The Hayagriva YAML File Format

The Hayagriva YAML file format enables you to feed a collection of literature items into Hayagriva. It is built on the YAML standard. This documentation starts with a basic introduction with examples into the format, explains how to represent several types of literature with parents, and then explores all the possible fields and data types. An example file covering many potential use cases can be found in the test directory of the repository.

Overview

In technical terms, a Hayagriva file is a YAML document that contains a single mapping of mappings. Or, in simpler terms: Every literature item needs to be identifiable by some name (the *key*) and have some properties that describe it (the *fields*). Suppose a file like this:

```yaml
harry:
    type: Book
    title: Harry Potter and the Order of the Phoenix
    author: Rowling, J. K.
    volume: 5
    page-total: 768
    date: 2003-06-21

electronic:
    type: Web
    title: Ishkur's Guide to Electronic Music
    serial-number: v2.5
    author: Ishkur
    url: http://www.techno.org/electronic-music-guide/
```

You can see that it refers to two items: The fifth volume of the Harry Potter books (key: `harry`) and a web page called "Ishkur's Guide to Electronic Music" (key: `electronic`). The key always comes first and is followed by a colon. Below the key, indented, you can find one field on each line: They start with the field name, then a colon, and then the field value.

Sometimes, this value can be more complex than just some text after the colon. If you have an article that was authored by multiple people, its `author` field can look like this instead:

```yaml
author: ["Omarova, Saule", "Steele, Graham"]
```

Or it could also be this:

```yaml
author:
    - Omarova, Saule
    - Steele, Graham
```

The `author` field can be an *array* (a list of values) to account for media with more than one creator. YAML has two ways to represent these lists: The former, compact, way where you wrap your list in square braces and the latter, more verbose way where you put each author on their own indented line and precede them with a hyphen so that it looks like a bullet list. Since in the compact form both list items and authors' first and last names are separated by commas you have to wrap the names of individual authors in double–quotes.

Sometimes, fields accept composite data. If, for example, you would want to save an access date for an URL for your bibliography, you would need the `url` field to accept that. This is accomplished like this:

```
url:
    value: http://www.techno.org/electronic-music-guide/
    date: 2020-11-30
```

There is also a more compact form of this that might look familiar if you know JSON:

```
url: { value: http://www.techno.org/electronic-music-guide/, date: 2020-11-30 }
```

By now, you must surely think that there must be an abundance of fields to represent all the possible information that could be attached to any piece of literature: For example, an article could have been published in an anthology whose title you would want to save, and that anthology belongs to a series that has a title itself... For this, you would already need three different title–fields? Hayagriva's data model was engineered to prevent this kind of field bloat, read the next section to learn how to represent various literature.

Representing publication circumstance with parents

Hayagriva aims to keep the number of fields it uses small to make the format easier to memorize and therefore write without consulting the documentation. Other contemporary literature management file formats like RIS and BibLaTeX use many fields to account for every kind of information that could be attached to some piece of media.

We instead use the concept of parents: Many pieces of literature are published within other media (e. g. articles can appear in newspapers, blogs, periodicals, ...), and when each of these items is regarded isolatedly and without consideration for that publication hierarchy, there are substantially fewer fields that could apply.

How does this look in practice? An article in a scientific journal could look like this:

```
kinetics:
    type: Article
    title: Kinetics and luminescence of the excitations of a nonequilibrium polarito
    author: ["Doan, T. D.", "Tran Thoai, D. B.", "Haug, Hartmut"]
    serial-number:
        doi: "10.1103/PhysRevB.102.165126"
    page-range: 165126-165139
    date: 2020-10-14
    parent:
        type: Periodical
        title: Physical Review B
        volume: 102
        issue: 16
        publisher: American Physical Society
```

This means that the article was published in issue 16, volume 102 of the journal "Physical Review B". Notice that the `title` field is in use for both the article and its parent – every field is available for both top–level use and all parents.

To specify parent information, write the `parent` field name and a colon and then put all fields for that parent on indented lines below.

The line `type: Periodical` could also have been omitted since each entry type has the notion of a default parent type, which is the type that the parents will have if they do not have a `type` field.

Sometimes, media is published in multiple ways, i. e. one parent would not provide the full picture. Multiple parents are possible to deal with these cases:

```
wwdc-network:
    type: Article
    author: ["Mehta, Jiten", "Kinnear, Eric"]
    title: Boost Performance and Security with Modern Networking
    date: 2020-06-26
    parent:
        - type: Conference
          title: World Wide Developer Conference 2020
          organization: Apple Inc.
          location: Mountain View, CA
        - type: Video
          runtime: "00:13:42"
          url: https://developer.apple.com/videos/play/wwdc2020/10111/
```

This entry describes a talk presented at a conference and for which a video is available from which the information was ultimately cited.

Just like the `author` field, `parents` can be a list. If it does, a hyphen indicates the start of a new parent.

Parents can also appear as standalone items and can have parents themselves. This is useful if you are working with articles from a journal that belongs to a series or cases like the one below:

```
plaque:
    type: Misc
    title: Informational plaque about Jacoby's 1967 photos
    publisher:
        name: Stiftung Reinbeckhallen
        location: Berlin, Germany
    date: 2020
    parent:
        type: Artwork
        date: 1967
        author: Jacoby, Max
        parent:
            type: Anthology
            title: Bleibtreustraße
            archive: Landesmuseum Koblenz
            archive-location: Koblenz, Germany
```

This plaque was created by a museum for a photo by Jacoby that belongs to a series that is usually archived at a different museum.

Reference

This section lists all possible fields and data types for them.

Fields

`type`

Data type:	entry type
Description:	media type of the item, often determines the structure of references.
Example:	`type: video`

`title`

Data type:	formattable string
Description:	title of the item
Example:	`title: "Rick Astley: How An Internet Joke Revived My Career"`

`author`

Data type:	person / list of persons
Description:	persons primarily responsible for the creation of the item
Example:	`author: ["Klocke, Iny", "Wohlrath, Elmar"]`

`date`

Data type:	date
Description:	date at which the item was published
Example:	`date: 1949-05`

`parent`

Data type:	entry
Description:	item in which the item was published / to which it is strongly associated to
Example:	

`abstract`

Data type:	formattable string
Description:	Abstract of the item (e.g. the abstract of a journal article).
Example:	`abstract: The dominant sequence transduction models are based on complex...`

`genre`

Data type:	formattable string
Description:	Type, class, or subtype of the item (e.g. "Doctoral dissertation" for a PhD thesis; "NIH Publication" for an NIH technical report). Do not use for topical descriptions or categories (e.g. "adventure" for an adventure movie).
Example:	`genre: Doctoral dissertation`

`editor`

Data type:	person / list of persons
Description:	persons responsible for selecting and revising the content of the item
Example:	

affiliated

Data type:	list of persons with role / list of lists of persons with role
Description:	persons involved with the item that do not fit `author` or `editor`
Example:	

call-number

Data type:	formattable string
Description:	The number of the item in a library, institution, or collection. Use with `archive`.
Example:	`call-number: "F16 D14"`

publisher

Data type:	publisher
Description:	publisher of the item
Example:	

location

Data type:	formattable string
Description:	location at which an entry is physically located or took place. For the location where an item was published, see `publisher`.
Example:	`location: Lahore, Pakistan`

organization

Data type:	formattable string
Description:	Organization at/for which the item was produced
Example:	`organization: Technische Universität Berlin`

issue

Data type:	numeric or string
Description:	For an item whose parent has multiple issues, indicates the position in the issue sequence. Also used to indicate the episode number for TV.
Example:	`issue: 5`

volume

Data type:	numeric or string
Description:	For an item whose parent has multiple volumes/parts/seasons ... of which this item is one
Example:	`volume: 2-3`

volume-total

Data type:	numeric
Description:	Total number of volumes/parts/seasons this item consists of
Example:	`volume-total: 12`

`edition`

Data type:	numeric or string
Description:	published version of an item
Example:	`edition: expanded and revised edition`

`page-range`

Data type:	numeric or string
Description:	the range of pages within the parent this item occupies
Example:	`page-range: 812-847`

`page-total`

Data type:	numeric
Description:	total number of pages the item has
Example:	`page-total: 1103`

`time-range`

Data type:	timestamp range
Description:	the time range within the parent this item starts and ends at
Example:	`time-range: 00:57-06:21`

`runtime`

Data type:	timestamp
Description:	total runtime of the item
Example:	`runtime: 01:42:21,802`

`url`

Data type:	url
Description:	canonical public URL of the item, can have access date
Example:	`url: { value: https://www.reddit.com/r/Acci-dentalRenaissance/comments/er1uxd/japan-ese_opposition_members_trying_to_block_the/, date: 2020-12-29 }`

`serial-number`

Data type:	string or dictionary of strings
Description:	Any serial number, including article numbers. If you have serial numbers of well–known schemes like `doi`, you should put them into the serial number as a dictionary like in the second example. Hayagriva will recognize and specially treat `doi`, `isbn` `issn`, `pmid`, `pmcid`, and `arxiv`. You can also include `serial` for the serial number when you provide other formats as well.
Example:	`serial-number: 2003.13722` or

`language`

Data type:	unicode language identifier
Description:	language of the item

Example:	`language: zh-Hans`

`archive`

Data type:	formattable string
Description:	name of the institution/collection where the item is kept
Example:	`archive: National Library of New Zealand`

`archive-location`

Data type:	formattable string
Description:	location of the institution/collection where the item is kept
Example:	`archive-location: Wellington, New Zealand`

`note`

Data type:	formattable string
Description:	short markup, decoration, or annotation to the item (e.g., to indicate items included in a review).
Example:	`microfilm version`

Data types

Entry

Entries are collections of fields that could either have a key or be contained in the `parent` field of another entry.

Entry Type

Needs a keyword with one of the following values:

- `article`. A short text, possibly of journalistic or scientific nature, appearing in some greater publication (default parent: `periodical`).

- `chapter`. A section of a greater containing work (default parent: `book`).

- `entry`. A short segment of media on some subject matter. Could appear in a work of reference or a data set (default parent: `reference`).

- `anthos`. Text published within an Anthology (default parent: `anthology`).

- `report`. A document compiled by authors that may be affiliated to an organization. Presents information for a specific audience or purpose.

- `thesis`. Scholarly work delivered to fulfill degree requirements at a higher education institution.

- `web`. Piece of content that can be found on the internet and is native to the medium, like an animation, a web app, or a form of content not found elsewhere. Do not use this entry type when referencing a textual blog article, instead use an `article` with a `blog` parent (default parent: `web`).

- `scene`. A part of a show or another type of performed media, typically all taking place in the same location (default parent: `video`).

- `artwork`. A form of artistic/creative expression (default parent: `exhibition`).

- `patent`. A technical document deposited at a government agency that describes an invention to legally limit the rights of reproduction to the inventors.

- `case`. Reference to a legal case that was or is to be heard at a court of law.
- `newspaper`. The issue of a newspaper that was published on a given day.
- `legislation`. Legal document or draft thereof that is, is to be, or was to be enacted into binding law (default parent: `anthology`).
- `manuscript`. Written document that is submitted as a candidate for publication.
- `original`. The original container of the entry before it was re–published.
- `post`. A post on a micro–blogging platform like Twitter (default parent: `post`).
- `misc`. Items that do not match any of the other Entry type composites.
- `performance`. A live artistic performance.
- `periodical`. A publication that periodically publishes issues with unique content. This includes scientific journals and news magazines.
- `proceedings`. The official published record of the events at a professional conference.
- `book`. Long–form work published physically as a set of bound sheets.
- `blog`. Set of self–published articles on a website.
- `reference`. A work of reference. This could be a manual or a dictionary.
- `conference`. Professional conference. This Entry type implies that the item referenced has been an event at the conference itself. If you instead want to reference a paper published in the published proceedings of the conference, use an `article` with a `proceedings` parent.
- `anthology`. Collection of different texts on a single topic/theme.
- `repository`. Publicly visible storage of the source code for a particular software, papers, or other data and its modifications over time.
- `thread`. Written discussion on the internet triggered by an original post. Could be on a forum, social network, or Q&A site.
- `video`. Motion picture of any form, possibly with accompanying audio (default parent: `video`).
- `audio`. Recorded audible sound of any kind (default parent: `audio`).
- `exhibition`. A curated set of artworks.

The field is case insensitive. It defaults to `Misc` or the default parent if the entry appears as a parent of an entry that defines a default parent.

Formattable String

A formattable string is a string that may run through a text case transformer when used in a reference or citation. You can disable these transformations on segments of the string or the whole string.

The simplest scenario for a formattable string is to provide a string that can be case–folded:

```
publisher: UN World Food Programme
```

If you want to preserve a part of the string but want to go with the style's behavior otherwise, enclose the string in braces like below. You must wrap the whole string in quotes if you do this.

```
publisher: "{imagiNary} Publishing"
```

To disable formatting altogether and instead preserve the casing as it appears in the source string, put the string in the `value` sub–field and specify another sub–field as `verbatim: true`:

```
publisher:
    value: UN World Food Programme
    verbatim: true
```

Title and sentence case folding will always be deactivated if your item has set the `language` key to something other than English.

You can also include mathematical markup evaluated by Typst by wrapping it in dollars.

Furthermore, every formattable string can include a short form that a citation style can choose to render over the longer form.

```
journal:
    value: International Proceedings of Customs
    short: Int. Proc. Customs
```

Person

A person consists of a name and optionally, a given name, a prefix, and a suffix for the (family) name as well as an alias. Usually, you specify a person as a string with the prefix and the last name first, then a comma, followed by a given name, another comma, and then finally the suffix. Following items are valid persons:

- `Doe, Janet`
- `Luther King, Martin, Jr.`
- `UNICEF`
- `von der Leyen, Ursula`

The prefix and the last name will be separated automatically using the same algorithm as BibTeX (p. 24) which can be summarized as "put all the consecutive lower case words at the start into the prefix."

Usually, this is all you need to specify a person's name. However, if a part of a name contains a comma, the prefix is not lowercased, or if one needs to specify an alias, the person can also be specified using sub–fields:

```
author:
    given-name: Gloria Jean
    name: Watkins
    alias: bell hooks
```

The available sub–fields are `name`, `given-name`, `prefix`, `suffix`, and `alias`. The `name` field is required.

List of persons with role

This data type requires a mapping with two fields: `names` which contains a list of persons or a single person and a `role` which specifies their role with the item:

```
role: ExecutiveProducer
names: ["Simon, David", "Colesberry, Robert F.", "Noble, Nina Kostroff"]
```

Possible `role` values

- `translator`. Translated the work from a foreign language to the cited edition.
- `afterword`. Authored an afterword.
- `foreword`. Authored a foreword.
- `introduction`. Authored an introduction.
- `annotator`. Provided value–adding annotations.
- `commentator`. Commented on the work.
- `holder`. Holds a patent or similar.
- `compiler`. Compiled the works in an Anthology.
- `founder`. Founded the publication.
- `collaborator`. Collaborated on the cited item.
- `organizer`. Organized the creation of the cited item.
- `cast-member`. Performed in the cited item.
- `composer`. Composed all or parts of the cited item's musical/audible components.
- `producer`. Produced the cited item.
- `executive-producer`. Lead Producer for the cited item.
- `writer`. Did the writing for the cited item.
- `cinematography`. Shot film/video for the cited item.
- `director`. Directed the cited item.
- `illustrator`. Illustrated the cited item.
- `narrator`. Provided narration or voice–over for the cited item.

The `role` field is case insensitive.

Date

A calendar date as ISO 8601. This means that you specify the full date as `YYYY-MM-DD` with an optional sign in front to represent years earlier than `0000` in the Gregorian calendar. The year 1 B.C.E. is represented as `0000`, the year 2 B.C.E. as `-0001` and so forth.

The shortened forms `YYYY` or `YYYY-MM` are also possible.

Timestamp

A timestamp represents some time in a piece of media. It is given as a string of the form `DD:HH:MM:SS,msms` but everything except `MM:SS` can be omitted. Wrapping the string in double–quotes is necessary due to the colons.

The left–most time denomination only allows values that could overflow into the next–largest denomination if that is not specified. This means that the timestamp `138:00` is allowed for 2 hours and 18 minutes, but `01:78:00` is not.

Timestamp range

A range of timestamps is a string containing two timestamps separated by a hyphen. The first timestamp in the string indicates the starting point, whereas the second one indicates the end. Wrapping the string in double–quotes is necessary due to the colons in

the timestamps.

```
time-range: "03:35:21-03:58:46"
```

String

Strings are sequences of characters as a field value. In most cases you can write your string after the colon, but if it contains a special character (`:`, `{`, `}`, `[`, `]`, `,`, `&`, `*`, `#`, `?`, `|`, `-`, `<`, `>`, `=`, `!`, `%`, `@`, `\`) it should be wrapped with double–quotes. If your string contains double–quotes, you can write those as this escape sequence: `\"`. If you instead wrap your string in single quotes, most YAML escape sequences such as `\n` for a line break will be ignored.

Numeric

Numeric variables are one or more numbers that are delimited by commas, ampersands, and hyphens. Numeric variables can express a single number or a range and contain only integers, but may contain negative numbers. Numeric variables can have a non–numeric prefix and suffix.

```
page-range: S10-15
```

Unicode Language Identifier

A Unicode Language Identifier identifies a language or its variants. At the simplest, you can specify an all–lowercase two–letter ISO 639–1 code like `en` or `es` as a language. It is possible to specify regions, scripts, or variants to more precisely identify a variety of a language, especially in cases where the ISO 639–1 code is considered a "macrolanguage" (`zh` includes both Cantonese and Mandarin). In such cases, specify values like `en-US` for American English or `zh-Hans-CN` for Mandarin written in simplified script in mainland China. The region tags have to be written in all–caps and are mostly corresponding to ISO 3166–1 alpha_2 codes.

Consult the documentation of the Rust crate unic–langid we use for parsing these language identifiers for more information.