`glossarium@0.5.9`

This document outline how to change the default behaviour of `glossarium` by implementing “user functions”. It is recommended to keep the default implementation and not to change the default behaviour of the package.

If you have a need that require to change the defaults, you are expected to be knowledgeable in writing complex typst code and to try to debug your issues first on you own. Be aware that helping regular users and fixing bugs will take priority over helping you debug your own implementation of `glossarium` internal functions.

Contents

1. Customization

This section shows how to change the default behaviour of `glossarium` by implementing user functions. It is recommended to keep to the user available interface and not to change the default behaviour of the package. If you have any suggestions or need help, please open an issue on the GitHub repository.

There are effectively two requirements for a user to use `glossarium`:
1. Write a `show: make-glossary` rule to transform all `@key` and `@key:pl` into `#gls(key)` or `#glspl(key)`
2. Call `print-glossary(entry-list)` somewhere in order to write all labels

The `glossarium` package provides a default behaviour to `print-glossary`. In ascending order, all implemented behaviours are:

1. `default-print-back-references(entry) -> contextual content`
2. `default-print-description(entry) -> content`
3. `default-print-title(entry) -> content`
4. `default-print-gloss(entry, ...) -> contextual content`
5. `default-print-reference(entry, ...) -> contextual content`
6. `default-print-glossary(entries, groups, ...) -> contextual content`

gloss A **gloss** is a single entry in the glossary. It is composed of a **title**, a **description**, and a list of **back references**.

reference A **reference** is the construct `glossarium` uses to manage the glossary. Internally, it is constructed using `figure()` and `label()`.

glossary A **glossary** is the list of all glosses. It is composed of a list of **entries** and a list of **groups**. The default **group** is `""`.

Each of these functions can be replaced by a user-defined function with the same signature. Then, the user can pass them directly to `print-glossary()`, e.g.,

```
let my-print-description(entry) = {
  let ent-description = entry.at("description", default: "")
  return text.with(size: 9pt)[#ent-description]
}
print-glossary(entry-list, user-print-description: my-print-description)
```

Keep in mind that some options are available from the start:

- `show-all (bool)` : show all entries, even if they are not referenced in the document
- `disable-back-references (bool)`: do not show back references
- `user-group-break (function: () => content)`: a function to call between groups

1.1. Default functions

The functions are listed in order of reverse call hierarchy:

1. `print-glossary(entry-list)`
    1. `default-print-glossary(entries, groups)`
        1. `default-print-reference(entry)`
            1. `default-print-gloss(entry)`
                1. `default-print-title(entry)`
                2. `default-print-description(entry)`
                3. `default-print-back-references(entry)`
        2. `default-group-break()`

⚠ Although `default-print-reference` is available to the user, it is not recommended to modify this function.

The full signatures for `default-print-gloss()`, `default-print-reference()`, and `default-print-glossary()` in the next sections.

1.1.1. Arguments

The functions take the following arguments:

1. `entries`: the normalized entry list
2. `groups`: the list of groups
3. `entry`: an entry

This is one `entry`:

```
(
  key: "WHO",
  short: "WHO",
  artshort: "a",
  plural: none,
  long: "World Health Organization",
  artlong: "a",
  longplural: none,
  description: "Lorem ipsum dolor sit amet.",
  group: "Organizations",
  sort: "WHO",
  styles: none,
  custom: none,
)
```

`groups`

```
("Organizations", "", "Countries")
```

`entries`

```
(
  (
    key: "WHO",
    short: "WHO",
    artshort: "a",
    plural: none,
    long: "World Health Organization",
    artlong: "a",
    longplural: none,
    description: "Lorem ipsum dolor sit amet.",
    group: "Organizations",
    sort: "WHO",
    styles: none,
    custom: none,
  ),
  (
    key: "WTO",
    short: "WTO",
    artshort: "a",
    plural: none,
    long: "World Trade Organization",
    artlong: "a",
    longplural: none,
    description: "Lorem ipsum dolor sit amet.",
    group: "Organizations",
    sort: "WTO",
    styles: none,
    custom: none,
  ),
)
```

1.1.2. `default-print-back-references(entry)`

The default implementation is:

```
#let default-print-back-references(entry) = {
  return get-entry-back-references(entry).join(", ")
}
```

Without going into details, assume that `get-entry-back-references(entry)` returns a list of back references. The function simply joins them with a comma.

For example, for World Health Organization (WHO), the back references are:

```
3, 5, 6
```

The value of `get-entry-back-references(entry)` is:

```
(
  link(dest: .., body: [3]),
  link(dest: .., body: [5]),
  link(dest: .., body: [6]),
)
```

where `dest` contains a `location`.

1.1.3. `default-print-description(entry)`

The default implementation is:

```
#let default-print-description(entry) = {
  return entry.at("description")
}
```

1.1.4. `default-print-title(entry)`

The default implementation is:

```
#let default-print-title(entry) = {
  let caption = []
  let txt = text.with(weight: 600)

  if has-long(entry) {
    caption += txt(emph(entry.short) + [ -- ] + entry.long)
  } else {
    caption += txt(emph(entry.short))
  }
  return caption
}
```

The function is fairly simple to understand:

- If the entry has a long description, it returns `text.with(weight: 600)[emph(entry.short) -- entry.long]`
- If the entry does not have a long description, it returns `text.with(weight: 600)[emph(entry.short)]`

1.1.5. `default-print-gloss(entry)`

The default implementation is

```
#let default-print-gloss(
  entry,
  show-all: false,
  disable-back-references: false,
  user-print-title: default-print-title,
  user-print-description: default-print-description,
  user-print-back-references: default-print-back-references,
) = context {
  let caption = []

  if show-all == true or count-refs(entry) != 0 {
    // Title
    caption += user-print-title(entry)

    // Description
    if has-description(entry) {
      // Title - Description separator
      caption += ": "

      caption += user-print-description(entry)
    }

    // Back references
    if disable-back-references != true {
      caption += " "

      caption += user-print-back-references(entry)
    }
  }

  return caption
}
```

- `default-print-gloss` is responsible for printing separators between the title, description, and back references.
- It also checks if the entry should be printed or not. If the entry is not referenced in the document, it will not be printed unless `show-all:`
  `true`.
- If back references are disabled (`disable-back-references: true`), they will not be printed.

The default behaviour will be displayed as such:

***WHO* – World Health Organization**: Lorem ipsum dolor sit amet. 3, 5, 6

Without back references:

***WHO* – World Health Organization**: Lorem ipsum dolor sit amet.

For a non-referenced entry:

show-all: false =>

show-all: true =>

***IMF* – International Monetary Fund**: Lorem ipsum dolor sit amet.

One important utility function is `count-refs`. For WHO, `#context count-refs(entry)` is

3

1.1.6. `default-print-reference(entry)`

⚠ There are few reasons to modify this function. It is recommended to keep the default behaviour, but an override option is provided for advanced users.

The default implementation is:

```
#let default-print-reference(
  entry,
  show-all: false,
  disable-back-references: false,
  user-print-gloss: default-print-gloss,
  user-print-title: default-print-title,
  user-print-description: default-print-description,
  user-print-back-references: default-print-back-references,
) = {
  return [
    #show figure.where(kind: __glossarium_figure): it => it.caption
    #par(
      hanging-indent: 1em,
      first-line-indent: 0em,
    )[
      #figure(
        supplement: "",
        kind: __glossarium_figure,
        numbering: none,
        caption: user-print-gloss(
          entry,
          show-all: show-all,
          disable-back-references: disable-back-references,
          user-print-title: user-print-title,
          user-print-description: user-print-description,
          user-print-back-references: user-print-back-references,
        ),
      )[] #label(entry.key)
      // Line below can be removed safely
      #figure(kind: __glossarium_figure, supplement: "")[] #label(entry.key + ":pl")
    ]
    #parbreak()
  ]
}
```

The function is responsible for creating the referenceable element and the label for the gloss.

- it uses `figure()` and `label()` to create the element and make it referenceable. `glossarium` uses `kind: __glossarium_figure` to uniquely identify glossary figures
- the figure's caption is the result of `user-print-gloss()` (see previous section Section 1.1.5)
- By default, an additional empty figure with label `key:pl` is created. This is useful for referencing plural forms of the glossary entry, e.g., `@WHO:pl`=WHOs. It can be safely removed.

⚠ Notice that `entry.key` is used as the label. This usage implies that glosses with duplicate keys will not work, as labels must be unique.

The code below panics:

```
#figure(caption: "test")[]#label("a")
#figure(caption: "test")[]#label("a")
@a
```

with error

```
error: label `<a>` occurs multiple times in the document
```

1.1.7. `default-print-glossary(entries, groups)`

```
#let default-print-glossary(
  entries,
  groups,
  show-all: false,
  disable-back-references: false,
  user-print-reference: default-print-reference,
  user-group-break: default-group-break,
  user-print-gloss: default-print-gloss,
  user-print-title: default-print-title,
  user-print-description: default-print-description,
  user-print-back-references: default-print-back-references,
) = {
  let body = []
  let previous-heading = query(selector(heading).before(here())).last()

  for group in groups.sorted() {
    let group-entries = entries.filter(x => x.at("group") == group)
    let group-ref-counts = group-entries.map(count-refs)
    let print-group = (
      group != ""
      and (
        show-all == true
        or group-ref-counts.any(x => x > 0)
      )
    )

    // Only print group name if any entries are referenced
    if print-group {
      body += [#heading(group, level: previous-heading.level + 1)]
    }
    for entry in group-entries.sorted(key: x => x.key) {
      body += user-print-reference(
        entry,
        show-all: show-all,
        disable-back-references: disable-back-references,
        user-print-gloss: user-print-gloss,
        user-print-title: user-print-title,
        user-print-description: user-print-description,
        user-print-back-references: user-print-back-references,
      )
    }

    body += user-group-break()
  }

  return body
}
```

The function is responsible for printing the glossary. It iterates over all groups and prints the entries. It also checks if the group should be printed or not. If the group is empty, it will not be printed unless `show-all:`
`true`.

See the default style in Section 1.2.1.

1.2. Styles

1.2.1. Default glossary

2. Glossary

EU **– European Union**: Lorem ipsum dolor sit amet.

IMF **– International Monetary Fund**: Lorem ipsum dolor sit amet.

UN **– United Nations**: Lorem ipsum dolor sit amet.

2.1. Countries

UAE **– United Arab Emirates**: Lorem ipsum dolor sit amet.

UK **– United Kingdom**: Lorem ipsum dolor sit amet.

USA **– United States of America**: Lorem ipsum dolor sit amet.

2.2. Organizations

WHO **– World Health Organization**: Lorem ipsum dolor sit amet. 3, 5, 6

WTO **– World Trade Organization**: Lorem ipsum dolor sit amet.