Alexandria

Alexandria allows a single document to have multiple bibliographies.

v0.2.1 July 08, 2025

https://github.com/SillyFreak/typst-alexandria

Clemens Koza

Contents

I Introduction

Alexandria enables multiple bibliographies within the same Typst document.

In *Alexandria*, each citation is associated with a *prefix*. `alexandria()` function declares a prefix, e.g. `"x-"`, for a group of bibliographical references. After this, you can use regular Typst citations, prepending the prefix to a bibliographic key to indicate that it refers to a specific group, e.g. `@x-quark` or `#cite(<x-netwok>)`. *Alexandria*'s `bibliographyx()` is the equivalent of the built-in `bibliography()` function for generating a bibliography limited to a specific prefix.

Typical usage looks like this:

```typ
#import "@preview/alexandria:0.2.1": *
#show: alexandria(prefix: "x-", read: path => read(path))
#show: alexandria(prefix: "y-", read: path => read(path))

... The text that references @x-quark and @x-netwok ...

#bibliographyx(
  "bibliography.bib",
  prefix: "x-",
  title: "X Bibliography",
)

... The section with references to @y-arggh and @y-distress ...

#bibliographyx(
  "bibliography.bib",
  prefix: "y-",
  title: "Y Bibliography",
)
```

Some known limitations:

- Internally, *Alexandria* citations are converted to links and are thus affected by `link` rules.
- Native bibliographies have `numbering: none` applied to its title, while *Alexandria*'s haven't. `show bibliography: set heading(..)` also won't work on them.
- Citations that are shown as footnotes are not supported yet – see issue #11.

If you find additional limitations or other issues, please report them at https://github.com/SillyFreak/typst-alexandria/issues.

II Separate bibliographies for document sections

Below we demonstrate how to create separate bibliographies with independent numbering for different sections of a document:

- Example II.a.a uses the native Typst bibliography
- Example II.a.b uses Alexandria to generate APA style references for all bibliographical entries from *bibliography.bib* (`full: true`)
- Example II.a.c shows a numbered IEEE style bibliography.

II.a Example

II.a.a Native Typst (APA)

For further information on pirate and quark organizations, see (Leeson, n.d.-a; -b). Aldrin discusses bibliographical distress.

Über den „Netzwok“ ist in der Arbeit von Astley & Morris (2020) zu lesen.

Bibliography

Aldrin, B. *An Insight into Bibliographical Distress*.

Astley, R., & Morris, L. (2020). At-scale impact of the Net Wok: A culinarically holistic investigation of distributed dumplings. *Armenian Journal of Proceedings*, *61*, 192–219.

Leeson, P. T. (n.d.-a). *The Pirate Organization*.

Leeson, P. T. (n.d.-b). *The Quark Organization*.

II.a.b Alexandria (APA)

For further information on pirate and quark organizations, see (Leeson, n.d.-a; -b). Aldrin discusses bibliographical distress.

Über den „Netzwok“ ist in der Arbeit von R. Astley und L. Morris [2] zu lesen.

Bibliography

Aldrin, B. *An Insight into Bibliographical Distress*.

Astley, R., & Morris, L. (2020). At-scale impact of the Net Wok: A culinarically holistic investigation of distributed dumplings. *Armenian Journal of Proceedings*, *61*, 192–219.

Hock, R. (2005). Glacier melt: a review of processes and their modelling. *Progress in Physical Geography: Earth and Environment*, *29*(3), 362–391. https://doi.org/10.1191/0309133305pp453ra

Kopp, J., Gerike, R., & Axhausen, K. W. (2015). Do sharing people behave differently? An empirical evaluation of the distinctive mobility patterns of free-floating car-sharing members. *Transportation*, *42*(3), 449–469.

Leeson, P. T. (n.d.-a). *The Pirate Organization*.

Leeson, P. T. (n.d.-b). *The Quark Organization*.

McIntosh, I. B., Swanson, V., Power, K. G., Raeside, F., & Dempster, C. (2006). Anxiety and Health Problems Related to Air Travel. *Journal of Travel Medicine*, *5*(4), 198–204.

Richardson, L., & Ruby, S. (2008). *RESTful Web Services* (1st ed.). O'Reilly Media.

Strong, E. (1925). *The psychology of selling and advertising* (1st ed.). McGraw-Hill Book Co.

Tolkien, J. R. R. (1954). *The Fellowship of the Ring: 1* (Vol. 1). Allen & Unwin.

II.a.c Alexandria (IEEE)

For further information on pirate and quark organizations, see [1], [2]. B. Aldrin discusses bibliographical distress.

Über den „Netzwok“ ist in der Arbeit von Astley & Morris (2020) zu lesen.

Bibliography

[1] P. T. Leeson, “The Pirate Organization.”

[2] P. T. Leeson, “The Quark Organization.”

[3] B. Aldrin, “An Insight into Bibliographical Distress.”

[4] R. Astley and L. Morris, “At-scale impact of the Net Wok: A culinarically holistic investigation of distributed dumplings,” *Armenian Journal of Proceedings*, vol. 61, pp. 192–219, 2020.

III Splitting bibliographies

In the previous example, the bibliographies were created for separate parts of a document, and each had its own independent numbering. This approach will not work when multiple bibliographies have to serve the same region of the document, because with overlapping numbers the citations become ambiguous. For this scenario, Alexandria allows decoupling *loading* and *collection* of the references from their *rendering*. Instead of a single `bibliographyx()` call:

- `load-bibliography()` loads all bibliographical entries with a specific prefix
- `get-bibliography()` composes a list of entries referenced in the document
- the user can manually filter this list by specific criteria, e.g. by the reference type
- `render-bibliography()` renders the user-specified list of references. This function could be called multiple times, each time with a different subset of references.

A sample Typst code that separates book references from all other types could look like this:

```typ
#import "@preview/alexandria:0.2.1": *
#show: alexandria(prefix: "x-", read: path => read(path))

... The text that cites entries from "x-" ...

#load-bibliography("bibliography.bib")

#context {
  // get the bibliography items + additional information
  let (references: bib_refs, ..bib_info) = get-bibliography("x-")

  // render the non-book bibliography
  render-bibliography(
    title: [Bibliography],
    (
      references: bib_refs.filter(ref => ref.details.type != "book"),
      ..bib_info, // provide other information from get-bibliography()
    ),
  )

  // render the books bibliography (could also be elsewhere in the document)
  render-bibliography(
    title: [Books],
    (
      references: bib_refs.filter(ref => ref.details.type == "book"),
      ..bib_info,
    ),
  )
}
```

Here's how the rendered output would look like. Note that the numbering in the bibliographies is not sequential. It is the result of making the lists non-overlapping to allow citations unambiguosly refer to specific bibliographic entries.

III.a Example

For further information on pirate and quark organizations, see [1], [2]. B. Aldrin discusses bibliographical distress in [3], and [4] is a hefty volume on various aspects of psychology.

Über den „Netzwok" ist in der Arbeit von Astley & Morris (2020) zu lesen.

Bibliography

[1] P. T. Leeson, "The Pirate Organization."

[2] P. T. Leeson, "The Quark Organization."

[3] B. Aldrin, "An Insight into Bibliographical Distress."

[5] R. Astley and L. Morris, "At-scale impact of the Net Wok: A culinarically holistic investigation of distributed dumplings," *Armenian Journal of Proceedings*, vol. 61, pp. 192–219, 2020.

Books

[4] E. Strong, *The psychology of selling and advertising*, 1st ed. New York, NY, USA: McGraw-Hill Book Co., 1925.

IV Module reference

IV.a alexandria

`alexandria(prefix: string, read: function) -> function`

This configuration function should be called as a show rule at the beginning of the document. The function makes sure that `ref()` and `cite()` commands can refer to Alexandria's custom bibliography entries and stores configuration for use by `load-bibliography()`.

```typ
#show: alexandria(prefix: "x-", read: path => read(path))
```

The `read` parameter can be skipped, in which case file paths can not be used for bibliography files and custom styles. This means you will need to pass `bytes` values to `bibliographyx()` and `load-bibliography()` instead of paths.

Parameters:

`prefix` (`string` = `none`) – a prefix that identifies labels referring to Alexandria bibliographies. Bibliography entries will automatically get that prefix prepended.

`read` (`function` = `none`) – pass `path => read(path)` into this parameter so that Alexandria can read your bibliography files.

`citegroup(prefix: string auto, body: content) -> content`

Creates a group of collapsed citations. The citations are given as regular content, e.g.

```typ
#citegroup[@a @b]
```

Only citations, references and spaces may appear in the body. Whitespace is ignored, and the rest is treated as a group of citations to collapse. It is an error to have non-alexandria references, or references from different bibliographies, in the same citation group.

Parameters:

`prefix` (`string` or `auto` = `auto`) – The prefix for which reference labels should be provided and citations should be processed.

`body` (`content`) – The body, containing at least one but usually more citations

```
load-bibliography(
  path: string bytes array,
  prefix: string auto,
  full: boolean,
  style: string bytes,
) -> content
```

Loads an additional bibliography. This reads the relevant bibliography file(s) and stores the extracted data in a state for later retrieval via `get-bibliography()`, but does not render anything yet. For simple use cases, `bibliographyx()` can be used directly.

Even though this only loads the bibliography, this function already requires knowledge of the citations that appear in the document, both to know which references to include (for non-`full` bibliographies) and in what styles, forms and languages these citations should be rendered.

The interface is similar to the built-in `bibliography()`, but not all features are supported (yet). In particular, the default values reflect `bibliography()`, but some of these are not supported yet and need to be set manually. Also, the title parameter (only needed for rendering) is skipped.

Parameters:

`path` (`string` or `bytes` or `array`) – The path to or binary file contents of the bibliography file(s).

`prefix` (`string` or `auto` = `auto`) – The prefix for which reference labels should be provided and citations should be processed.

`full` (`boolean` = `false`) – Whether to render the full bibliography or only the references that are used in the document.

`style` (`string` or `bytes` = `"ieee"`) – The style of the bibliography. Either a built-in style, a file name that is read by the `read()` function registered via `alexandria()`, or binary file contents of a CSL file.

```
get-bibliography(prefix: string auto) -> dict
```

Returns a previously loaded bibliography. This is used implicitly by `bibliographyx()` and Alexandria citations to retrieve rendered data, and can be used directly for more complex use cases. Usually, the returned data will be ultimately rendered using `render-bibliography()`.

The result is a dictionary with the following keys:

- `prefix`: the string prefix used by Alexandria to identify this bibliography (and passed to this function), used as a prefix for all labels rendered by Alexandria.
- `references`: an array of reference dictionaries which can be rendered into a bibliography. The array is sorted by the appearance of references according to the style used.
- `citations`: an array of citations dictionaries which can be rendered into the various citations in the document. The array is sorted by the appearance of citations in the document.
- `hanging-indent`: a boolean indicating whether the citation style uses a hanging indent for its entries.

The `references` in turn each contain

- `key`: the reference key without prefix.

- `reference`: a representation of the Typst content that should be rendered; this is processed by `render-bibliography()` to produce the actual context.
- optionally `prefix`: this is *not* the Alexandria prefix but another Typst content representation for styles that require it. For example, in IEEE style this would represent "[1]" and so on.
- `details`: a dictionary containing several fields of structured data about the reference. Among these are `type`, `title`, `author`, `date`, etc. A full list can be found in the Hayagriva docs.

The `citations` are representations of the Typst content that should be rendered at their respective citation sites.

This function is contextual.

Parameters:

`prefix` (`string` or `auto`) – The prefix for which the bibliography should be retrieved, or `auto` if there is only one bibliography and that one should be retrieved.

```
render-bibliography(bib: dict, title: none content auto) -> content
```

Renders the provided bibliography data (as returned by `get-bibliography()`) with the given title. For simple use cases, `bibliographyx()` can be used directly, which also handles the data retrieval.

You will usually only need to call this directly if you *don't* pass the exact return value of `get-bibliography()` as an argument. Instead, you'll want to preprocess that data, e.g. by filtering out some `references` entries that should appear elsewhere in the document. Note that generally, you'll need to ultimately render all references, or you'll get unresolved citations.

Parameters:

`bib` (`dict`) – The bibliography data

`title` (`none` or `content` or `auto` = `auto`) – The title of the bibliography. Note that `auto` is currently not supported.

```
bibliographyx(
  path: string bytes array,
  prefix: string auto,
  title: none content auto,
  full: boolean,
  style: string bytes,
) -> content
```

Renders an additional bibliography. The interface is similar to the built-in `bibliography()`, but not all features are supported (yet). In particular, the default values reflect `bibliography()`, but some of these are not supported yet and need to be set manually.

```typ
#bibliographyx(
  "bibliography.bib",
```

```
3   title: "Bibliography",
4   full: true,
5   style: "ieee",
6 )
```

This function is based on `load-bibliography()`, `get-bibliography()`, and `render-bibliography()` and simply reproduces the rendering of the built-in bibliography without modification.

Parameters:

`path` (`string` or `bytes` or `array`) – The path to or binary file contents of the bibliography file(s).

`prefix` (`string` or `auto` = `auto`) – The prefix for which reference labels should be provided and citations should be processed.

`title` (`none` or `content` or `auto` = `auto`) – The title of the bibliography. Note that `auto` is currently not supported.

`full` (`boolean` = `false`) – Whether to render the full bibliography or only the references that are used in the document.

`style` (`string` or `bytes` = `"ieee"`) – The style of the bibliography.