

# Typst Compiler Version: 0.13.1, version

## Normalization

### entry-list-0

```
(
  (
    key: "potato",
    short: "potato",
    long: "long potato",
    plural: "potatoes",
    longplural: "long potatoes",
    description: [Lorem ipsum dolor sit amet.],
  ),
),
(
  (
    key: "potato F",
    short: "potato@F",
    long: "long potato@F",
    longplural: "long potatoes@F",
    description: [This potato should not appear in the
document.],
    group: "F",
  ),
),
)
```

### entry-list-1

```
(
  (
    key: "potato A",
    short: "potato@A",
    plural: "potatoes@A",
    description: [Lorem ipsum dolor sit amet.],
    group: "A",
  ),
),
(
  (
    key: "potato B",
    short: "potato@B",
    long: "long potato@B",
    plural: "potatoes@B",
    longplural: "long potatoes@B",
    description: [Lorem ipsum dolor sit amet.],
    group: "B",
  ),
),
(
  (
    key: "hidden potato",
    short: "hiddenpotato",
    group: "show-all test",
    description: sequence(
      [This potato does not appear, unless],
      [ ],
      raw(text: "print-glossary", block: false),
      [ ],
      [is called with],
      [ ],
      raw(text: "show-all: true", block: false),
    ),
  ),
),
)
```

### entry-list-2

```
(
  (
    key: "potato C",
    short: "potato@C",
    long: "long potato@C",
    plural: "potatoes@C",
    longplural: "long potatoes@C",
    description: [Lorem ipsum dolor sit amet.],
    group: "C",
  ),
),
(
  (
    key: "potato D",
    short: "potato@D",
    long: "long potato@D",
    plural: "potatoes@D",
    description: [Lorem ipsum dolor sit amet.],
    group: "D",
  ),
),
(
  (
    key: "potato E",
    short: "potato@E",
    long: "long potato@E",
    longplural: "long potatoes@E",
    description: [Lorem ipsum dolor sit amet.],
    group: "E",
  ),
),
)
```

## \_\_normalize\_entry\_list n°0

```
(
  (
    key: "potato",
    short: "potato",
    artshort: "a",
    plural: "potatoes",
    long: "long potato",
    artlong: "a",
    longplural: "long potatoes",
    description: [Lorem ipsum dolor sit amet.],
    group: "",
    sort: "potato",
  ),
)
```

## \_\_normalize\_entry\_list n°1

```
(
  (
    key: "potato A",
    short: "potato@A",
    artshort: "a",
    plural: "potatoes@A",
    long: none,
    artlong: "a",
    longplural: none,
    description: [Lorem ipsum dolor sit amet.],
    group: "A",
    sort: "potato A",
  ),
  (
    key: "potato B",
    short: "potato@B",
    artshort: "a",
    plural: "potatoes@B",
    long: "long potato@B",
    artlong: "a",
    longplural: "long potatoes@B",
    description: [Lorem ipsum dolor sit amet.],
    group: "B",
    sort: "potato B",
  ),
  (
    key: "hidden potato",
    short: "hiddenpotato",
    artshort: "a",
    plural: none,
    long: none,
    artlong: "a",
    longplural: none,
    description: sequence(
      [This potato does not appear, unless],
      [ ],
      raw(text: "print-glossary", block: false),
      [ ],
      [is called with],
      [ ],
      raw(text: "show-all: true", block: false),
    ),
    group: "show-all test",
    sort: "hidden potato",
  ),
)
```

## \_\_normalize\_entry\_list n°2

```
(
  (
    key: "potato C",
    short: "potato@C",
    artshort: "a",
    plural: "potatoes@C",
    long: "long potato@C",
    artlong: "a",
    longplural: "long potatoes@C",
    description: [Lorem ipsum dolor sit amet.],
    group: "C",
    sort: "potato C",
  ),
)
```

```
),
(
  key: "potato D",
  short: "potato@D",
  artshort: "a",
  plural: "potatoes@D",
  long: "long potato@D",
  artlong: "a",
  longplural: none,
  description: [Lorem ipsum dolor sit amet.],
  group: "D",
  sort: "potato D",
),
(
  key: "potato E",
  short: "potato@E",
  artshort: "a",
  plural: none,
  long: "long potato@E",
  artlong: "a",
  longplural: "long potatoes@E",
  description: [Lorem ipsum dolor sit amet.],
  group: "E",
  sort: "potato E",
),
(
  key: "potato F",
  short: "potato@F",
  artshort: "a",
  plural: none,
  long: "long potato@F",
  artlong: "a",
  longplural: "long potatoes@F",
  description: [This potato should not appear in the
document.],
  group: "F",
  sort: "potato F",
),
)
```

## User API

### ref

@potato: long potato (potato)

@potato:pl: potatoes

### gls

First: potato

Display: potato text

Force long: long potato (potato)

1st ref. when long is empty defaults to short: potato@A

Force long when empty defaults to short: (potato@A)

2nd ref.: potato@A

1st ref.: long potato@B (potato@B)

### glspl

First: long potatoes@C (potatoes@C)

Force long: long potatoes (potatoes)

Force long when empty defaults to plural: (potatoes@A)

2nd ref.: potatoes@A

1st ref. when long is empty defaults to plural: potatoes@B

1st ref. when longplural is empty: long potato@Ds (potatoes@D)

2nd ref. when longplural is empty: potatoes@D

1st ref. when plural is empty: long potatoes@E (potato@Es)

### gls-description

Lorem ipsum dolor sit amet.

### gls-key

potato C

### gls-short

potato@D

### gls-plural

potatoes@A

### gls-longplural

long potatoes@B

### gls-longplural.with(link: true)

long potatoes@B

### gls-description

Lorem ipsum dolor sit amet.

### gls-group

D

### count-refs

2

### count-all-refs

All refs: (  
 ("potato", 6),  
 ("potato A", 5),  
 ("potato B", 2),  
 ("hidden potato", 0),  
 ("potato C", 1),  
 ("potato D", 2),  
 ("potato E", 1),  
 ("potato F", 0),  
 ("potato G", 0),  
 ("potato H", 6),

```
    ("potato I", 6),  
)
```

Only list 4: (("potato H", 6), ("potato I", 6))

#### **there-are-refs**

true: **true** ) false: **false**

There are refs to potato A in the document.

**test \_\_attribute\_is\_empty error**

**test \_\_key\_not\_found error**

**#print-glossary.with(show-all: true, disable-back-references: true)**

*potato* – long potato: Lorem ipsum dolor sit amet.

A

*potato@A*: Lorem ipsum dolor sit amet.

B

*potato@B* – long potato@B: Lorem ipsum dolor sit amet.

**show-all test**

*hiddenpotato*: This potato does not appear, unless print-glossary is called with show-all: true

**print-glossary (show-all: false)**

C

*potato@C* – long potato@C: Lorem ipsum dolor sit amet. 3

D

*potato@D* – long potato@D: Lorem ipsum dolor sit amet. 3, 3

E

*potato@E* – long potato@E: Lorem ipsum dolor sit amet. 3

G

*potato@G* – long potato@G: This potato should appear in the document.

### Test #print-glossary(none) error

#### Test only short/long

*potato@H*: This potato only has a short key.

*potato I* – long potato@I: This potato only has a long key.

long potato@I (potato I)

potato Is

a potato I

Potato I

Potato Is

A potato I

potato@H

potato@Hs

a potato@H

Potato@H

Potato@Hs

A potato@H

### Test neither short nor long error

#### #default-print-reference.with(show-all: true)

Forcibly print nonreference entry.

*potato@F* – long potato@F: This potato should not appear in the document.

#### Test use-key-as-short

*potato-use-key-as-short* 6

potato-use-key-as-short

#### capitalization

*LASER* 6, 6

*laser* 6, 6

LASER

LASERs

laser

Laser

#### always-long

1st: long (short)

2nd: long (short)

*short* – long 6, 6

#### no reference link

no-link

*no-link* 6