Alexandria

v0.1.3 March 13, 2025

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

Clemens Koza

Abstract

Alexandria allows a single document to have multiple bibliographies.

Contents

I Introduction

Alexandria allows adding multiple bibliographies to the same document. Its two main functions are `alexandria()` and `bibliographyx()`. Typical usage would look something like this:

```typ
#import "@preview/alexandria:0.1.3": *
#show: alexandria(prefix: "x-", read: path => read(path))

...

#bibliographyx(
  "bibliography.bib",
  // title: auto is not yet supported so it needs to be specified
  title: "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.
- Adjacent citations aren't collapsed.
- Citations that are shown as footnotes are not supported yet.

The example on the next page demonstrates some of these. If you find additional limitations or other issues, please report them at https://github.com/SillyFreak/typst-alexandria/issues.

II Example – native Typst version (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.

III Example – Alexandria version (APA)

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

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

III.a 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.

IV Example – Alexandria version (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.

IV.a 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.

[5] R. Hock, “Glacier melt: a review of processes and their modelling,” *Progress in Physical Geography: Earth and Environment*, vol. 29, no. 3, pp. 362–391, 2005, doi: 10.1191/0309133305pp453ra.

[6] J. R. R. Tolkien, *The Fellowship of the Ring*, vol. 1. in The Lord of the Rings, vol. 1. London: Allen & Unwin, 1954.

[7] J. Kopp, R. Gerike, and K. W. Axhausen, “Do sharing people behave differently? An empirical evaluation of the distinctive mobility patterns of free-floating car-sharing members,” *Transportation*, vol. 42, no. 3, pp. 449–469, 2015.

[8] L. Richardson and S. Ruby, *RESTful Web Services*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2008.

[9] I. B. McIntosh, V. Swanson, K. G. Power, F. Raeside, and C. Dempster, “Anxiety and Health Problems Related to Air Travel,” *Journal of Travel Medicine*, vol. 5, no. 4, pp. 198–204, 2006.

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

V Splitting bibliographies

The previous three examples showed using Alexandria to render three separate bibliographies for different parts of a document: Example II used the native bibliography, Example III used Alexandria to show APA style references, and Example IV showed IEEE style. Particularly, with IEEE, all references are numbered and multiple separate Alexandria bibliographies would reuse the same 1-based numbering.

This approach is thus not suitable for multiple bibliographies that serve the same regions of a document. For this purpose, Alexandria also supports splitting the *loading* and *rendering* of a bibliography, giving you the opportunity to preprocess the bibliography entries. Instead of calling `bibliographyx()` directly, you'd use `load-bibliography()` followed by `get-bibliography()` and `render-bibliography()`.

An example could look like this:

```typ
#import "@preview/alexandria:0.1.3": *
#show: alexandria(prefix: "x-", read: path => read(path))

...

// load the bibliography so that the data is available to citations and rendering
#load-bibliography("bibliography.bib")

#context {
  // get the bibliography items
  let (references, ..rest) =  get-bibliography("x-")

  // render the bibliography
  render-bibliography(
    title: [Bibliography],
    (
      // instead of giving it all references, only consider non-book references
      references: references.filter(x => x.details.type != "book"),
      // `render-bibliography()` also needs the non-reference information
      // that was returned by `get-bibliography()`
      ..rest,
    ),
  )

  // render the rest of the bibliography
  // (this could also be somewhere else in the document)
  render-bibliography(
    title: [Books],
    (
      references: references.filter(x => x.details.type == "book"),
      ..rest,
    ),
  )
}
```

VI Example – Splitting a bibliography

Here is a rendered example of using this approach. You can see how the single call to `load-bibliography()` results in the entries using distinct numbers.

Note how all references are rendered once, although in a different presentation from usual. This is generally a requirement for citations being able to refer to their corresponding reference's label. In this particular case, this is not a concern since there are no citations and references were rendered due to the `full` option, but in general this is a concern.

VI.a Bibliography

[1] 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.

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

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

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

[5] R. Hock, "Glacier melt: a review of processes and their modelling," *Progress in Physical Geography: Earth and Environment*, vol. 29, no. 3, pp. 362–391, 2005, doi: 10.1191/0309133305pp453ra.

[7] J. Kopp, R. Gerike, and K. W. Axhausen, "Do sharing people behave differently? An empirical evaluation of the distinctive mobility patterns of free-floating car-sharing members," *Transportation*, vol. 42, no. 3, pp. 449–469, 2015.

[9] I. B. McIntosh, V. Swanson, K. G. Power, F. Raeside, and C. Dempster, "Anxiety and Health Problems Related to Air Travel," *Journal of Travel Medicine*, vol. 5, no. 4, pp. 198–204, 2006.

VI.b Books

[6] J. R. R. Tolkien, *The Fellowship of the Ring*, vol. 1. in The Lord of the Rings, vol. 1. London: Allen & Unwin, 1954.

[8] L. Richardson and S. Ruby, *RESTful Web Services*, 1st ed. Sebastopol, CA, USA: O'Reilly Media, 2008.

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

VII Module reference

VII.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().

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

The `read` parameter kann 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.

```
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",
  title: "Bibliography",
  full: true,
  style: "ieee",
)
```

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.