed in **table 6.1**) are presented in **table 6.2**.*

Table 6.2 \setupbtx[cite] parameters

alternative	=num	default cite form
left	=	opening string
right	=	closing string
inbetween	=\btxspace	
range	=\endash	
command	=	
style	=	
sorttype	=normal	
compress	=yes	

A demonstration of how these parameters can be manipulated in an individual \cite *call also illustrates the different ways of formatting names.*

Table 6.3 authorconversion

ac	\cite[alternative=author,etallimit=,authorconversion=ac][article]
name	*Last, Last, Last, van Last, van Last, Last, van Last, and De La Last*
normal	*First Last, First Middle Last, First-Combined Last, First van Last, First Middle van Last, First Middle Last Junior, First Middle van Last Junior III, and First Middle De La Last Junior III*
normalshort	*F. Last, F.M. Last, F.-C. Last, F. van Last, F.M. van Last, F.M. Last Junior, F.M. van Last Junior III, and F.M. De La Last Junior III*
inverted	*Last, First, Last, First Middle, Last, First-Combined, Last, First van, Last, First Middle van, Last Junior, First Middle, van Last Junior III, First Middle, and Last Junior III, First Middle De La*
invertedshort	*Last, F., Last, F.M., Last, F.-C., van Last, F., van Last, F.M., Last Junior, F.M., van Last Junior III, F.M., and De La Last Junior III, F.M.*

Setups

The next step in the customization of citation variants is through the overlaying of setups that handle the actual rendering of the citation. These rarely will need to be changed as most tuning can be done through the adjustment of parameters such as those given above.features, such as in the apa *specification where a missing date is replaced by the notation "n..d.".*

It is to be pointed-out that original citation variants can be easily added. As an example, imagine that one might like to access the field abstract *that is normally not rendered (like many other unused dataset fields). One would start by defining a parameter namespace inheriting from the* cite *namespace, followed by a simple setup (remember that the* apa *specification is currently active):*

```
\definebtx[apa:cite:abstract][apa:cite]
\startsetups btx:apa:cite:abstract
    \btxcitereference
```

```
    \btxflush{abstract}
\stopsetups

\startnarrower
\cite[abstract][boekplan::Hagen2010metafun]
\stopnarrower
```

If no special manipulation is known, the field with the same name (if found) will be simply flushed. This will only work, however, if the field is identified as either required or optional, that is not flagged as ignored in **table 3.2** *(see also* **appendix A**).

But don't expect too much support for such low level rendering control.

6.2 Custom list renderings

The rendering of lists is much more flexible and configurable than the rendering of citation markers. This is because the nature of data to be rendered requires many tools and helpers to deal with all of the eventual contingencies inherent in describing bibliographical references.

The same 'low-level' formatting parameters used for citations also apply for list *namespaces, though their settings may differ from those of the citations. Consider the* apa *style that specifies the use of "et al." in citations but "..." in the bibliography list, "and" in citations but "&" in lists, last names only in* authoryear *citations, etc.*

Bibliographies as lists

At another level of detail, the bibliography list is rendered in a standard ConT_EXt list environment that can be setup using the command \setupbtxlist *(which is only* \setuplist *working in the protected bibliography environment:* btx*). The root settings, appropriate for a numbered bibliography list, are:*

```
\setuplist
  [btx]
  [prefixstopper=:,
   state=start,
   alternative=a,
   before=\blank,
   after=\blank]
```

whereas an unnumbered, author-year sorted list might have:

```
\setupbtxlist
  [apa]
```

```
  [alternative=paragraph,
   margin=3\emwidth]
```

as already seen in **section 4**. *The above show that* \setupbtxlist()[name] *is a synonym for* \setuplist[btx:name].

```
\setupbtxlist [..*..]

*   alternative  = TEXT = \namedbtxlistparameter {alternative}
    style        = TEXT = \namedbtxlistparameter {style}
    color        = TEXT = \namedbtxlistparameter {color}
    headstyle    = TEXT = \namedbtxlistparameter {headstyle}
    headcolor    = TEXT = \namedbtxlistparameter {headcolor}
    width        = DIMENSION = \namedbtxlistparameter {width}
    distance     = DIMENSION = \namedbtxlistparameter {distance}
    hang         = NUMBER = \namedbtxlistparameter {hang}
    align        = = \namedbtxlistparameter {align}
    headalign    = = \namedbtxlistparameter {headalign}
    margin       = cd:yes cd:no = \namedbtxlistparameter {margin}
    before       = COMMAND = \namedbtxlistparameter {before}
    inbetween    = COMMAND = \namedbtxlistparameter {inbetween}
    after        = COMMAND = \namedbtxlistparameter {after}
    display      = cd:yes cd:no = \namedbtxlistparameter {display}
    command      = COMMAND = \namedbtxlistparameter {command}
```

*Each specification will have its own list namespace (*btx:specification*) that inherits from the root* btx *namespace. This model of inheritance holds true for the citation and list details as described earlier.*

Setups

The layout of the information presented in the list is entirely controlled through setups, with the help of some underlying Lua code organizing the data stored in the dataset. These setups rely on further setups as well as some convenient helpers or defined special commands. To understand this, consider as an illustration the setups defined for the default *specification (taken from the source file* publ-imp-default.mkvi*):*

```
\startsetups btx:default:list:book
    \texdefinition{btx:default:author}
    \texdefinition{btx:default:title}
    \texdefinition{btx:default:editionset}
    \texdefinition{btx:default:publisher}
    \texdefinition{btx:default:year}
    \btxperiod
\stopsetups

\startsetups btx:default:list:article
    \texdefinition{btx:default:author}
```

```
    \texdefinition{btx:default:title}
    \texdefinition{btx:default:journal}
    \texdefinition{btx:default:year}
    \btxperiod
\stopsetups
```

This specification could be extended to handle publication categories other than book and article simply by defining an additional setup, almost (see the following).

LUA tables

The qualification above is to bring attention to an important Lua component that is defined in a companion source file: publ-imp-default.lua. This file defines a hierarchical Lua table containing an element categories that itself contains the entries book, article, etc. The entries for each category themselves contain the entries required and optional listing the dataset fields that are to be used. Any field that is not declared either required or optional will be ignored.[2] Note that if a category is not declared in this Lua table, than all of its fields will be considered 'optional'. Thus, ignoring this level of control and simply defining additional setups might be sufficient for most use.

The advantage of the Lua table is a great simplification of the logic of the helper setups. Fields that might be irrelevant for one category yet used in another can be tested; if the field is to be 'ignored', than a fetch will return nothing. Indeed, the above two setups could be replaced by a single setup as journal is irrelevant and thus ignored for the category book but not for the category journal. However, an oversimplification such as just described would be rather confusing and would not extend well to a more complete specification (such as apa) handling many different categories.

Note, as well, that fields 'ignored' for certain categories in the bibliography list will also be ignored in citations. For example, choosing to ignore title in some category, say article, will cause \cite[title] to fail (return nothing) when referring to an entry of that category. In the example shown in **section 6.1**, the abstract field would not return anything if it did not appear in the Lua table.

This Lua table also defines (as 'types', outside of 'categories') what fields are to be interpreted as names, what fields are to eventually be interpreted as a number or range of numbers, and what fields are to be interpreted specially as a (semi-colon) separated list (for example, keywords).

A more subtle feature of the Lua level is the notion of 'sets'. Related fields can be grouped into sets, for example.

```
book = {
    sets = {
        author    = { "author", "editor", },
        editionset = { "edition", "volume", "number" },
```

```
    },
},
```

A fetch of the set author (i.e. \btxflush{author}, to be described below) will return the author field if this is present, otherwise it will return the editor field, for an edited book, for example. Testing for the presence of a set in a data entry is equivalent to a logical or in testing for the presence of each element of the set. This mechanism can appear confusing yet it has served to greatly simplify the logic of the various setups.[3]

The author set is a bit special given its use in the authoryear(s), authornum, and short citation variants. This set determines what fields are to enter into the citation tags, a generalization of the notion of author. For example, in the APA style, this set would include the author, editor, or title fields or an article and the author, editor, publisher, or title fields for a book and still other sequences of fields for other categories of publications.

Defined helpers

Fetching data from the dataset is performed using the command encountered above: \btxflush. If the field (or no elements of the set) is not found, then this command will return nothing. Alternately, one might need to explicitly test for the presence or absence of the field, in order to conditionally include punctuation or not to trigger separators, for example. Three test macros have been defined:

Table 6.4 Conditional macros

```
\btxdoifelse{fieldname}{action when found}{action when not found}
\btxdoif    {fieldname}{action when found}
\btxdoifnot {fieldname}                    {action when not found}
```

In many cases, the readability can be improved by using further setups, for instance:

```
\btxdoifelse {author} {
    \fastsetup{btx:apa:author:yes}
} {
    \fastsetup{btx:apa:author:nop}
}
```

Note that the choice between using setups (defined through \startsetups ... \ and recalled through \fastsetup) versus textdefinitions without arguments (defined through \starttexdefinition ... \stoptexdefinition and recalled through \texdefinition) is a question of taste and opportunity. One should keep in mind not get carried away abusing setups and texdefinitions for simple code fragments that are to be used uniquely.

An extra conditional is available for testing interactivity:

```
\btxdoifelseinteraction{action when true}{action when false}
```

There is also a conditional `\btxinteractive` which is more efficient, although in practice efficiency is not so important here.

In addition to sets, there are derived or special fields such as num (for the reference number), `suffix` (to be appended in some cases to `year`), `short` (for names), etc. These all can be retrieved using `\btxflush`. Sometimes one might want to force access to a particular data field (such as `author`) rather than eventually an element of a set having the same name. There are three basic commands to flush data and a few others to flush associated information:

Table 6.5 Field access macros

`\btxflush`	*fetch a derived or explicit field*
`\btxdetail`	*fetch a derived field (e.g. short)*
`\btxfield`	*fetch a explicit field (e.g. year)*
`\btxfieldname`	*fetch the field name*
`\btxfieldtype`	*fetch the field type (e.g. author, range, . . .*
`\btxfoundname`	*fetch the field name of a set (or field)*
`\btxfoundtype`	*fetch the field type of a set (or field)*

A few helpers are provided to inject symbols but also take care of leading and trailing spaces:[4]

Table 6.6 Punctuation macros

`\btxspace`	*before after*
`\btxnbsp`	*before after*
`\btxnobreakspace`	*before after (same as `\btxnbsp`)*
`\btxperiod`	*before. after*
`\btxcomma`	*before, after*
`\btxcommabreak`	*before,after (allows a line break)*
`\btxleftparenthesis`	*before (after*
`\btxrightparenthesis`	*before) after*
`\btxrightparenthesiscomma`	*before), after*
`\btxrightparenthesisperiod`	*before). after*
`\btxleftbracket`	*before [after*
`\btxrightbracket`	*before] after*
`\btxrightbracketcomma`	*before], after*
`\btxrightbracketperiod`	*before]. after*

Footnotes

[1] For example, a variant on a specification can be defined by only addressing specific differences, for example `specification=rmp` would be associated with `default=aps`, as this describes a style used by the journal: The Review of Modern Physics, mostly based on the style of the American Physical Society.

[2] The difference between 'required' and 'optional' is only a question of diagnostics as described in the Appendices. Their functional meaning is rather 'handled' in contrast to 'ignored'.

[3] Sometimes a set will be defined using the same name as its first element, sometimes it can be given a unique name (typically ending in 'set').

[4] These make use of the ConTEXt command `\removeunwantedspaces`. There is also a 'secret' command `\removepunctuation` that can be quite useful, but also lead to undesired consequences when wielded blindly!

7 Exporting datasets

A dataset can be much more complete than necessary. Furthermore, it may be desirable to isolate bibliography data by chapter, for example, or by any other structural element such as part. To achieve this, it is possible to export the entries of a dataset following certain criteria.

Take as an example, the bibliography of the present manual that was entered in the source as buffers for convenience. The entries can be saved to a legitimate bibTEX format file using:

```
\savebtxdataset
  [default]
  [mkvi-publications.bib]
  [alternative=bib,
   criterium=all]
```

This can also be used to convert older bibliography source files (for example, kept in the MkII format) into the `bib`, `lua` or even `xml` format that could be more convenient for future manipulation.

8 The LUA view

The following is reserved for Lua programmers.

Because we manage data at the Lua end it is tempting to access it there for other purposes. This is fine as long as you keep in mind that aspects of the implementation may change over time, although this is unlikely once the modules become stable.

The entries are collected in datasets and each set has a unique name. In this document we have the set named `example`. *A dataset table has several fields, and probably the one of most interest is the* `luadata` *field. Each entry in this table describes a publication. Take, for example* `publications.datasets.example.luadata["article"]:`

```
t={
 ["abstract"]="Abstract...",
 ["author"]="First Last and First Middle Last and Last, First-Combined and First
van Last and van Last, First Middle and Last, Junior, First Middle and van Last,
Junior III, First Middle and De La, Last, Junior III, First Middle",
 ["category"]="article",
 ["comment"]="Comment...",
 ["day"]="DD-ignored!",
 ["doi"]="doi-specification",
 ["index"]=1,
 ["journal"]="Journal name",
 ["keywords"]="keyword1; keyword2; keyword3",
 ["language"]="english",
 ["month"]="MM-ignored!",
 ["note"]="note...",
 ["number"]="number",
 ["pages"]="ff--tt",
 ["review"]="Review...",
 ["subtitle"]="Subtitle",
 ["tag"]="article",
 ["title"]="Article title",
 ["title:fr"]="Titre de l'article",
 ["type"]="Special Issue",
 ["url"]="url",
 ["volume"]="volume",
 ["warning"]="Notice that Comment= also exists",
 ["xcrossref"]="crossref",
 ["year"]="YYYY",
}
```

There is a companion entry in the parallel `details` *table,*
`publications.datasets.example.details["article"]:`

```
t={
 ["authorhash"]="Last Last Last Last Last Last van Last Last",
 ["authorsuffix"]=2,
 ["referencenumber"]=15,
```

```
 ["shorthash"]="LLL+99",
 ["shortsuffix"]=2,
}
```

tracking further information derived from the publication entry and its use.

You can loop over the entries using regular Lua code combined with MkIV helpers:

```lua
local dataset = publications.datasets.example

context.starttabulate { "|l|l|l|" }
context.NC() context("tag")
context.NC() context("short")
context.NC() context("title")
context.NC() context.NR()
context.HL()
for tag, entry in table.sortedhash(dataset.luadata) do
    local detail = dataset.details[tag] or { }
    context.NC() context.type(tag)
    context.NC() context(detail.shorthash)
    context.NC() context(entry.title)
    context.NC() context.NR()
end
context.stoptabulate()
```

This results in:

tag	short	title
advancedonline	LLL+99	Advanced article title
article	LLL+99	Article title
article-1		Article title
book	BBB99	Book title
booklet	Boo99	Booklet title
conference	Con99	Conference title
editedbook	EE99	Book title
editorial	Edi99	Editorial title
electronic	Ele99	Electronic title
film	Pro99	Film title
inbook	InB99	InBook title
incollection	InC99	InCollection title
inproceedings	InP99	InProceedings title
literal	ZZZ99	
magazine	Mag99	Magazine article title
manual	Man99	Manual title
mastersthesis	Mas99	MastersThesis title
misc	Mis99	Misc title
music	Com99	Music title

newspaper	New99	Article title
noauthor	Pub99	Book title
other	Oth99	Other title
patent	Pat99	Patent title
periodical	Edi99	Periodical title
phdthesis	PhD99	PhDThesis title
proceedings	Edi99	Proceedings title
proceedings-organization	Org99	Proceedings title
selfpublished	Boo99	Book title
selfpublished2	Boo99	Book title
standard	Sta99	Standard title
techreport	Tec99	TechReport title
thesis	The99	Thesis title
unpublished	Unp99	Unpublished title

Notice that the years in this example dataset given as YYYY are interpreted as if they were 9999.

You can manipulate a dataset after loading. Of course this assumes that you know what kind of content you have and what you need for rendering. As example we load a small dataset.

```
\definebtxdataset[drumming]
\usebtxdataset[drumming][mkiv-publications.lua]
```

Because we're going to do some Lua, we could have loaded this dataset using:

```
\startluacode
publications.load("drumming","mkiv-publications.lua","lua")
\stopluacode
```

The dataset has three entries:[1]

```
return {
    ["GH0001"] = {
        category  = "book",
        title     = "Rhythmic Illusions",
        subtitle  = "for drums",
        author    = "Gavin Harrison",
        publisher = "Warner",
        isbn      = "1576236870",
        year      = "1996",
        comment   = "plus cd",
    },
    ["GH0002"] = { -- no reference in brittisch library
        category  = "book",
        title     = "Rhythmic Perspectives",
```

```
        subtitle  = "a multidimensional study of rhythmic composition",
        author    = "Gavin Harrison",
        publisher = "Alfred Publishing Co., Inc",
        year      = "1999",
        comment   = "plus cd",
    },
    ["GH0003"] = {
        category  = "book",
        title     = "Rhythmic Designs",
        subtitle  = "a study of practical creativity",
        author    = "Gavin Harrison and Terry Branham",
        publisher = "Hudson",
        year      = "2010",
        comment   = "plus dvd",
    },
}
```

As you can see, we can have a subtitle. As an exercise, we will combine the title and subtitle into one:

```
\startluacode
local luadata = publications.datasets.drumming.luadata

for tag, entry in next, luadata do
    if entry.subtitle then
        if entry.title then
            entry.title = entry.title .. ", " .. entry.subtitle
        else
            entry.title = entry.subtitle
        end
        entry.subtitle = nil
        logs.report("btx",
            "combining title and subtitle of entry tagged %a into %a",
            tag,entry.title)
    end
end
\stopluacode
```

As a hash comes in a different order each run (something that demands a lot of care in multi-pass workflows that save data in between), so it is probably better to use this instead:

```
\startluacode
local ordered = publications.datasets.drumming.ordered

for i=1,#ordered do
    local entry = ordered[i]
    if entry.subtitle then
```

59

```
        if entry.title then
            entry.title = entry.title .. ", " .. entry.subtitle
        else
            entry.title = entry.subtitle
        end
        entry.subtitle = nil
        logs.report("btx",
            "combining title and subtitle of entry tagged %a into %a",
            entry.tag,entry.title)
    end
end
\stopluacode
```

This loops processes in the order of definition. Alternately, one can sort by tag:

```
\startluacode
local luadata = publications.datasets.drumming.luadata

for tag, entry in table.sortedhash(luadata) do
    if entry.subtitle then
        if entry.title then
            entry.title = entry.title .. ", " .. entry.subtitle
        else
            entry.title = entry.subtitle
        end
        entry.subtitle = nil
        logs.report("btx",
            "combining title and subtitle of entry tagged %a into %a",
            entry.tag,entry.title)
    end
end
\stopluacode
```

The original data is stored in a Lua table, hashed by tag. Starting with Lua 5.2 each run of Lua gets a different ordering of such a hash. In older versions, when you looped over a hash, the order was undefined, but the same as long as you used the same binary. This had the advantage that successive runs, something we often have in document processing gave consistent results. In today's Lua we need to do much more sorting of hashes before we loop, especially when we save multi–pass data. It is for this reason that the xml tree is sorted by hash key by default. That way lookups (especially the first of a set) give consistent outcomes.

We can now simply typeset the entries with:

```
\definebtxrendering[drumming][dataset=drumming,method=dataset]
\placebtxrendering[drumming]
```

1. Harrison, G. (1996). *Rhythmic Illusions, for drums*. Warner.

2. _____ (1999). *Rhythmic Perspectives, a multidimensional study of rhythmic composition.* Alfred Publishing Co., Inc.

3. Harrison, G. & Branham, T. (2010). *Rhythmic Designs, a study of practical creativity.* Hudson.

Because we just want to show the entries, and have no citations that force them to be shown, we have to set the `method` *to* `dataset`*.*

Of course, none of these manipulations in Lua are really necessary, as the rendering could be setup as:

```
\starttexdefinition btx:default:title
    \btxfetch{author}
    \btxdoif{subtitle} {
        \btxcomma
        \btxfetch{subtitle}
    }
\stoptexdefinition
```

which is indeed the case in many of the styles (the `default` *style uses* `\btxcolon`*).[2]*

It is always a question of how much should be done in Lua and how much should be done in T$_E$X. In the end, it is often a question of taste.

Footnotes

[1] Gavin Harrison is in my (Hans) opinion one of the most creative, diverse and interesting drummers of our time. It's also fascinating to watch him play and a welcome distraction from writing code and manuals.

[2] The specifications could be modified to use a parameter `inbetween={, }` for titles:subtitles that the user can easily setup as needed. But as such style questions are, in general, well defined in the specifications, this was not deemed necessary.

9 Extensions

As TeX and Lua are both open and accessible in ConTeXt it is possible to extend the functionality of the bibliography related code. For instance, you could add extra loaders, sketched as follows:

```
\startluacode
function publications.loaders.myformat(dataset,filename)
    local t = { }
    -- Load data from 'filename' and convert it to a Lua table 't' with
    -- the key as hash entry and fields conforming the luadata table
    -- format.
    loaders.lua(dataset,t)
end
\stopluacode
```

This would then permit the loading a database (into a dataset) with the command:

```
\usebtxdataset[default][myfile.myformat]
```

The myformat suffix is recognized automatically. If you want to use another suffix, you can do this:

```
\usebtxdataset[default][myformat::myfile.txt]
```

10 Other use of datasets

Because a bibliography is just a kind of database, you can use the publications mechanism for other purposes as well. During the re-implementation ojca came up with the following definitions:

```
\startbuffer[duane]
@IMAGE {tug2013,
    title       = "TUG 2013",
    url         = "http://tug.org/tug2013/",
    url_image   = "http://tug.org/tug2013/tug2013-color-300.jpg",
    url_thumb   = "http://tug.org/tug2013/t2013-thumb.jpg",
    description = "Official drawing of the TUG 2013 conference",
    author      = "Duane Bibby",
    year        = 2013,
    copyright   = "TUG",
}

@IMAGE {tug2014,
    title       = "TUG 2014",
    url         = "http://tug.org/tug2014/",
    url_image   = "http://tug.org/art/tug2014-color.jpg",
    url_thumb   = "http://tug.org/tug2014/t2014-thumb.jpg",
    description = "Official drawing of the TUG 2014 conference",
    author      = "Duane Bibby",
    year        = 2014,
    copyright   = "TUG",
}
\stopbuffer
```

For documentation purposes we can have a definition in a buffer so that we can show it verbatim but also load it. The following code defines a dataset, loads the buffer and sets up a rendering.

```
\definebtxdataset
  [duane]

\usebtxdataset
  [duane]
  [duane.buffer]

\definebtx
  [duane]
  [default=,
   specification=duane,
   authorconversion=normal]

\definebtx
  [duane:list]
```

```
    [duane]

\definebtx
  [duane:list:author]
  [duane:list]

\definebtx
  [duane:list:image]
  [duane:list]

\definebtxrendering
  [duane]
  [specification=duane,
   dataset=duane,
   method=dataset,
   numbering=no,
   criterium=all]
```

Instead of for instance apa we create a specification named duane. Because categories are rendered with a setup, we can define the following:

```
\startsetups btx:duane:list:image
  \tbox \bgroup
    \bTABLE[offset=1ex]
      \bTR
        \bTD[ny=4]
          \dontleavehmode
          \externalfigure[\btxflush{url_thumb}][width=3cm]
        \eTD
        \bTD \btxflush{title} \eTD
      \eTR
      \bTR
        \bTD \btxflush{author} \eTD
      \eTR
      \bTR
        \bTD \btxflush{description} \eTD
      \eTR
      \bTR
        \bTD
          \goto{high res variant}[url(\btxflush{url_image})]
        \eTD
      \eTR
    \eTABLE
  \egroup
\stopsetups

\placebtxrendering[duane][criterium=all]
```

	TUG 2013
	Duane Bibby
	Official drawing of the TUG 2013 conference
	high res variant

	TUG 2014
	Duane Bibby
	Official drawing of the TUG 2014 conference
	high res variant

An alternative rendering is:

```
\startsetups btx:duane:list:image
  \bgroup
    \bTABLE[offset=1ex]
      \bTR
        \bTD
          \dontleavehmode
          \goto{\externalfigure[\btxflush{url_thumb}][width=3cm]}
            [url(\btxflush{url_image})]
        \eTD
        \bTD
          \bold{\btxflush{title}}
          \blank
          \btxflush{description}
          \blank [3*line]
          \btxflush{author}
        \eTD
      \eTR
    \eTABLE
  \egroup
\stopsetups

\placebtxrendering[duane]
```
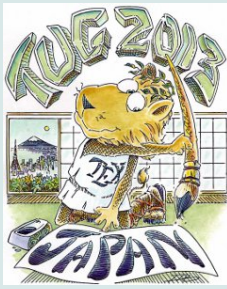
TUG 2013

Official drawing of the TUG 2013 conference

Duane Bibby

TUG 2014

Official drawing of the TUG 2014 conference

Duane Bibby

We only get the second rendering because we specified `criterium` *as* `all`*. Future version of ConTEXt will probably provide sorting options and ways to plug in additional functionality.*

11 Tracing

There are several tracing options. If you want to see where a citations refers to and where a list entry point back to, you can say:

\enabletrackers[publications.crosslinks**]**

This injects markers in both places. One list entry can point to multiple citations. The other tracers a more for debugging and can generate lots of messages.

```
publications
publications.authorhash
publications.cite
publications.cite.missing
publications.cite.references
publications.detail
publications.duplicates
publications.match
publications.sorters
publications.suffixes
```

You can also trace the databases. Take this one:

```
\startbuffer[phony]
@comment { warning :
    Hello Allan!
    How are you doing today?
}

@CoMMeNT { message :
    Hello Allan!
    How are you doing today?
}

@comment {
    all kind of crap
}

@Article{Myarticle,
  Title                   = {My title},
  Author                  = {Myself, Me},
  Journal                 = {My favorite journal},
  Year                    = {2014},
  Pages                   = {1},
}

@Foo{foo,
  Goo = goo,
}
\stopbuffer
```

When we load this database (buffer) with:

`\usebtxdataset [phony] [phony.buffer]`

We get this on the console and in the log

```
publications > adding bib data to set 'phony' from source 'phony'
publications > phony > warning : Hello Allan!
publications > phony > warning : How are you doing today?
publications > phony > message : Hello Allan!
publications > phony > message : How are you doing today?
```

You can use this feature to add warnings to your database for entries that you need to check.

You can also use comment to hide entries:

A Dataset fields

The field names for each entry category are colored according to their status with respect to the bibliography rendering style. The 'required' field names are colored in green, where required means that if this field is missing, one is probably not using the correct entry type or category. Some fields are 'optional' (and some 'required' fields can become conditionally optional, see below). Fields that normally would be ignored (defined as neither required, nor optional) are colored in dark yellow. Some fields are treated as 'special', that is are used internally, not necessarily directly rendered, and are colored here in blue.

tag	category	fields
advancedonline	article	author journal language message title url
article	article	abstract author comment day doi journal keywords language month note number pages review subtitle title title:fr type url volume warning xcrossref year
article-1	article	abstract author comment day doi journal keywords language month note number pages review subtitle title title:fr type url volume warning xcrossref year
book	book	abstract address author comment doi edition editor keywords language month note number percentcrossref publisher review series title type url volume year
booklet	booklet	abstract address author comment doi howpublished keywords language month note percentcrossref review title url year
conference	conference	abstract address author booktitle comment doi editor keywords language month note number organization pages percentcrossref publisher review series title url volume year
editedbook	book	address comment editor language publisher title year
editorial	newspaper	abstract comment day editor journal keywords language message month note number pages percentcrossref review title type url year
electronic	electronic	abstract address author comment doi howpublished keywords language month note organization percentcrossref review title url year

film	film	abstract address comment director keywords language note percentcross-ref producer publisher review title url year
inbook	inbook	abstract address author chapter comment doi edition editor keywords language month note number pages percentcrossref publisher review series title type url volume year
incollection	incollection	abstract address author booktitle chapter comment doi edition editor keywords language month note number pages percentcrossref publisher review series title type url volume year
inproceedings	inproceedings	abstract address author booktitle comment doi editor keywords language month note number organization pages percentcrossref publisher review series title url volume year
literal	literal	key text
magazine	magazine	abstract author comment day journal keywords language message month note number pages percentcrossref review title url year
manual	manual	abstract address author comment doi edition keywords month note organization review title url year
mastersthesis	mastersthesis	abstract address author comment doi keywords language month note percentcrossref review school title type url year
misc	misc	abstract author comment doi howpublished keywords language month note percentcrossref review title url year
music	music	abstract address album artist comment composer keywords language note percentcrossref publisher review title url year
newspaper	newspaper	abstract author comment day journal keywords language message month note number pages percentcrossref review title url year
noauthor	book	address comment language publisher title year

other	other	abstract author comment doi keywords language note percentcrossref review title url year
patent	patent	abstract author comment doi keywords language nationality note number percentcrossref review title url year yearfiled
periodical	periodical	abstract author comment day editor journal keywords language month note number organization percentcrossref review series title url volume year
phdthesis	phdthesis	abstract author comment doi keywords language note percentcrossref review school title url year
proceedings	proceedings	abstract address comment doi editor keywords month note number organization percentcrossref publisher review series title url volume year
proceedings-organization	proceedings	abstract address comment day doi keywords month note number organization percentcrossref publisher review series title url volume year
selfpublished	book	address author comment language title year
selfpublished2	book	author crossref
standard	standard	abstract author comment doi institution keywords language note organization percentcrossref review title url year
techreport	techreport	abstract address author comment doi institution keywords language month note number percentcrossref review title type url year
thesis	thesis	abstract address author comment doi keywords language month note percentcrossref review school title type url year
unpublished	unpublished	abstract author comment doi keywords language month note percentcrossref review title url year

B Dataset completeness

\showbtxdatasetcompleteness[example**]**

The listing can be rather long ... The header for each entry gives the `index`, `category` and `tag`.

3	**article**	**advancedonline**
required	author	LastnameA, FirstnameA and LastnameB, FirstnameB and LastnameC, FirstnameC and LastnameD, First-nameD and LastnameE, FirstnameE and LastnameF, FirstnameF and LastnameG, FirstnameG
	title	Advanced article title
optional	journal	Journal name
	url	url
special	language	english
extra	message	This is an example of an advanced online publication.

1	**article**	**article**
required	author	First Last and First Middle Last and Last, First-Combined and First van Last and van Last, First Middle and Last, Junior, First Middle and van Last, Junior III, First Middle and De La, Last, Junior III, First Middle
	title	Article title
optional	year	YYYY
	subtitle	Subtitle
	type	Special Issue
	journal	Journal name
	volume	volume
	number	number
	pages	ff--tt
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	keyword1; keyword2; keyword3
	language	english
extra	abstract	Abstract...
	day	DD-ignored!
	month	MM-ignored!
	review	Review...
	title:fr	Titre de l'article
	warning	Notice that Comment= also exists
	xcrossref	crossref

2	**article**	**article-1**
required	author	First Last and First Middle Last and Last, First-Combined and First van Last and van Last, First Middle and Last, Junior, First Middle and van Last, Junior III, First Middle and De La, Last, Junior III, First Middle
	title	Article title
optional	year	YYYY
	subtitle	Subtitle
	type	Special Issue
	journal	Journal name
	volume	volume
	number	number
	pages	ff--tt
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	keyword1; keyword2; keyword3
	language	english
extra	abstract	Abstract...
	day	DD-ignored!
	month	MM-ignored!
	review	Review...
	title:fr	Titre de l'article
	warning	Notice that Comment= also exists
	xcrossref	crossref

7	**book**	**book**
required	author	BookAuthorLastnameA, Firstname and BookAuthor-LastnameB, Firstname Middle and BookAuthorLast-nameC, Firstname-Combined
	editor	EditorLastname, Firstname
	publisher	Publisher
	title	Book title
optional	year	YYYY
	month	Month
	type	Translated title, for example
	edition	edition
	volume	volume
	number	number
	series	series
	address	Address
	doi	doi-specification
	url	url

	note	note...
special	comment	Comment...
	keywords	book
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

11	**booklet**	**booklet**
required	author	BookletAuthorLastname, Firstname
	title	Booklet title
optional	howpublished	howpublished
	year	YYYY
	month	Month
	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	booklet
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

12	**conference**	**conference**
required	author	ConferenceAuthorLastname, Firstname
	editor	EditorLastname, Firstname
	publisher	publisher
	title	Conference title
	booktitle	Booktitle
	year	YYYY
optional	month	Month
	series	series
	address	Address
	organization	<Symposium conducted at the meeting of> Organization
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	conference
	language	english
extra	abstract	Abstract...
	number	number
	pages	ff--tt

	percentcrossref	crossref
	review	Review...
	volume	volume

9	**book**	**editedbook**
required	editor	EditorLastnameA, Firstname and EditorLastnameB, Firstname
	publisher	Publisher
	title	Book title
optional	year	YYYY
	address	Address
special	comment	Editors as authors
	language	english

6	**newspaper**	**editorial**
required	editor	EditorLastname, Firstname
	title	Editorial title
	year	YYYY
	journal	Newspaper name
optional	type	Editorial
	number	number
	month	Month
	day	DD
	url	url
	note	note...
special	comment	Comment...
	keywords	editorial
	language	english
extra	abstract	Abstract...
	message	This is a newspaper editorial.
	pages	ff--tt
	percentcrossref	crossref
	review	Review...

13	**electronic**	**electronic**
required	title	Electronic title
optional	year	YYYY
	month	Month
	author	ElectronicAuthorLastname, Firstname
	address	Address
	organization	Organization
	howpublished	howpublished
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...

	keywords	electronic
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

14	**film**	**film**
required	producer	ProducerLastName, Firstname
	director	DirectorLastname, Firstname
	title	Film title
	year	YYYY
	address	address
	publisher	studio
optional	note	note...
	url	url
special	comment	Comment...
	keywords	misc
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

15	**inbook**	**inbook**
required	author	InBookAuthorLastname, Firstname
	editor	EditorLastname, Firstname
	publisher	publisher
	title	InBook title
	year	YYYY
optional	type	type
	edition	edition
	volume	volume
	number	number
	pages	ff--tt
	series	series
	month	Month
	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	inbook
	language	english
extra	abstract	Abstract...
	chapter	Chapter title
	percentcrossref	crossref
	review	Review...

16	incollection	incollection
required	author	InCollectionAuthorLastname, Firstname
	editor	EditorLastname, Firstname
	publisher	publisher
	title	InCollection title
	booktitle	Booktitle
	year	YYYY
optional	type	type
	edition	edition
	volume	volume
	number	number
	pages	ff--tt
	series	series
	month	Month
	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	incollection
	language	english
extra	abstract	Abstract...
	chapter	Chaptertitle
	percentcrossref	crossref
	review	Review...

17	inproceedings	inproceedings
required	author	InProceedingsAuthorLastname, Firstname
	editor	EditorLastname, Firstname
	publisher	publisher
	title	InProceedings title
	booktitle	Booktitle
	year	YYYY
optional	month	Month
	series	series
	address	Address
	organization	<Symposium conducted at the meeting of> Organization
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	inproceedings
	language	english
extra	abstract	Abstract...

	number	number
	pages	ff--tt
	percentcrossref	crossref
	review	Review...
	volume	volume

32	**literal**	**literal**
required	key	ZZZZ-end
	text	Copied literally to the bibliography

4	**magazine**	**magazine**
required	author	MagazineLastname, Firstname
	title	Magazine article title
	year	YYYY
	journal	Magazine name
optional	number	number
	month	Month
	day	DD
	url	url
	note	note...
special	comment	Comment...
	keywords	magazine
	language	english
extra	abstract	Abstract...
	message	This is a magazine article.
	pages	ff--tt
	percentcrossref	crossref
	review	Review...

18	**manual**	**manual**
required	title	Manual title
optional	author	ManualLastname, Firstname
	organization	Organization
	title	Manual title
	organization	Organization
	address	Address
	edition	Edition
	month	Month
	year	YYYY
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	manual
extra	abstract	Abstract...
	review	Review...

19	**mastersthesis**	**mastersthesis**
required	author	MastersThesisAuthorLastname, Firstname
	title	MastersThesis title
	school	school
	year	YYYY
optional	type	type
	month	Month
	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	mastersthesis
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

21	**misc**	**misc**
optional	author	MiscAuthorLastname, Firstname
	title	Misc title
	year	YYYY
	month	Month
	howpublished	howpublished
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	misc
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

22	**music**	**music**
required	composer	ComposerLastName, Firstname
	artist	ArtistLastname, Firstname
	title	Music title
	album	Album title
	title	Music title
	album	Album title
	year	YYYY
	address	address
	publisher	label
optional	note	note...
	url	url

special	comment	Comment...
	keywords	misc
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

5	**newspaper**	**newspaper**
required	author	NewspaperLastname, Firstname
	title	Article title
	year	YYYY
	journal	Newspaper name
optional	number	number
	month	Month
	day	DD
	url	url
	note	note...
special	comment	Comment...
	keywords	newspaper
	language	english
extra	abstract	Abstract...
	message	This is a newspaper article.
	pages	ff--tt
	percentcrossref	crossref
	review	Review...

10	**book**	**noauthor**
required	publisher	Publisher
	title	Book title
optional	year	YYYY
	address	Address
special	comment	Publisher as author
	language	english

23	**other**	**other**
required	author	OtherAuthorLastname, Firstname
	title	Other title
	year	YYYY
optional	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	other
	language	english
extra	abstract	Abstract...

	percentcrossref	crossref
	review	Review...

24	**patent**	**patent**
required	nationality	nationality
	number	number
	year	YYYY
	yearfiled	YYYYfiled
optional	author	PatentAuthorLastname, Firstname
	title	Patent title
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	patent
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

25	**periodical**	**periodical**
required	title	Periodical title
	year	YYYY
optional	editor	EditorLastname, Firstname
	series	series
	volume	volume
	number	number
	month	Month
	organization	Organization
	url	url
	note	note...
special	comment	Complete issue of a periodical, such as a special issue of a journal
	keywords	periodical
	language	english
extra	abstract	Abstract...
	author	PeriodicalAuthorLastname, Firstname
	day	DD
	journal	Periodical Journal
	percentcrossref	crossref
	review	Review...

26	**phdthesis**	**phdthesis**
required	author	PhDThesisAuthorLastname, Firstname
	title	PhDThesis title
	school	school

	year	YYYY
optional	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	phdthesis
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

27	**proceedings**	**proceedings**
required	editor	EditorLastname, Firstname
	organization	\<Symposium conducted at the meeting of> Organization
	publisher	Publisher
	title	Proceedings title
	year	YYYY
optional	publisher	Publisher
	organization	\<Symposium conducted at the meeting of> Organization
	volume	volume
	number	number
	series	series
	month	Month
	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	proceedings
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

28	**proceedings**	**proceedings-organization**
required	organization	Organization
	publisher	Publisher
	title	Proceedings title
	year	YYYY
optional	publisher	Publisher
	organization	Organization
	volume	volume
	number	number
	series	series
	month	Month

	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Conference preceedings with no identified edi-tor(s)
	keywords	proceedings
extra	abstract	Abstract...
	day	DD
	percentcrossref	crossref
	review	Review...

8	**book**	**selfpublished**	
required	author	BookAuthorLastname, Firstname	
	title	Book title	
optional	year	YYYY	
	address	Address	
special	comment	A self-published book	
	language	english	

33	**book**	**selfpublished2**	**=> selfpublished**
required	author	BookAuthorLastname2, Firstname	
	title	Book title	
optional	year	YYYY	
	address	Address	
special	crossref	selfpublished	

29	**standard**	**standard**
required	author	StandardAuthorLastname, Firstname
	institution	institution
	organization	Organization
	year	YYYY
	title	Standard title
	subtitle	[missing value]
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	standard
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

30	**techreport**	**techreport**
required	author	TechReportAuthorLastname, Firstname

	institution	institution
	title	TechReport title
	title	TechReport title
	institution	institution
	year	YYYY
optional	institution	institution
	address	Address
	type	type
	number	number
	month	Month
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	techreport
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

20	**thesis**	**thesis**
required	author	ThesisAuthorLastname, Firstname
	title	Thesis title
	school	school
	year	YYYY
	type	type of thesis
optional	month	Month
	address	Address
	doi	doi-specification
	url	url
	note	note...
special	comment	Comment...
	keywords	mastersthesis
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

31	**unpublished**	**unpublished**
required	author	UnpublishedAuthorLastname, Firstname
	title	Unpublished title
	note	note...
optional	year	YYYY
	month	Month
	doi	doi-specification
	url	url

special	comment	Comment...
	keywords	unpublished
	language	english
extra	abstract	Abstract...
	percentcrossref	crossref
	review	Review...

C Performance

The move from external bibTeX processing to internal processing has the advantage that we stay within the same run. In the traditional approach we had roughly the following steps:

- In the first ConTeXt run information is collected and written to file.
- fter the first run the bibTeX program converts that file to another one, a so called `bbl` file with TeX commands.
- Successive runs use that file for typesetting references and producing lists of publications that are reffered to.

In the MkIV approach the bibliographic database is loaded in memory each run and processing also happens each run. On paper this looks less efficient but as Lua is quite fast, in practice performance is much better.

Probably most demanding is the treatment of authors as we have to analyze names, split multiple authors and reassemble firstnames, vons, surnames and juniors. When we sort by author sorting vectors have to be made which also has a penalty. However, in practice the user will not notice a performance degradation. We did some tests with a list of 500.000 authors, sorted them and typeset them as list (producing some 5400 dense pages in a small font and with small margins). This is typical one of these cases where using LuajitTeX saves quite time. On my machine it took just over 100 seconds to get this done. Unfortunately not all operating systems performed equally well: 32 bit versions worked fine, but 64 bit linux either crashed (stalled) the machine or ran out of memory rather fast, while MacOSX and Windows performed fine. In practice you will never run into this, unless you produce massive amounts of bibliographic entries. LuaJIT has some benefits but also some drawbacks. In practice you will not run into these problems.

Bibliography

6   O. Patashnik, "BibTeXing", 1988. (6)

7   N. Markey, "Tame the BeaST: The B to X of BibTeX", 2009. (6)

8   R. Ierusalimschy, *Programming in Lua*, Lua.org, 2006. (11 and 41)

9   American Psychological Association, *Publication Manual of the American Psychological Association* (Sixth ed.), American Psychological Association, Washington, DC, 2010. (25)

List of tables

Index of names

Index

Hashed authors

hash	vons	surnames	initials	firstnames	juniors
Aditya Mahajan		Mahajan	A.	Aditya	
Alan Braslau		Braslau	A.	Alan	
Alexander Berdnikov		Berdnikov	A.	Alexander	
ArtistLastname, Firstname		ArtistLastname	F.	Firstname	
Avi Naiman		Naiman	A.	Avi	
Bogusław Jackowski		Jackowski	B.	Bogusław	
BookAuthorLastname, Firstname		BookAuthorLastname	F.	Firstname	
BookAuthorLastname2, Firstname		BookAuthorLastname2	F.	Firstname	
BookAuthorLastnameA, Firstname		BookAuthorLastnameA	F.	Firstname	
BookAuthorLastnameB, Firstname Middle		BookAuthorLastnameB	F. M.	Firstname Middle	
BookAuthorLastnameC, Firstname-Combined		BookAuthorLastnameC	F.-C.	Firstname-Combined	
BookletAuthorLastname, Firstname		BookletAuthorLastname	F.	Firstname	
Cleve Moler		Moler	C.	Cleve	
ComposerLastName, Firstname		ComposerLastName	F.	Firstname	
ConferenceAuthorLastname, Firstname		ConferenceAuthorLastname	F.	Firstname	
Darwin, C.		Darwin	C.	C.	
David Kahaner		Kahaner	D.	David	
De La, Last, Junior III, First Middle	De La	Last	F. M.	First Middle	Junior III
Debra Anne Adams		Adams	D. A.	Debra Anne	
DirectorLastname, Firstname		DirectorLastname	F.	Firstname	
Duane Bibby		Bibby	D.	Duane	
EditorLastname, Firstname		EditorLastname	F.	Firstname	
EditorLastnameA, Firstname		EditorLastnameA	F.	Firstname	
EditorLastnameB, Firstname		EditorLastnameB	F.	Firstname	
ElectronicAuthorLastname, Firstname		ElectronicAuthorLastname	F.	Firstname	
First Last		Last	F.	First	
First Middle Last		Last	F. M.	First Middle	
First van Last	van	Last	F.	First	
Gavin Harrison		Harrison	G.	Gavin	
Gerhardt, C. G.		Gerhardt	C. G.	C. G.	
Hans Hagen		Hagen	H.	Hans	
Ierusalimschy, R.		Ierusalimschy	R.	R.	
InBookAuthorLastname, Firstname		InBookAuthorLastname	F.	Firstname	
InCollectionAuthorLastname, Firstname		InCollectionAuthorLastname	F.	Firstname	
InProceedingsAuthorLastname, Firstname		InProceedingsAuthorLastname	F.	Firstname	
Jacques André		André	J.	Jacques	
Janusz M. Nowacki		Nowacki	J. M.	Janusz M.	
Janusz Nowacki		Nowacki	J.	Janusz	
John D. Hobby		Hobby	J. D.	John D.	
Last, First-Combined		Last	F.-C.	First-Combined	
Last, Junior, First Middle		Last	F. M.	First Middle	Junior
LastnameA, FirstnameA		LastnameA	F.	FirstnameA	
LastnameB, FirstnameB		LastnameB	F.	FirstnameB	
LastnameC, FirstnameC		LastnameC	F.	FirstnameC	
LastnameD, FirstnameD		LastnameD	F.	FirstnameD	
LastnameE, FirstnameE		LastnameE	F.	FirstnameE	
LastnameF, FirstnameF		LastnameF	F.	FirstnameF	
LastnameG, FirstnameG		LastnameG	F.	FirstnameG	
Leibniz, G. W.		Leibniz	G. W.	G. W.	
MagazineLastname, Firstname		MagazineLastname	F.	Firstname	
ManualLastname, Firstname		ManualLastname	F.	Firstname	
Marek Ryćko		Ryćko	M.	Marek	
Markey, Nicolas		Markey	N.	Nicolas	
MastersThesisAuthorLastname, Firstname		MastersThesisAuthorLastname	F.	Firstname	
Michael A. Harrison		Harrison	M. A.	Michael A.	
Michael E. Lesk		Lesk	M. E.	Michael E.	
MiscAuthorLastname, Firstname		MiscAuthorLastname	F.	Firstname	
NewspaperLastname, Firstname		NewspaperLastname	F.	Firstname	
OtherAuthorLastname, Firstname		OtherAuthorLastname	F.	Firstname	
Patashnik, Oren		Patashnik	O.	Oren	
PatentAuthorLastname, Firstname		PatentAuthorLastname	F.	Firstname	
Paul M. English		English	P. M.	Paul M.	
Pehong Chen		Chen	P.	Pehong	
PhDThesisAuthorLastname, Firstname		PhDThesisAuthorLastname	F.	Firstname	
Piotr Pianowski		Pianowski	P.	Piotr	
Piotr Strzelczyk		Strzelczyk	P.	Piotr	
ProducerLastName, Firstname		ProducerLastName	F.	Firstname	
R. Fletcher		Fletcher	R.	R.	
Robert W. Gray		Gray	R. W.	Robert W.	
Roger Hersch		Hersch	R.	Roger	
StandardAuthorLastname, Firstname		StandardAuthorLastname	F.	Firstname	
Stephen Nash		Nash	S.	Stephen	
Steven C. Johnson		Johnson	S. C.	Steven C.	
Taco Hoekwater		Hoekwater	T.	Taco	
TechReportAuthorLastname, Firstname		TechReportAuthorLastname	F.	Firstname	
Terry Branham		Branham	T.	Terry	

Hashed authors

```
ThesisAuthorLastname, Firstname            ThesisAuthorLastname      F.      Firstname
UnpublishedAuthorLastname, Firstname       UnpublishedAuthorLastname F.      Firstname
Willi Egger                                Egger                     W.      Willi
van Last, First Middle              van    Last                      F. M.   First Middle
van Last, Junior III, First Middle         van Last                  F. M.   First Middle      Junior III
Abraxas Software, Inc.                     Abraxas Software, Inc.
Adobe Systems Incorporated                 Adobe Systems Incorporated
```