Alexandria

Alexandria allows a single document to have multiple bibliographies.

v0.3.0 April 23, 2025

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

Clemens Koza

Contents

I Introduction

Alexandria enables multiple bibliographies within the same Typst document.

With *Alexandria*, each citation citation is associated with a *prefix*. To specify the prefix, one can explicitly prepend it to the bibliograhic key (e.g. `@x-quark` or `#cite(<y-arggh>)`) or set the default prefix with this `alexandria-prefix()` function. `bibliographyx()` allows generating a bibliography limited to the citations with a specific prefix.

Typical usage would look something like this:

```typ
#import "@preview/alexandria:0.3.0": *
#show: alexandria("bibliography.bib", reader: path => read(path))

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

#bibliographyx(
  prefix: "x",
  title: "X Bibliography",
)

...

Lets use `y-` prefix by default for the rest of the document.

#alexandria-prefix("y")

...
Here we are referencing @arggh and @y-distress.
...

#bibliographyx(
  prefix: "y",
  title: "Y Bibliography",
)
```

With this setup, you can use regular Typst citations (to keys starting with the configured prefix) to cite entries in an Alexandria bibliography.

Some known limitations:

- Alexandria citations are converted to links and are thus affected by `link` rules.
- Native bibliographies have `numbering: none` applied to its title, while Alexandrias' 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 Example: 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 uses the native Typst bibliography
- Example II.b uses explicit prefix specification (`x`) for Alexandria bibliography with APA-style references
- Example II.c shows how `alexandria-prefix()` could be used to specify the default prefix (`y`) and generate the IEEE-style bibliography.

II.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.

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.b Alexandria (APA)

For further information on pirate and quark organizations, see (P. T. Leeson, n.d.;). 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.

Leeson, P. T. *The Pirate Organization.*

Leeson, P. T. *The Quark Organization.*

II.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 we 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 *collection* of the references from their *rendering*. Instead of a single `bibliographyx()` call:

- `collect-citations()` assembles the combined list of all bibliographical entries with prefixes that match the user-specified criteria
- `render-bibliography()` renders the subset of this list, further filtering its entries by the prefix or other properties. `render-bibliography()` could be called multiple times, each time with a different filter.

An example Typst code could look like this:

```typ
#import "@preview/alexandria:0.3.0": *
#show: alexandria("bibliography.bib", reader: path => read(path))

...

#collect-citations("a_and_b", prefix-filter: ("a", "b"))

#render-bibliography("a_and_b",
  filter: ref => ref.prefixes.contains("a"),
  title: "A Bibliography",
)

#render-bibliography("a_and_b",
  filter: ref => ref.prefixes.contains("b") and ref.details.type != "book",
  title: "B Articles",
)

#render-bibliography("a_and_b",
  filter: ref => ref.prefixes.contains("b") and ref.details.type == "book",
  title: "B Books",
)
```

Here's the rendered output:

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.

A Bibliography

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

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

B Articles

[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.

B 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()`
- `alexandria()`
- `citegroup()`
- `collect-citations()`
- `render-bibliography()`
- `bibliographyx()`

```
alexandria-prefix(prefix) -> state
```

Sets the default prefix for Alexandria references and citations.

```typ
#alexandria-prefix(prefix: "x")
```

- prefix (string | none): the default prefix to use for the Alexandria references and citations.

```
alexandria(
  sources: string bytes array ,
  prefix-delim: string none ,
  reader: function ,
  prefix: string none auto ,
  allowed-prefixes,
) -> function
```

This configuration function should be called as a show rule at the beginning of the document. It enables customized processing of the `ref()` and `cite()` commands.

```typ
#show: alexandria(("books.bib", "papers.bib"), prefix: "x", reader: path =>
read(path))
```

The `reader` parameter can be skipped, in which case the function will expect that `sources` are the raw `bytes` file contents.

Parameters:

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

`prefix-delim` (`string` or `none` = `"-"`) – The delimiter that separates the prefix from the bibliographic key in the `cite()` command

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

`prefix` (`string` or `none` or `auto` = `auto`) – The default prefix to apply to citations that do not specify it explicitly. See `alexandria-prefix()`.

`allowed-prefixes` (= `auto`) – The prefixes that are allowed in the citations. If `auto`, any prefixes are allowed

```
citegroup(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:

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

```
collect-citations(id: string , prefix-filter: string array function auto none ,
style: string bytes ) -> content
```

Assembles a list of citations that reference specified prefixes and prepares the citations and bibliographical references for rendering by `render-bibliography()`.

Parameters:

`id` (`string`) – The unique identifier to assign to the collection.

`prefix-filter` (`string` or `array` or `function` or `auto` or `none` = `auto`) – The filter for the citation prefixes to include into this list. Could be a single prefix, an array of prefixes or a function that gets the prefix and returns true if it has to be included. `auto` filters by the default prefix set by `alexandria-prefix()`, and `none` instructs to accept all prefixes.

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

```
render-bibliography(id: string , filter: none function , title: none content auto )
 -> content
```

Renders the subset of the bibliographical references from the list generated by the collect-citations call.

Parameters:

`id` (`string`) – The unique identifier of the list created by collect-citations

`filter` (`none` or `function` = `none`) – The function to filter the references. It receives a reference as an input and should return true if the reference has to be included. The reference has a `prefixes` field, which is an array of all prefixes that are used in the citations (from the `id` list) referencing this entry. The `details` field contains the actual information about the reference.

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

```
bibliographyx(
  prefix: string auto ,
  title: none content auto ,
  reader: function ,
  style: string bytes ,
) -> content
```

Renders the bibliography associated with a specific prefix. 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",
  title: "Bibliography",
  style: "ieee",
)
```

The function is a simple wrapper that calls `collect-citations()` and `render-bibliography()` to render the bibliographic list for a specific prefix without modifications.

Parameters:

`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.

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

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