Alexandria

Alexandria allows a single document to have multiple bibliographies.

v0.2.2 December 12, 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:0.2.2": *
#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.
- Styling links will only affect individual citations within cite groups (collapsed citations), while cite rules would affect the whole group—see issue #24.
- Native bibliographies have `numbering: none` applied to its title, while *Alexandria*'s haven't. `show bibliography: set heading(...)` also won't work on them.
- Another styling-related difference is that grid-like bibliographies (e.g. IEEE) aren't affected by `show grid` rules in regular Typst, but they are in Alexandria.

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. (n.d.). *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. (n.d.). *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 *collecting* 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:0.2.2": *
#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

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

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

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

Parameters:

`prefix` (`string` = `none`) – a prefix that identifies citations from a specific Alexandria's bibliography.

`read` (`function` = `none`) – the function to process the `path` and `style` parameters of `bibliographyx()` and `load-bibliography()` commands. Pass `path => read(path)` to read the contents of the bibliography and style 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[@x:a @x:b]
```

Only citations, references and spaces may appear in the body. Mixing non-Alexandria references or references from different prefixes in the same citation group is **not supported**.

Parameters:

`prefix` (`string` or `auto` = `auto`) – the optional prefix for the citations within a group. It only needs to be specified if more than one prefix was registered.

`body` (`content`) – the body, containing one or more citations.

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

Loads the bibliography for a given prefix. The function reads the bibliography from the given file(s), which is used later by `get-bibliography()`. It does not render any content. For simple cases, `bibliographyx()` can be used directly.

Even though this function only loads the bibliography, it 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. Some parameters, like `title`, have to be specified when calling `render-bibliography()`, which actually renders the bibliography.

Parameters:

`path` (`string` or `bytes` or `array`) – the path(s) to the bibliography file(s), which is passed to the `read` function registered via `alexandria()`, or its binary contents if no `read` function provided.

`prefix` (`string` or `auto` = `auto`) – the optional prefix for which the bibliography is loaded. It only needs to be specified if more than one prefix was registered.

`full` (`boolean` = `false`) – whether the bibliography for the given prefix should include all bibliographical entries from `path` or only the ones cited in the document.

`style` (`string` or `bytes` = `"ieee"`) – the style of the bibliography. Either a built-in style, a path to a CSL file passed to `read()` registered via `alexandria()`, or its binary contents.

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

Collects all references that have to be rendered. This function is called by `bibliographyx()` and when rendering Alexandria citations. It can also be directly called by the user for more complex use cases. Before calling this function, you must call `load-bibliography()` to load the bibliography data. To actually render the bibliographical list, the result of `get-bibliography()` has to be passed to `render-bibliography()`.

The result is a dictionary with the following keys:

- `prefix`: the unique string prefix that identifies this Alexandria bibliography.
- `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 elements of the `references` array have the following fields:

- `key`: the original bibliography key (without Alexandria's prefix).
- `content`: a Typst representation of the bibliographical entry; used by `render-bibliography()` for rendering bibliographical items.
- optional `first-field`: Typst content for certain bibliography styles. For example, in IEEE style it represents "[1]", "[2]", etc.
- `details`: a dictionary containing information about this reference, including `type`, `title`, `author`, and `date` fields. The 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 optional prefix for which the bibliography should be retrieved. It only needs to be specified if more than one prefix was registered.

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

Renders the given list of bibliographical references. For simple use cases, `bibliographyx()` can be called directly.

You will only need to call this function directly if you want to postprocess the results of `get-bibliography()`, e.g. by filtering out the `references` entries that should appear in another bibliography elsewhere in the document. Note that, to avoid unresolved citations, all references generated by `get-bibliography()` have to appear in some `render-bibliography()` call.

Parameters:

`bib` (`dict`) – the bibliography data prepared by the `get-bibliography()` call.

`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 the bibliography for a given 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",
  prefix: "x:",
  title: "Bibliography",
  full: true,
  style: "ieee",
)
```

This convenience function calls `load-bibliography()`, `get-bibliography()`, and `render-bibliography()` to reproduce the behavior of the built-in `bibliography()` call.

Parameters:

`path` (`string` or `bytes` or `array`) – the path(s) to the bibliography file(s), which is passed to the `read` function registered via `alexandria()`, or its binary contents if no `read` function provided.

`prefix` (`string` or `auto` = `auto`) – the optional prefix for which the bibliography is generated. It only needs to be specified if more than one prefix was registered.

`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 all bibliographical entries from `path` or only the ones cited in the document.

`style` (`string` or `bytes` = `"ieee"`) – the style of the bibliography. Either a built-in style, a path to a CSL file passed to `read` registered via `alexandria()`, or its binary contents.