

Manuale d'uso

Dall'installazione locale Amusewiki al Bookbuilder, passando per il
collettore Mycorrhiza

Indice

Amusewiki e il Muse Markup	5
Paragrafi	7
Paragrafi e citazioni centrati o allineati a destra	7
Fonte	7
Paragrafi letterali	8
Interruzioni di linea	8
Interruzioni di pagina	9
Livelli della titolazione	11
Quarto livello, una “sottosezione”	11
Gli esempi dei livelli	11
Titolo alternativo	11
Direttive all’inizio del documento	13
Correcting the hyphenation in the PDF output	15
Bold, italicized and monospace text, non breaking space.	16
Non-breaking space.	17
Footnotes	18
Secondary footnotes (support added in Text::Amuse 0.91, 2017-12-10)	18
Poetic stanzas	20
Lists	22
Breaking lists	24
Nested lists	24
Breaking list items	25
Complete example	26
List continuation	29
Generation of data tables	31
GitHub Markdown tables	31
Floating tables	32
Hyperlinks with or without description (and images)	33
Images	33
Floating images and adjusting width	33
Examples	34
Inserting a horizontal line	36
Named anchors	37

Lines to omit from published output	39
Plays and bibliographies	40
Preventing the markup to be interpreted (verbatim)	42
Language support (version 1.80 2022-02-12)	43
Differences between <code>Text::Amuse</code> and Emacs Muse	44
Inline markup	44
Underlining	44
Tags for emphasis	44
Superscript and subscript tags	44
Allowed characters before lightweight markup	44
Block markup	44
Languages switches (<code><[ISO]></code>) and the ruby tag:	45
 Mycorrhiza	 46
Fare una ricerca avanzata tramite query nella barra di ricerca	48
Operators	48
AND	48
OR	48
NOT	48
XOR	48
Bracketed expressions	48
NEAR	48
ADJ	49
Phrase searches	49
Searching for proper names	49
Range searches	49
Synonyms	49
Wildcards	49
Partially entered query matching	49
 Metadata editing	 51
 Bookbuilder	 52
Introduction	53
 Getting the texts into the bookbuilder	 54
 The Bookbuilder interface	 55
PDF options	55
Fonts	55
Page dimensions	55
The division factor and the binding correction	55
Table of contents options	56
Two side layout	56

Running headings	56
The cover image	56
Imposition	57
Crop marks	58
EPUB	58
Collections	59
Partial selections	59
 Gestire un archivio con l'accesso al collettore	 60
Best practices sull'uso degli utenti	61
Best practices sui software di catalogazione	62
Possibilità di intestazione	62
Risultati di ricerca	62
Scheda del testo	62
Dati ancora non utilizzati o ad uso interno dell'archivio	63
Alcuni esempi di software e/o fonti di dati	64
abebooks (genera csv)	64
Calibre (genera csv o scansione albero)	64
koha (oai-pmh)	65
csv	65
amusewiki (oai-pmh)	65
vufind (oai-pmh)	65
 Workflows	 66
Eliminare un testo su Amusewiki	67
Aggiungere un libro	68
Lavorare su Amusewiki	68
Aggiungere una rivista o una collezione di riviste	69
Titolazione	69
Lavorare su Amusewiki	69

Amusewiki e il Muse Markup

Un documento Muse utilizza regole di marcatura speciali e contestuali per determinare come debba essere formattato il risultato finale. Per esempio, se un paragrafo è indentato, Muse assume che dovrebbe essere una citazione. L'indentazione è molto usata per determinare se il paragrafo è qualcosa di diverso da un testo "normale".

Non ci sono troppe regole di marcature, e tutte cercano di essere il più semplici possibili in modo che ti possa focalizzare sulla creazione del documento piuttosto che sulla sua formattazione.

Questa documentazione descrive Muse, che è stato scritto da John Wiegley, quindi mantenuto da Michael Olson e pesantemente modificato e parzialmente riscritto per questo utilizzo da Marco Pessotto.

Paragrafi

I paragrafi in Muse devono essere separati da una linea vuota.

Paragrafi e citazioni centrati o allineati a destra

Una linea che comincia con sei o più colonne di spazio bianco (ma meno di 20) indica un paragrafo centrato.

In alternativa, puoi usare il tag `<center>` che circonda parti che devono essere pubblicate come paragrafi centrati.

Come qui

Ma se la linea comincia con più di 20 spazi, stai creando un paragrafo allineato a destra. Questo è utile per le firme.

Come questo allineato a destra

Lo stesso risultato può essere ottenuto con il tag `<right>` .

Usando i tag:

Come questo centrato

E questo a destra

E questa è la citazione:

Se infatti una linea comincia con dello spazio bianco (almeno due spazi), ma comunque meno che sei colonne, viene indicato un paragrafo citato. In alternativa puoi usare il tag `<quote>` per circondare la parte che vuoi pubblicare come paragrafo citato.

Fonte

Una linea che comincia con sei o più colonne di spazio bianco (ma meno di 20) indica un paragrafo centrato.

In alternativa, puoi usare il tag `=<center>=` che circonda parti che devono essere pubblicate come paragrafi centrati.

Come qui

Ma se la linea comincia con più di 20 spazi, stai creando un paragrafo allineato a destra. Questo è utile per le firme.

Come questo allineato a destra

Lo stesso risultato può essere ottenuto con il tag `=<right>=` .

Usando i tag:

```
<center>
Come questo centrato
</center>
```

```
<right>
E questo a destra
</right>
```

E questa è la citazione:

Se infatti una linea comincia con dello spazio bianco (almeno due spazi), ma comunque meno che sei

Per favore ricorda che l'indentazione deve essere coerente se preferisci mantenere le linee brevi e la rompi se inserisci una nuova linea. Linee lunghe (utilizzando la regola 1 linea 1 paragrafo) sono anche perfettamente adeguate. La regole dei paragrafi separati da una linea bianca si applica ugualmente, comunque.

Quindi, `<tags>` che aprono e chiudono blocchi, **devono essere messi su una linea a sé stante e non devono essere mischiati con il circostante segnato dagli spazi iniziali, specialmente nel casi di elenchi e tabelle.**

Paragrafi letterali

Il tag `<example>` è usato per gli esempi, dove lo spazio bianco deve essere preservato, il testo reso in monospace ed ogni carattere speciale evitato nello stile finale.

Esempio:

```
<example>
Il tag =<example>= è usato per gli esempi, dove lo spazio bianco deve essere preservato, il testo re
</example>
```

Non esiste un tag `<literal>` come nel Muse markup originale perché non è uno strumento privato e sarà esposto ad internet.

Per favore nota che la resa manterrà le linee non modificate. Questo vuol dire che è molto probabile che otterrai delle linee troppo lunghe. Per evitare questo, un valore sicuro è quello di usare 60 caratteri per linea. Valori maggiori usali a tuo rischio.

Una sintassi alternativa per il tag `<example>` è `{{{ }}}`:

```
{{{
Anche questo è verbatim
}}}
```

Interruzioni di linea

Se hai bisogno di un interruzione di linea, usa il tag `<br>`. La maggior parte delle volte non ne avrai bisogno, perché Muse identifica automaticamente i paragrafi attraverso le linee vuote. Ma se vuoi

paragraphs by means of blank lines. Se vuoi preservare in un solo paragrafo più righe di testo, allora usa il markup `<verse>`.

Questa linea si interromperà `<br>` E continuerà

Questa linea si interromperà
E continuerà

Se vuoi aggiungere una linea bianca, aggiungi il tag `<br>` su una linea a sé stante:

Qui aggiungiamo una linea bianca

`<br>`

E qui proseguiamo

Qui aggiungiamo una linea bianca

E qui proseguiamo

Interruzioni di pagina

Se metti esattamente 5 “*” separati da uno spazio e indentati da 6 o più spazi (come un paragrafo dentrato), avrai un’interruzione di pagina nel PDF.

Questo codice interromperà la pagina

* * * * *

Uguualmente, usandone solo tre avrai un semplice decoratore, e non sarà trattato in modo particolare.

* * *

Si tratterà semplicemente di un paragrafo centrato con tre “*”.

Livelli della titolazione

Un titolo diventa un capitolo o una sezione nel risultato di stampa in funzione dello stile. Per indicare un titolo, comincia un nuovo paragrafo con uno o più asterischi, seguiti da uno spazio e il titolo. Quindi comincia un altro paragrafo per inserire il testo di quella sezione.

Tutti i livelli di titolo saranno pubblicati, e sono supportati 5 livelli.

Il primo livello è una "parte", e dovrebbe essere usato solo per i testi più grandi. In questo documento è usato per il titolo principale.

Il secondo livello è il "capitolo". Comincia con una pagina nuova nel risultato in PDF.

Il terzo livello è senza dubbio quello più usato. Attualmente separa le sezioni di un articolo. per un esempio guarda "paragrafi letterali" poco più su.

Il quarto livello scende ulteriormente.

Quarto livello, una "sottosezione"

Il quinto livello è molto, molto basso e non rientra neppure nell'indice.

Quinto livello 1

Del testo.

Quinto livello 2

Altro testo.

Gli esempi dei livelli

* Primo livello, aka parte

** Secondo livello, aka chapter capitolo

*** Terzo livello, aka sezione

**** Quarto livello, aka sottosezione

***** Quinto livello, aka sottosottosezione

Come indicare un titolo alternativo

[Questa sintassi è disponibile da *Text::Amuse* version 1.40, rilasciata il 2020-02-16.]

A volte è preferibile avere un titolo alternativo, solitamente più corto, per l'indice. Puoi indicarlo utilizzando questa sintassi:

** Titolo alternativo | Come indicare un titolo alternativo

Ovvero separando il titolo corto da quello lungo con una barretta | circondata da spazi.
In questo esempio, "Titolo alternativo" andrà nell'indice, mentre la seconda parte nel documento.

Direttive all'inizio del documento

Directives are lines beginning with the ‘#’ character that come before any paragraphs or sections in the document.

Directives are of the form `#directive content of directive`.

You can use any combination of uppercase and lowercase letters for directives, even if the directive is not in the list below. The directives are completely arbitrary. You can put there whatever you want. It's the template job to pick them up. In the templates shipped with this bundle there is support for the following directives:

The following is a list of directives that Muse uses.

- **#author**

The author of the text.

- **#title**

The title of the document

- **#lang**

The language code of the document (2 or 3 letters). Defaults to **en**.

- **#LISTtitle**

This directive is used (defaulting to **#title**) to alphabetically sort the titles. It's handy if you want, for example, sort “A title” under “T” and not under “A”.

In this case you may write `#LISTtitle Title`

- **#subtitle**

The subtitle (if any)

- **#SORTauthors**

If not provided, this default to **#author**. It's a list separated by semicolons or commas with the various authors. While **#author** affects the display only, this one is used to index the document.

- **#SORTtopics**

As for authors, it's a list (comma- or semicolon-separated) list of topics for the current text. Used to index the document.

- **#date**

The **year** of publishing of the document. More information should be provided in the **#notes** directive.

- **#notes**

This directive is used for additional information here (original title, translators, credits, etc.).

- **#source**

This directive is used for the source or the text (url, scanned from original, original contribution, etc.). The preferred format is “Retrieved on March 8, 2012 from the url”

- **#publisher**

Publisher data, if any.

- **#isbn**

ISBN, if any.

- **#rights**

Copyright info, if any.

- **#seriesname**

If the book belongs to a serie, put it here.

- **#seriesnumber**

If the book belongs to a serie, use this slot for the number.

- **#hyphenation**

See below.

A number of directives affects the output formats. The directives have a mandatory argument, so ”1” usually means a generic true value.

- **#slides 1**

Indicates that the document is meant to produce slides as well as the regular formats.

- **#DELETED REASON**

Indicates that the text is not meant to be published.

- **#cover file.jpg**

Put the named image (jpg or png, in the same directory of the muse file) on the cover page.

- **#coverwidth 0.5**

Specify the width of the cover image in a fraction of the text block width. This is mostly to give you some control over its dimensions if the default doesn’t look good.

- **#nocoverpage 1**

Do not create a cover page but start the text right away.

- **#notoc 1**

Do not create a table of contents.

- **#nofinalpage 1**

Do not create a final page.

- **#impressum 1**

Create an impressum page after the cover page.

- `#continuefootnotes 1`
Continue the footnotes numbering across chapters.
- `#centerchapter 1`
Use centered chapter titles.
- `#centersection 1`
Use centered section titles (all levels).
- `#formats CODE`
Select the custom formats for this text.

Correcting the hyphenation in the PDF output

Sometimes you may notice some words with incorrect hyphenation in your document. You can fix this adding the breakpoint in the `#hyphenation` directive.

E.g.

```
#title Test
#lang it
#hyphenation al-be-rel-lo que-sto
```

Questo alberello...

You can as many words with breakpoints as you wish, separated by spaces, but you can't insert numbers or special characters (accents and diacritics are fine, though). You specify a breakpoint with the hyphen character “-”. Using a word without any hyphen will prevent the hyphenation for that word.

Bold, italicized and monospace text, non breaking space.

To emphasize text, surround it with certain specially recognized characters.
The following example will produce:

emphasis **strong emphasis** *very strong emphasis* verbatim and monospace

```
*emphasis*
**strong emphasis**
***very strong emphasis***
=verbatim and monospace=
```

Each of these forms may span multiple lines, but not multiple paragraphs.

You can also use the `<code>` tag to indicate code and monospace text. This is handy for regions that have a “=” in them.

If the “*” confuse you or the preview is screwed up, you can use inline tag `<em>` and `<strong>`, which are guaranteed to work in any case.

The above example rewritten with tags:

```
<em>emphasis</em>
<strong>strong emphasis</strong>
<strong><em>very strong emphasis</em></strong>
<code>verbatim and monospace</code>
```

And produces the same thing.

emphasis **strong emphasis** *very strong emphasis* verbatim and monospace

Please note that there is no support for the underline. Underlining is an handwritten substitute for the italics. You simply don’t need it.

Also, small caps are missing, mainly because on the HTML they look awful and a very few fonts have decent small caps.

Since Text::Amuse version 0.96 (released 2018-01-27), which restored Emacs Muse compatibility in this regard, material in `<code>` tags and equivalent markup between equal signs is also verbatim, but with a monospace font.

Other tags are `<sub>` and `<sup>` for subscript and superscript. And there is also a `<del>` tag for overstriking.

This is the `<sup>`superscript`</sup>` and this is a `<sub>`subscript`</sub>`, and this is `<del>`something deleted and overstriked`</del>`

This is the ^{superscript} and this is a _{subscript}, and this is ~~something deleted and overstriked~~

If you nest the same tag (e.g. `<em>`this `<em>`and this`</em></em>`, you are going to get weird results (and doesn’t make any sense), so don’t do it. = and * when surrounding words have the same meaning of the respective tags:

`*this*` is the same as `<em>this</em>`

`=this=` is the same as `<code>this</code>`

Starting with Text::Amuse version 1.72 (released on February 6, 2021), there are two additional inline tags.

`<sf>Sans Serif font</sf>` and `<sc>Small Caps</sc>`

Resulting in Sans Serif font and SMALL CAPS.

Caution is advised when using the `sans` tag. There is no semantic attached to it and basically provides access to an alternate font. There is little use for that in a web application (e.g., Amusewiki pages), where if the main font is already a sans serif (like in the default Amusewiki theme), it would be invisible. However, you can customize the output adding a CSS rule for the class `muse-sf`.

However, there is a potential use for this font, as you can switch to a secondary font (labeled as sans-serif, but not necessarily a sans one) if the main one is missing some characters (e.g., you suddenly need greek fonts, and your main one doesn't have them). In this case you could read `sf` as "switch font".

Non-breaking space.

Non breaking space (0xA0 Unicode, NO-BREAK SPACE) is just a regular character, but it's somehow complicate to manage, because very often appears like a normal space. You can use a double tilde `~~` to insert a non breaking space which is explicit in the muse document.

This feature was added in Text::Amuse 0.94 but it was present in Emacs Muse.

Footnotes

A footnote reference is simply a number in square brackets. To define the footnote, place this definition starting the line with a digit in square brackets.

This is the text, and we refer to a footnote [1]

Here the text continues.

[1] This footnote
spans more lines in the source

You can continue the footnote on another paragraph, as long as it
has the same amount of indentation of the previous item.

[2] But this is not, because of the initial
whitespace.

[3] Footnotes which don't have a referrer will
disappear on the PDF output and preserved in the
HTML. But will lead to incorrect code, as it will
point to a non-existent anchor

This is the result:

This is the text, and we refer to a footnote¹
Here the text continues.
[2] But this is not, because of the initial whitespace.

You can break the footnotes lines (even if it's not recommended), but keep the indentation consistent, as shown above.

Secondary footnotes (support added in Text::Amuse 0.91, 2017-12-10)

Rarely needed, but supported, are the secondary footnotes, i.e. an additional apparatus. They obey the same rules as the regular footnotes, but they are marked with curly brackets instead of square ones.

¹ This footnote spans more lines in the source

You can also place secondary footnotes in regular footnotes. This is meant for critical edition, but you may use them to differentiate between author's notes and translator's notes.

This is a regular [4] footnote, and this a secondary {1}

[4] Regular footnote, and has a secondary one on it {2}

{1} Secondary footnote body (1)

{2} Secondary footnote body (2)

Which produces:

This is a regular² footnote, and this a secondary⁽¹⁾

You can continue the footnote on another paragraph, as long as it has the same amount of indentation of the previous item.

² Regular footnote, and has a secondary one on it⁽²⁾

⁽¹⁾ Secondary footnote body (1)

⁽²⁾ Secondary footnote body (2)

Poetic stanzas

Poetry requires that whitespace be preserved, but without resorting to monospace. To indicate this, use the following markup, reminiscent of email quotations, or use the `verse` tag.

```
> A line of Emacs verse;  
>   forgive its being so terse.
```

```
<verse>  
  A line of Emacs verse;  
  forgive its being so terse.  
</verse>
```

This yields:

```
A line of Emacs verse;  
  forgive its being so terse.
```

```
A line of Emacs verse;  
  forgive its being so terse.
```

Multiple stanzas may be included in one set of `<verse>` tags, as follows.

```
<verse>  
A line of Emacs verse;  
  forgive its being so terse.
```

```
In terms of terse verse,  
  you could do worse.  
</verse>
```

Or this

```
> A line of Emacs verse;  
>   forgive its being so terse.  
>  
> In terms of terse verse,  
>   you could do worse.
```

```
A line of Emacs verse;  
  forgive its being so terse.
```

```
In terms of terse verse,  
  you could do worse.
```

Or this

A line of Emacs verse;
 forgive its being so terse.
In terms of terse verse,
 you could do worse.

Lists

Lists are given using special characters at the beginning of a line. Whitespace must occur before bullets or numbered items, to distinguish from the possibility of those characters occurring in a real sentence.

Description lists are marked by some initial whitespace, the term, a double colon surrounded by whitespace, and the description body.

Normal text.

- bullet item one
- bullet item two

An enumerated list follows.

1. Enum item one
2. Enum item two

A list with roman numbering

- i. First
- ii. Second
- iii. Third

A list with upper roman numbering

- I. First
- II. Second
- III. Third

A list with upper letters

- A. first
- B. second
- C. third

A list with lower letters

- a. first
- b. second
- c. third

A description list

First term :: definition and description

Second term :: definition and description

Please note the consistent indentation, **especially** for roman numbering.
Normal text.

- bullet item one
- bullet item two

An enumerated list follows.

1. Enum item one
2. Enum item two

A list with roman numbering

- i. First
- ii. Second
- iii. Third

A list with upper roman numbering

- I. First
- II. Second
- III. Third

A list with upper letters

- A. first
- B. second
- C. third

A list with lower letters

- a. first
- b. second
- c. third

A description list

First term definition and description

Second term definition and description

Breaking lists

If for some reason you want to break the list without starting a regular paragraph, you can do so by inserting a `<br>` tag (which adds some white space between them) or a comment (invisible). E.g.

List:

- a. bullet item one
- a. bullet item two, and will break

; a comment

- a. bullet item one
- a. bullet item two, and break

`<br>`

- a. bullet item one
- a. bullet item two, and end

Resulting in:

List:

- a. bullet item one
- b. bullet item two, and will break
- a. bullet item one
- b. bullet item two, and break

- a. bullet item one
- b. bullet item two, and end

Nested lists

It is possible to nest lists of the same or different kinds. The “level” of the list is determined by the amount of initial whitespace.

Normal text.

- Level 1, bullet item one
 - 1. Level 2, enum item one
 - 2. Level 2, enum item two
- Level 1, bullet item two
 - 1. Level 2, enum item one
 - 2. Level 2, enum item two
 - i. Level 3, enum item i
 - ii. Level 3, enum item ii

- 3. Level 2, enum item three
- Back to Level 1, third bullet
 - a. Level 2, enum item “a”
 - b. Level 2, enum item “b”
 - I. Level 3, enum item “I”
 - One term :: description
 - Another term :: description
- Back to the bullets

Normal text.

- Level 1, bullet item one
 - 1. Level 2, enum item one
 - 2. Level 2, enum item two
- Level 1, bullet item two
 - 1. Level 2, enum item one
 - 2. Level 2, enum item two
 - i. Level 3, enum item i
 - ii. Level 3, enum item ii
 - 3. Level 2, enum item three
- Back to Level 1, third bullet
 - a. Level 2, enum item “a”
 - b. Level 2, enum item “b”
 - I. Level 3, enum item “I”
 - One term** description
 - Another term** description
- Back to the bullets

Breaking list items

If you want to break up a line within any list type, just put one blank line between the end of the previous line and the beginning of the next line, using the same amount of initial indentation.

Keep in mind that if you put random indentation you’ll get random and probably unexpected results (but it should not crash — if it does, please contact me).

Also, you can be lazy with numbered list. The parser actually doesn’t care if you number them properly, or just do something like that.

- 1. first
- 1. second
- 1. third

or

- a. first
- a. second
- a. third

There results will always be:

- 1. first
- 2. second
- 3. third

or

- a. first
- b. second
- c. third

Complete example

Normal text.

- Level 1, bullet item one, this is the first paragraph. I can break the line, keeping the same amount of indentation

Here I have the same amount of indentation, and it continues the item above.

- 1. Level 2, enum item one. I can break the line, keeping the same amount of indentation

Here I have the same amount of indentation, and it continues the item above.

- 2. Level 2, enum item two which continues

Here I have the same amount of indentation, and it continues the item above.

- Level 1, bullet item two which continues

Here I have the same amount of indentation, and it continues the item above.

- 1. Level 2, enum item one which continues

Here I have the same amount of indentation, and
it continues the item above.

- 2. Level 2, enum item two
which continues

Here I have the same amount of indentation, and
it continues the item above.

- i. Level 3, enum item i

Here I have the same amount of indentation,
and it continues the item above.

- ii. Level 3, enum item ii

Here I have the same amount of indentation,
and it continues the item above.

- 3. Level 2, enum item three
which continues

Here I have the same amount of indentation, and
it continues the item above.

- Back to Level 1, third bullet

Here I have the same amount of indentation, and it
continues the item above.

- a. Level 2, enum item ‘a’
which continues

Here I have the same amount of indentation, and
it continues the item above.

- b. Level 2, enum item ‘b’
which continues

Here I have the same amount of indentation, and
it continues the item above.

- I. Level 3, enum item ‘I’

Here I have the same amount of indentation,
and it continues the item above.

And inside this item :: a description

Which continues here.

Another term :: a description

Which continues here.

- Back to the bullets

Here I have the same amount of indentation, and it continues the item above.

Normal text.

- Level 1, bullet item one, this is the first paragraph. I can break the line, keeping the same amount of indentation

Here I have the same amount of indentation, and it continues the item above.

1. Level 2, enum item one. i can break the line, keeping the same amount of indentation

Here I have the same amount of indentation, and it continues the item above.

2. Level 2, enum item two which continues

Here I have the same amount of indentation, and it continues the item above.

- Level 1, bullet item two which continues

Here I have the same amount of indentation, and it continues the item above.

1. Level 2, enum item one which continues

Here I have the same amount of indentation, and it continues the item above.

2. Level 2, enum item two which continues

Here I have the same amount of indentation, and it continues the item above.

- i. Level 3, enum item i

Here I have the same amount of indentation, and it continues the item above.

- ii. Level 3, enum item ii

Here I have the same amount of indentation, and it continues the item above.

3. Level 2, enum item three which continues

Here I have the same amount of indentation, and it continues the item above.

- Back to Level 1, third bullet

Here I have the same amount of indentation, and it continues the item above.

- a. Level 2, enum item “a” which continues

Here I have the same amount of indentation, and it continues the item above.

- b. Level 2, enum item “b” which continues

Here I have the same amount of indentation, and it continues the item above.

I. Level 3, enum item “T”

Here I have the same amount of indentation, and it continues the item above.

And inside this item a description

Which continues here.

Another term a description

Which continues here.

- Back to the bullets

Here I have the same amount of indentation, and it continues the item above.

List continuation

If you need to start a list from an index different than 1, starting with Text::Amuse 0.90 (released on August 30, 2017), you can do so simply using it in the list. Indexes provided are so respected, unless they are the number 1. So the rule is: **if you need automatic numbering, just use 1. or a. or A. or i. or I., otherwise use your custom numbering.** Any other solution will be confusing and you can expect undefined behaviour.

- b. This list with start with b
- d. This will be “d” as well
 - 3. This starts at 3
 - 2. This is 2 as well.
 - iv. Roman iv.
 - i. Roman v. as i. acts as automatic numbering
- a. This will be e. because it’s index 1 and continues the list (interrupted at d.)
- a. This will be f. because it’s incrementing the previous item.
 - 1. Normal numeric
 - 1. Normal numeric 2.
 - 1. Normal numeric 3.
 - V. Roman V.
 - X. Roman X.

And results in:

- b. This list with start with b
- d. This will be “d” as well
 - 3. This starts at 3
 - 2. This is 2 as well.
 - iv. Roman iv.
 - v. Roman v. as i. acts as automatic numbering
- e. This will be e. because it’s index 1 and continues the list (interrupted at d.)
- f. This will be f. because it’s incrementing the previous item.
 - 1. Normal numeric
 - 2. Normal numeric 2.

3. Normal numeric 3.

V. Roman V.

X. Roman X.

Generation of data tables

Only simple tables are supported. The syntax is as follows (just keep the indentation consistent and separate each cell by one or more textbars).

Indentation (one or more leading spaces) is required to trigger the table rendering.

```
Triple bars ||| Separate footer fields
Double bars  || Separate header fields
Single bars   | Separate body fields
Here are more | body fields
|+ This is the caption +|
```

Double bars	Separate header fields
Single bars	Separate body fields
Here are more	body fields
Triple bars	Separate footer fields

This is the caption

The ordering of the footer, header and table body blocks is irrelevant for the output (HTML requires you put first the header, then the footer, then the body). Ordering of the single rows is of course preserved. Inside the cells you can do pretty much what you want (besides headers and lists): mark them up freely.

GitHub Markdown tables

They mainly work like the native tables, with some differences:

- the line starts with the pipe symbol and doesn't need to have leading whitespace.
- You can mark the header with hyphens (at least 3), drawing a dashed line, as shown above (it needs to be the second line for this purpose).
- You can control the alignment of the column using the line marking the header, placing a colon on the left, on the right, or on both ends (centering) of the dashed line. Please note that the header itself is not required. See the examples below.

The line marking the header determines the alignment of the column (see where the colons are):

```
| Left aligned | Centered | Right aligned | Justified text |
| :-----: | :-----: | -----: | ----- |
| Content Cell | Cell 1 | *Right* | This text is justified and can be very long |
| See **here** | Cell 2 | Right | This *text* is justified and can be very long |
|+ legend +|
```

Output:

And without header (but still controlling the alignment):

Left aligned	Centered	Right aligned	Justified text
Content Cell	Cell 1	<i>Right</i>	This text is justified and can be very long
See here	Cell 2	Right	This <i>text</i> is justified and can be very long

legend

```
| :----- | :-----: | -----: | ----- |
| Content Cell | Cell 1 | *Right text* | This text is justified and can be very long |
| See **here** | Cell 2 | Right | This *text* is justified and can be very long |
|+ legend +|
```

Output:

Content Cell	Cell 1	<i>Right text</i>	This text is justified and can be very long
See here	Cell 2	Right	This <i>text</i> is justified and can be very long

legend

This feature was introduced on Text::Amuse 1.51, released 2020-04-02.

Floating tables

When composing tables, please keep in mind the following:

- tables are never splat over pages. If they don't fit a page, they will run off the page. So keep them short or split them yourself.
- tables *without* a caption are inserted at the point they are in the source. Table *with* a caption are converted into a float, so they are in no way guaranteed they will appear in the same exact point of the source. Anyway, the result is way better, and you have a caption, so you can refer to them in the text.

Hyperlinks with or without description (and images)

A hyperlink can reference a URL or a place on the same document. In addition, descriptive text can be specified, which should be displayed rather than the link text in output styles that supports link descriptions. The syntax is as follows.

```
[[link target][link description]]
[[link target without description]]
```

So, the home of this project is
[[http://amusewiki.org] [AMuseWiki]],
which can be found at
[[http://amusewiki.org]] Bare links will not
get the hyperlinking. So http://thisisspam.org won't
get the hyperlinking.

So, the home of this project is AMuseWiki, which can be found at <http://amusewiki.org> Bare links will not get the hyperlinking. So <http://thisisspam.org> won't get the hyperlinking.

Images

Images are special case of this kind of linking.

[XXX]

We assume that we have the “m-l-manual-logo.png” file on the same directory of the file, and the result is:

[XXX]

Now, let's add a caption.

[[XXX][This is *our* logo]]

This is *our* logo

Remote urls are not permitted. Also, the path checking is rather strict, so please use just alphanumeric filenames for your images.

Floating images and adjusting width

By default, image width in the HTML output is kept to the original one, whatever it is, but limited via CSS, while in the PDF expands to fill the page width. This creates problems.

Starting with version 0.07, it's possible to set the width of the image, **in percent**, just appending it to the file name after a whitespace.

It's also possible to make it a right or left float, adding the character `l` or `r`, or to mark it as a “fullpage” with `f`.

This one will have a width to 80% of the page width:

`[[XXX]]`

This one will have a width to 80% of the page width and will be a left float.

`[[XXX]]`

This one will have a width to 80% of the page width and will be a right float.

`[[XXX]]`

The Following will be marked as a fullpage.

`[[XXX]]`

Floating without a width doesn't make sense, as in the PDF will be no room for wrapping text around it.

Examples

The next figures use these codes:

`[XXX]`

`[XXX]`

`[XXX]`

Suggestions that I write my memoirs came to me when I had barely begun to live, and continued all through the years. But I never paid heed to the proposal. I was living my life intensely — what need to write about it? Another reason for my reluctance was the conviction I entertained that one should write about one's life only when one had ceased to stand in the very torrent of it. “When one has reached a good philosophic age,” I used to tell my friends, “capable of viewing the tragedies and comedies of life impersonally and detachedly — particularly one's own life — one is likely to create an autobiography worth while.” Still feeling adolescently young in spite of advancing years, I did not consider myself competent to undertake such a task. Moreover, I always lacked the necessary leisure for concentrated writing.

right float

My enforced European inactivity left me enough time to read a great deal, including biographies and autobiographies. I discovered, much to my discomfiture, that old age, far from ripening wisdom and mellowness, is too often fraught with senility, narrowness, and petty rancour. I would not risk such a calamity, and I began to think seriously about writing my life.

left float

The great difficulty that faced me was lack of historical data for my work. Almost everything in the way of books, correspondence, and similar material that I had accumulated during the thirty-five years of my life in the United States had been confiscated by the Department of Justice raiders and never returned. I lacked even my personal set of the *Mother Earth* magazine, which I had published for twelve years.

It was a problem I could see no solution for. Sceptic that I am, I had overlooked the magic power of friendship, which had so often in my life made mountains move. My staunch friends Leonard D. Abbott, Agnes Inglis, W. S. Van Valkenburgh, and others soon put my doubts to shame.

Agnes, the founder of the Labadie Library in Detroit, containing the richest collection of radical and revolutionary material in America, came to my aid with her usual readiness. Leonard did his share, and Van spent all his free time in research work for me.

The great difficulty that faced me was lack of historical data for my work. Almost everything in the way of books, correspondence, and similar material that I had accumulated during the thirty-five years of my life in the United States had been confiscated by the Department of Justice raiders and never returned. I lacked even my personal set of the Mother Earth magazine, which I had published for twelve years.

It was a problem I could see no solution for. Sceptic that I am, I had overlooked the magic power of friendship, which had so often in my life made mountains move. My staunch friends Leonard D. Abbott, Agnes Inglis, W. S. Van Valkenburgh, and others soon put my doubts to shame.

Agnes, the founder of the Labadie Library in Detroit, containing the richest collection of radical and revolutionary material in America, came to my aid with her usual readiness. Leonard did his share, and Van spent all his free time in research work for me.

full page

Inserting a horizontal line

Four or more dashes indicate a horizontal rule. Be sure to put blank lines around it, or it will be considered part of the proceeding or following paragraph, like this: -----.

Example

Results:

Named anchors

If you start a line (regardless of the indentation) with a word prefixed by an hash, e.g. `#anchor` – where `anchor` can be any (ASCII) word that starts with a letter and contains only ASCII letters, digits and dashes – it defines an anchor at that point into the document. This point can be referenced using “`#anchor`” as the target in a Muse link.

If you need to start a line with an hash, wrap it in `<verbatim>` E.g.

```
<verbatim>#hashtag</verbatim> verbatim.  
=#hashtag= verbatim as code.
```

Yielding:

```
#hashtag verbatim. #hashtag verbatim as code.
```

The support for this feature was introduced in Text::Amuse version 0.70 and improved in 0.82. Compatibility with Emacs Muse was restored in version 1.10, released on April 23, 2018, allowing text material on the same line of the anchor.

Anchors adjacent to section headers will be attached to them, while lonely anchors not adjacent to any material will attach to the next first regular text.

Example:

```
#title My text
```

```
#badanchor
```

```
This is not an anchor, it's part of the header with a dummy name.
```

```
<br>
```

```
#begin
```

```
This will work.
```

```
#an-anchor At the beginning of the line.
```

```
<verbatim>#hashtag</verbatim> verbatim.
```

```
=#hashtag= verbatim as code.
```

```
#anchor-to-section
```

```
*** Section
```

```
#anchor
```

```
Here we have our text.
```

```
More text.
```

```
*** Next section with an anchor
```

`#nextsection`

And here we can link to `[[#anchor][our anchor]]` and to the `[[#begin][begin]]` and to `[[#nextsection][this section]]`.

Please note that the name in the HTML output is not kept verbatim. Instead a `text-amuse-label` prefix is added (if you need to link it externally).

Also note that the link is lost when producing imposed PDFs (and implicitly lost when printing any PDF), so it's recommended to keep this in mind when writing the document, as the text of the link should work even if the link is not present.

Bad example:

See `[[#anchor][here]]`

The poor soul printing it and reading it on paper will see just "See here". Here where?

Good example:

See `[[#anchor][the previous chapter]]` for more information.

If you give preference to the printing, you may want to have a page number instead of a hyperlink. You can then use this syntax:

See `[[#anchor][the previous chapter on page ??]]` for more information.

I.e., the double question mark ?? will be replaced by the page number in the PDF and kept verbatim in the HTML. This feature was introduced on Text::Amuse 1.50, released 2020-03-28.

Lines to omit from published output

```
; Comment text goes here.
```

That is, only a semi-colon at the beginning of a line, followed by a literal space, will cause that line to be treated as a comment and hidden from the visible output.

You can alternatively surround the region with the `<comment>` tag.

The HTML output will be wrapped in a `<div>` with the `display:none` property, so it can be turned visible changing the CSS.

```
<comment>
This won't be published, but in the HTML is there, only hidden
</comment>
```

Plays and bibliographies

Unlike the original Muse, this implementation doesn't support external sources for citations, but provides an environment which can be used to compose the list of cited works without resorting to lists, wrapping all in a `<biblio>` tag.

```
<biblio>
```

```
The author, *Title*, published on  
[[http://amusewiki.org][AMuseWiki]], with a very, very,  
very, very, very, very, very, very, very, very,  
very, long description
```

```
Another author, *Another title*, published in the real  
world. with a very, very, very, very, very, very, very,  
very, very, very, very, very, long description
```

```
</biblio>
```

The author, *Title*, published on AMuseWiki, with a very, very, very, very, very, very, very, very, very,
very, very, very, long description

Another author, *Another title*, published in the real world. with a very, very, very, very, very, very, very,
very, very, very, very, very, long description

The list is wrapped in a `<div>` with class set to `biblio`, while the TeX source gets wrapped in the `amusebiblio` environment. You are free to change class and environment definition if you need to. A reversed indentation is suggested.

The same goes with the `<play>` environment, which is supposed to wrap theatrical plays, when you want a reverse indentation and more spacing between the paragraph (without resorting to `<verse>` and hardcoding the spaces).

```
<play>
```

```
**Pol.** Ophelia, walke you heere. Gracious so please ye <br>  
We will bestow our selues: Reade on this booke, <br>  
That shew of such an exercise may colour <br>  
Your lonelinesse. We are oft too blame in this, <br>  
'Tis too much prou'd, that with Deuotions visage, <br>  
And pious Action, we do surge o're <br>  
The diuell himselfe
```

```
**King.** Oh 'tis true: <br>  
How smart a lash that speech doth giue my Conscience? <br>  
The Harlots Cheeke beautied with plaist'ring Art <br>  
Is not more vgly to the thing that helpes it, <br>  
Then is my deede, to my most painted word. <br>
```

Oh heauie burthen!

</play>

Pol. Ophelia, walke you heere. Gracious so please ye
We will bestow our selues: Reade on this booke,
That shew of such an exercise may colour
Your loneliness. We are oft too blame in this,
'Tis too much prou'd, that with Deuotions visage,
And pious Action, we do surge o're
The diuell himselfe

King. Oh 'tis true:
How smart a lash that speech doth giue my Conscience?
The Harlots Cheeke beautied with plaist'ring Art
Is not more vgly to the thing that helps it,
Then is my deede, to my most painted word.
Oh heauie burthen!

Preventing the markup to be interpreted (verbatim)

Sometimes you want to write something like [1] without meaning a footnote reference, or asterisks without meaning an emphasis, e.g. *this*.

You can do so wrapping the words with the `<verbatim>` tag.

Example:

Sometimes you want to write something like `<verbatim>[1]</verbatim>` without meaning a footnote reference, or asterisks without meaning an emphasis, e.g. `<verbatim>*this*</verbatim>`.

Since Text::Amuse version 0.96 (released 2018-01-27), which restored Emacs Muse compatibility in this regard, material in `<code>` tags and equivalent markup between equal signs is also verbatim, but with a monospace font.

Language support (version 1.80
2022-02-12)

You can mark parts of a text as using another language with the 1.80 `<[ISO]></[ISO]>` syntax, where ISO is the language code. The tag can be used as inline and block tag.

See the font management documentation to assign a specific font to a language.

E.g. (note that the monospace font in the PDF is without japanese characters):

```
#title Various languages
```

```
#lang en
```

Dante says <[it]>Lasciate ogni speranza, voi ch'intrate.</[it]>

You can inline some random Japanese with a

```
[[https://en.wikipedia.org/wiki/Ruby_character][ruby]]:
```

<[ja]>XXXXXXXX <ruby>X|X</ruby></[ja]>

And a whole random Japanese paragraph:

$\langle [ja] \rangle$

`<ruby>` | `</ruby>`

XXXXXXXXXXXX<ruby>X|XX</ruby>XXXXXXXXXXXXXXXXXXXX

`<ruby>``|`

</[ia]>

(Please note the ruby syntax in the japanese snippet).

This yields:

Dante says *Lasciate ogne speranza, voi ch'intrate*.

You can inline some random Japanese with a ruby: 松江騒擾事件^先

And a whole random Japanese paragraph:

先導者は先づ確固たる自信である。次に力である。次に勇氣である。而して自身の生命に対する自身の責任である。先導者は如何なる場合にも自分の仕事に他人の容喙を許さない。

Differences between Text::Amuse and Emacs Muse

Unfortunately, the Emacs Muse project is dead. However, pandoc supports the muse syntax.

Inline markup

Underlining

Underlining has been dropped.

Tags for emphasis

Emphasis and strong can also be written with tags, like `<em>emphasis</em>`, `<strong>strong</strong>` and `<code>code</code>`.

Superscript and subscript tags

Added tag `<sup>` and `<sub>` for superscript and subscript.

Allowed characters before lightweight markup

Asterisk and equal symbols (`*`, `**`, `***` `=`) are interpreted as markup elements if they are paired (an opening one and a closing one).

The opening one must be preceded by something which is not an alphanumerical character (or at the beginning of the line) and followed by something which is not a space.

The closing one must be preceded by something which is not a space, and followed by something which is not an alphanumerical character (or at the end of the line).

Unlike Emacs Muse (as of version 3.20.2), any non-alphanumeric characters are allowed before the opening `*` and `=`. For example, the following markup is interpreted as bold code by Amusewiki

```
***=Bold code=**
```

while Emacs Muse interprets it as bold `"=Bold code="`.

Block markup

The tables supported are the native one (with `|||` as separator) and the GitHub Markdown tables.

While Emacs Muse requires the bars to be surrounded by spaces to trigger a table, Text::Amuse requires some initial indentation. To write compatible table, please indent and keep the bars surrounded by spaces.

Since version 0.60, the code blocks, besides the `<example>` tag, can also be written as `{{{ }}}}, e.g.`

```
{{{
  if ($perl) {...}
}}}
```

Since 0.91, there is full support for secondary footnotes with number inside curly brackets. Emacs Muse is missing this feature.

Lists require whitespace after list item marker.

-This is not a list item.

1.This is not an ordered list item.

Languages switches (<[ISO]>) and the ruby tag:

They are `Text::Amuse` specific which were introduced on version 1.80 (2022-02-12).

Mycorrhiza

Questo testo è il manuale d'uso del collettore, in aggiornamento ed in continua modifica

Fare una ricerca avanzata tramite query nella barra di ricerca

This document describes the query syntax supported by the Xapian::QueryParser class. The syntax is designed to be similar to other web based search engines, so that users familiar with them don't have to learn a whole new syntax.

Operators

AND

expression AND expression matches documents which are matched by both of the subexpressions.

OR

expression OR expression matches documents which are matched by either of the subexpressions.

NOT

expression NOT expression matches documents which are matched by only the first subexpression. This can also be written as expression AND NOT expression. If FLAG_PURE_NOT is enabled, then NOT expression will match documents which don't match the subexpression.

XOR

expression XOR expression matches documents which are matched by one or other of the subexpressions, but not both. XOR is probably a bit esoteric.

Bracketed expressions

You can control the precedence of the boolean operators using brackets. In the query one OR two AND three the AND takes precedence, so this is the same as one OR (two AND three). You can override the precedence using (one OR two) AND three.

The default precedence from highest to lowest is:

+, - (equal) NEAR, ADJ (equal) AND, NOT (equal) XOR OR '+' and '-' A group of terms with some marked with + and - will match documents containing all of the + terms, but none of the - terms. Terms not marked with + or - contribute towards the document rankings. You can also use + and - on phrases and on bracketed expressions.

NEAR

one NEAR two NEAR three matches documents containing those words within 10 words of each other. You can set the threshold to n by using NEAR/n like so: one NEAR/6 two.

ADJ

ADJ is like NEAR but only matches if the words appear in the same order as in the query. So one ADJ two ADJ three matches documents containing those three words in that order and within 10 words of each other. You can set the threshold to n by using ADJ/n like so: one ADJ/6 two.

Phrase searches

A phrase surrounded with double quotes ("") matches documents containing that exact phrase. Hyphenated words are also treated as phrases, as are cases such as filenames and email addresses (e.g. /etc/passwd or president@whitehouse.gov).

Searching within a free-text field If the database has been indexed with prefixes on terms generated from certain free-text fields, you can set up a prefix map so that the user can search within those fields. For example author:dickens title:shop might find documents by dickens with shop in the title. You can also specify a prefix on a quoted phrase (e.g. author:"charles dickens") or on a bracketed subexpression (e.g. title:(mice men)).

Searching for proper names

If a query term is entered with a capitalised first letter, then it will be searched for unstemmed.

Range searches

The QueryParser can be configured to support range-searching using document values.

The syntax for a range search is start..end - for example, 01/03/2007..04/04/2007, \$10..100, 5..10kg.

Open-ended ranges are also supported - an empty start or end is interpreted as no limit, for example: ..2010-06-17, \$10.., \$..100, ..5kg.

Synonyms

The QueryParser can be configured to support synonyms, which can either be used when explicitly specified (using the syntax ~term) or implicitly (synonyms will be used for all terms or groups of terms for which they have been specified).

Wildcards

The QueryParser supports using a trailing '' *wildcard*, which matches any number of trailing characters, so *wildc* would match *wildcard*, *wildcarded*, *wildcards*, *wildcat*, *wildcats*, etc. This feature is disabled by default - pass `Xapian::QueryParser::FLAG_WILDCARD` in the flags argument of `Xapian::QueryParser::parse_query(query_string, flags)` to enable it, and tell the QueryParser which database to expand wildcards from using the `QueryParser::set_database(database)` method.

You can limit the number of terms a wildcard will expand to by calling `Xapian::QueryParser::set_max_expansion()`. This supports several different modes, and can also be used to limit expansion performed via `FLAG_PARTIAL` - see the API documentation for details. By default, there's no limit on wildcard expansion and `FLAG_PARTIAL` expands to the most frequent 100 terms.

Partially entered query matching

The QueryParser also supports performing a search with a query which has only been partially entered. This is intended for use with "incremental search" systems, which don't wait for the user to finish typing their search before displaying an initial set of results. For example, in such a system a user

would enter a search, and the system would display a new set of results after each letter, or whenever the user pauses for a short period of time (or some other similar strategy).

The problem with this kind of search is that the last word in a partially entered query often has no semantic relation to the completed word. For example, a search for "dynamic cat" would return a quite different set of results to a search for "dynamic categorisation". This results in the set of results displayed flicking rapidly as each new character is entered. A much smoother result can be obtained if the final word is treated as having an implicit terminating wildcard, so that it matches all words starting with the entered characters - thus, as each letter is entered, the set of results displayed narrows down to the desired subject.

A similar effect could be obtained simply by enabling the wildcard matching option, and appending a "*" character to each query string. However, this would be confused by searches which ended with punctuation or other characters.

This feature is disabled by default - pass `Xapian::QueryParser::FLAG_PARTIAL` flag in the `flags` argument of `Xapian::QueryParser::parse_query(query_string, flags)` to enable it, and tell the `QueryParser` which database to expand wildcards from using the `QueryParser::set_database(database)` method.

Metadata editing

Bookbuilder

Introduction

With the Bookbuilder you can customize the PDF and the EPUB of the texts distributed on an amusewiki instance. You can change fonts, dimension, etc. This article will cover the Bookbuilder feature in great detail. Anyway, the web interface should already give you an idea of what you can do.

Getting the texts into the bookbuilder

The prerequisite is that your browser has javascript and cookies enabled (otherwise the application can't store your selections).

To start using the Bookbuilder you first need to add one or more texts.

We will start covering the topic of a single text and all its options.

The Bookbuilder can be found on the navigation menu, with a book icon.

XXX

However, following the link will tell you that no texts has been added yet.

XXX

If you haven't already done so, you will have to pass the antispam question before proceeding.

So, find a text (`/random` could do for the purpose of this tutorial) and look at the download buttons.

The last two buttons with a big plus sign are the Bookbuilder buttons.

XXX

The first one is to add a whole text, the second one (with a list icon) is for selecting individual parts (this topic will be covered later).

Clicking on the plus button will add the text to the Bookbuilder.

XXX

You can continue browsing the site or visit the `/bookbuilder` page now, clicking the link in the navigation menu or the confirmation message.

Normally your selections are kept until you close the browser, and sometimes even longer (depending on the traffic on the site, if you clean the cookies, etc.). But let's assume they are kept while you don't close the browser, so you can browse the site freely, add more texts, and finally come back to build your files.

The Bookbuilder interface

XXX

The upper part of the page consists of a table which lists the texts added so far. You may want to browse the site and add more texts if you wish so, but for now let's stop here and see what the Bookbuilder has to offer.

Given that so far we have only one text, the "Move up" and "Move down" buttons have no use. Deleting a text will remove it from the Bookbuilder. "Clear the book list" will remove all of them, while "Reset all" will clear the list and restore the factory settings.

The first choice to do is the output format, which usually is PDF or EPUB.

PDF options

Feel free to experiment with the options. After you have built a text, the options are not lost, you can go back to the Bookbuilder page and rebuild the PDF changing some other options until you achieve the desired result.

Fonts

XXX

First of all, you may want to change the fonts used in the PDF. There is a link to the font preview page, </bookbuilder/fonts>, with the title "Please select the desired fonts", where you can take a look and see how they look like and if the character coverage is enough for your text (which should be enough for Latin characters, but it's a more delicate matter for, e.g., Russian).

The main font is what matters, while the mono font is relevant only for code listings.

Finally you can choose the font size, in the standard 9, 10, 11 and 12 points. Usually 10 or 11 is a good choice.

Page dimensions

The paper format choice is, of course, an important one and determines the page dimensions, *before any other processing like imposition*.

XXX

The menu consists of a dropdown menu with predefined dimensions (mostly standard) but also offers a way to set custom ones. Please note that to use the custom dimensions you need to set both the width and the height, otherwise the dimensions set in the dropdown menu takes precedence. When both width and height are selected, the dropdown will disappear, and will reappear resetting the dimensions to "Standard".

XXX

The division factor and the binding correction

The options presented here is probably cryptic, but it has a sane default. This is the way the text block is built:

XXX

The grid presented here shows a page built with a division factor of 9. The width and the height are divided in 9 equal parts, and 3 parts are used for the margins. Increasing the division factor leads to narrower margins, because it will always use 3 parts for the margins, while the text block would grow. More details can be found in the documentation of the backend used, KOMA-Script.

The menu offers division factors from 9 (wide margins) to 15 (narrow margins).

In the picture you will notice the binding correction gap. That is the amount of space eaten by the binding. If you are clipping a booklet, you may want to set it to 0, while if you're binding it, you have to take account of the paper that gets used by the binding, which is not visible at all and in the worst case scenario could make you book unreadable because the binding would eat the text as well (and then you can throw everything away...).

Table of contents options

Under normal circumstances, a text with sections or chapters generates a table of contents. However, you may want to disable it (to save paper or because you feel it's not needed): you can do so checking the "Never generate a table of content" checkbox.

If, instead, the text does not have sectioning, by default the text will start on the next page after the cover. Checking the "For texts without a table of contents, start the text on the very first page" checkbox would make the body starting at page 1.

Two side layout

XXX

This apparently innocent checkbox means a great deal.

If not checked, the text block would be centered, with equal left and right margins, and centered page numbering.

If checked, the layout will take account of left and right page, the inner margin being half of the outer margin, and page numbers on the outer margin. Also, a dropdown menu will appear, asking where to start a chapter, if on any page, or on a right-hand page only.

If you are going to produce a PDF for reading on screen, you want to choose reduced page dimension (A5 max), disable the two side layout and set the binding correction to 0.

If you are going to do a booklet or a book, you want to activate the two side option.

Running headings

The running headings, also known as headers, are the lines with usually author or title or section name which can be found on the top of a page.

Now, deciding what to put there can be tricky, because a random text doesn't have a fixed structure with sections and chapters, or the subtitle is missing, or the author is missing, so the dropdown menu prompts to some choices, also stating what happens if you do not have a two-sides layout (because usually you don't want the same text on the left header and on the right header).

Choose the safest option which works for your text.

The cover image

A word of warning: usually it's best to let the Bookbuilder take care of the inner of the book or booklet, and create the cover on a different file, because usually you want to print it on a different kind of paper, add some texts on the backside, etc.

Anyway, if you are going to produce a file to distribute on the internet for reading on screen, and

you want some graphics on it, you have the chance to do so adding a cover image.

XXX

You can get a preview of the file by selecting the file from your computer, scrolling down the page and hitting “Update”, which will save your settings.

Once you have uploaded it, you have the chance to replace it selecting another one or remove it checking the appropriate checkbox and updating again.

Another warning: the image by default will expand its width to the text-block width (i.e., will have margins). Now, if the width/height ratio is not suitable, it’s possible that some undesired results will follow, like the image going on the next page and so.

To control the image size you can set the width in percent. So, if the image is too high, reduce the width to a suitable size to make all the cover material fit nicely in the page. You will probably have to rebuild a couple of times.

Again, probably you want to create a cover yourself on a separate file.

At this point, if you are not going to print the file or if you are just printing on a plain A4, all you have to do is to press the “Build it” button on the bottom of the page.

XXX

The link under “Your file is ready” is the file for you, but you can also grab the LaTeX sources and attached files with the “Sources” link.

Imposition

If you don’t know what imposition means, you can read about it on wikipedia.

You can activate the imposition clicking on the “I want an imposed (ready to print) PDF” checkbox, and you will be presented with a wide range of schemas. Every schema is accompanied by an image which should make the output clear.

Be aware that the page dimensions you set before are altered. If in your schema on every sheet fit two pages, the width will double. So, if you set the page dimensions to A5, if you impose with the 2up schema, you will get a final physical size of A4.

XXX

The contextual help should be clear enough. If not so, please report it as an amusewiki bug.

If you are going to print a pamphlet, you want the **2up** schema, which also prompt for the signature size, defaulting to imposing the whole book on a single signature, meaning that the last page will be together with the first page on the first physical sheet. All you have to do is to clip in the middle, fold once and you have your booklet.

Anyway, you can also set the signature size, i.e. imposing the text in more booklets which then you have to bind together.

While most of the schemas have fixed signatures, the **2up** and **4up** schemas can have different signature sizes.

As we said, the default is to impose everything on a single signature. However, for larger books, this is not always desirable. You can set it to a fixed number of pages you decided, or let the system to compute the best signature size (i.e., the one which will generate less blank pages to fill the last signature). This is the “optimized signature” option. All the signatures will have the same size, but this size is not known in advance, and will be ranging from 40 to 80.

Again, feel free to experiment.

XXX

Also take a look at the checkbox at the bottom of the page: “Fill with blank pages before the last page if the signature is not full”. Say your book is going to have 10 pages, and you impose on a signature of 16 pages, this means you are going to have the 6 pages blank. If this checkbox is not checked, the blank pages will fall at the end. If you fold it, the last page is going to be empty.

However, given that amusewiki produces PDFs with a kind of back cover, it's nice to have it placed at the very end. So, in our example, pages 1-9 with the text, then pages 10-15 with blank pages, and finally page 16 with the original last page (which was p. 10, but now is shifted at the end). And this is what that option does.

Crop marks

Checking the “Add crop marks to the PDF” will reveal some other options.

XXX

Crop marks are the marks placed at the corners of a form to indicate where the page is to be trimmed.

If you are not going to trim the paper, you don't need it.

When using this option, please note that you are now setting the size of the logical page, not the physical paper dimensions. So the same logic as before, when we were setting the page dimensions apply.

Say you want to print a booklet with dimension 13x20cm, using the **2up** schema. You have to set the 130mm and 20mm when you choose the page dimension, while here you should set A5 (so it leaves some margins to cut), and will result in an A4 PDF.

Activating the crop marks also cause the cutting correction to be applied, so outer pages in the signature will have some inner space to compensate the paper thickness after the folding. You can fine-tune that with the “paper thickness” dropdown menu.

Be aware that the file needs some more time to be produced when you activate the crop marks.

Please note the crop marks near the margins

EPUB

Now that we got our first PDF, we can go back to the `/bookbuilder` page and instead of choosing PDF, we can choose EPUB. Please note that the PDF settings will not be forgotten.

XXX

When choosing the EPUB output, most of the options will disappear, leaving only the cover image options (same logic as the PDF) and an EPUB-specific option, which permits to embed fonts into it.

Please note that not all the devices and software which read EPUB support the font embedding, and the size of the EPUB with embedded fonts is going be much larger.

Anyway, if you want to create a nice EPUB with fonts of your choice, you have the chance to do so. Again, refer to `/bookbuilder/fonts` for a preview.

To create the file, hit the “Build it” button as we did for the PDF.

Collections

Amusewiki has the ability to merge texts together. You just have to add another text to the bookbuilder, with exactly the same procedure as before (the plus button on each page).

XXX

However, when building a collection, you are asked for some details. The only mandatory one is the title of the collection, but you can also specify author, subtitle, date (for the front page), source and notes (for the back page).

This gives you full control of what to put on those pages of the resulting file (both EPUB and PDF).

Partial selections

You can also choose to add only some sections of a text.

You can do so visiting the text and in the menu, and clicking instead of the plus button, the plus button with a list, which brings you to a new page where the text structure is shown and you can pick the pieces you are interested in.

XXX

XXX

Beware that when you have a complex structure, if you select only an higher level (in the example here, “The Muse markup”) without the lower levels, you are going to get only its title and the text until the first subsection. Anyway, the interface is going to select the subsections for you when you select an higher level. You are free to deselect them if you are not interested in those parts.

Clicking on the “Add the selected parts to the bookbuilder” will bring us back to the text, with a notice that we have added it.

XXX

If you now visit the /bookbuilder page, will see a link in the “Partial selection” column (with a “Yes”). If you visit it, it will bring you back to the text structure view, where you can edit and resubmit the selections.

Please note that doing so will add the text again (yes, you can have duplicates), so you have the remove the previous copy with the “Delete” button. New additions are naturally added at the bottom.

The rest of the interface works transparently as you were building a single text.

**Gestire un archivio con l'accesso al
collettore**

Best practices sull'uso degli utenti

Gli utenti biblioteca hanno la possibilità di creare utenti a loro volta. Sarebbe il caso che questi utenti di secondo livello venissero creati solo per persone fidate, e possibilmente con la scadenza (che può essere rinnovata). Soprattutto nel caso che quegli utenti possano lavorare sui metadati. Questo perché da un lato il senso di questo progetto è alimentare l'importanza degli spazi fisici, e dall'altro perché la collaborazione si basa sulla fiducia, ed interventi sconsiderati o superficiali sui metadati possono rovinare anni di lavoro e di cura in pochi clic. Siate parsimoniosi!

Best practices sui software di catalogazione

Ci sono molti software che possono essere utilizzati per facilitare il lavoro di catalogazione e l'interoperabilità con ALN. Presentiamo alcune possibilità, senza tralasciare il fatto che si può anche semplicemente compilare manualmente un foglio di calcolo. In ogni caso è molto utile avere una memoria scritta delle proprie regole di catalogazione, oltre che un diario del lavoro che viene fatto, anche cartaceo, in modo da non incorrere in errori di sintassi, metodo o sincronizzazione tra ciò che viene fatto localmente e ciò che viene catalogato.

Possibilità di intestazione

Vediamo ora, divise per schermata, le informazioni prioritarie da fornire affinché venga utilizzato il massimo potenziale del sito

Risultati di ricerca

Questi sono i campi principali che andrebbero riempiti, se possibile, per un livello minimo di completezza del testo

Autore

In questo campo ci va il nome di chi ha scritto il testo. Nel caso ci siano autori multipli, vanno separati da ”;”

La direttiva di inizio documento per Amusewiki è #author

Titolo

Come dice l'intestazione, questo è il campo del titolo

La direttiva di inizio documento per Amusewiki è #title

Sottotitolo

Il sottotitolo, se presente, va in questo campo

La direttiva di inizio documento per Amusewiki è #subtitle

Lingue

Molto importante, per poter distinguere i testi ed impostare le correlazioni tra le lingue, è che venga inserita la lingua del testo, possibilmente utilizzando l'ISO 639-2. L'elenco completo delle sigle può anche essere trovato nella repository del codice.

La direttiva di inizio documento per Amusewiki è #lang

Scheda del testo

La schermata di maggior dettaglio, invece, permette allo stato attuale di visualizzare i seguenti dati

Pubblicato (data di pubblicazione)

Anno di pubblicazione, in numero. Se corredato dal mese o da altri elementi testuali essi compariranno nel campo ”data e luogo di pubblicazione”. Ricordarsi, ovviamente, che i valori testuali non vengono tradotti automaticamente.

La direttiva di inizio documento per Amusewiki è #date

Città di stampa

Luogo di stampa, la sede della casa editrice

Editore

Nome della casa editrice

La direttiva di inizio documento per Amusewiki è #publisher

Collana

La collana in cui è inserito il testo

Descrizione Materiale

Dimensioni, peso, numero di pagine, stato documentale, tipo di rilegatura: tutto ciò che ha a che fare con la fisicità del libro descritto

Link esterno

Se può essere utile, il link ad una fonte esterna. Solitamente questo dato viene compilato automaticamente quando avviene l'importazione da una risorsa online

Shelf Location Code

Il luogo fisico in cui si trova il libro nella biblioteca, ovvero la codifica del posto a scaffale

La direttiva di inizio documento per Amusewiki è #slc

ISBN

Codice univoco ISBN, se presente

La direttiva di inizio documento per Amusewiki è #isbn

Il codice UNIMARC è 010 - a

Note

Dalla descrizione a qualsivoglia specificazione si ritenga utile aggiungere. Molto utile sarebbe inserire in questo campo il titolo originale del testo, anche per facilitare la creazione delle relazioni di traduzione tramite la ricerca direttamente sul sito

La direttiva di inizio documento per Amusewiki è #note

Dati ancora non utilizzati o ad uso interno dell'archivio

Non tutte le informazioni attualmente trovano posto e vengono visualizzate. Tuttavia, per il futuro, potrebbe essere utile predisporre alcuni campi da utilizzare per la catalogazione

Tipologia (es. libro, opuscolo, giornale...)

Una classificazione sommaria, divisa per macrocategorie, di quello che stiamo catalogando

Serie

Nel caso di periodici o di più volumi che fanno parte di un'unica opera più ampia, questo campo dovrebbe essere rappresentato dal nome della testata o dell'opera completa

La direttiva di inizio documento per Amusewiki è #seriesname

Sequenziale di serie

Ordinamento sequenziale all'interno della serie. **Attenzione, nel caso di periodici in cui la numerazione è basata sull'anno, questo dato può generare confusione, in quanto la numerazione deve essere assoluta**

La direttiva di inizio documento per Amusewiki è #seriesnumber

Sottocollana

Il sottoinsieme della collana

Cartaceo

Un identificativo ad uso interno degli archivi, per distinguere quei testi che sono presenti anche (o solo) in originale cartaceo da quelli esclusivamente digitali. Il valore dovrebbe essere un sì/no (booleano)

Tag

Categorie arbitrarie utilizzate dai singoli cataloghi. Non vengono importati anche perché differiscono troppo le lingue e la dimensione soggettiva per poterle unificare

Nazione di stampa

Lo Stato in cui è stato stampato il testo

Alcuni esempi di software e/o fonti di dati

Il codice che gestisce l'importazione dei dati può essere visualizzato a questa pagina. Più nel dettaglio, UNIMARC e MARC21.

abebooks (genera csv)

Calibre (genera csv o scansione albero)

Questo software è molto pratico per gestire principalmente file digitali (PDF, Epub). L'editing dei metadati è abbastanza semplice. La sincronizzazione dei file con ALN avviene attraverso la copia della struttura di cartelle che Calibre genera su uno spazio cloud a cui ALN ha accesso tramite chiave SSH e grazie alla lettura dei metadati importa e rende disponibile direttamente sul sito. Questa copia può essere resa molto intuitiva con l'uso di software FTP/SFTP quali la versione client di Filezilla o i comandi da terminale (qualcosa di simile a questo: `rsync -avhe 'ssh -p23' -delete /path1/ user@url:/path2`). La struttura delle cartelle generata da Calibre è la seguente:

1. cartella "nome_biblioteca"
2. cartella "nome_autore"
3. cartella "titolo + numero sequenziale"
4. file .OPF (Metadati), cover.jpeg e titolo.pdf/epub/txt (a seconda del caso; se ci sono più formati vi sarà un file titolo.formato per ogni tipo di file)

Piccoli accorgimenti sull'uso di Calibre

- L'albero di cartelle è dinamico e si aggiorna in automatico quando vengono modificati i metadati dall'interfaccia software di Calibre, quindi bisogna usare quei file manualmente il meno possibile, se no si rischia di compromettere il database. Inoltre, non si può lavorare via server: occorre inoltre che il database e la cartella della biblioteca siano su un solo computer/memoria esterna, e ci si lavori tramite un solo terminale alla volta.
- Quando viene cambiato il titolo viene rinominata in automatico la cartella nell'albero dei file. Questo può dare dei problemi con la sincronizzazione, perché si creano doppioni in quanto la cartella col nome nuovo viene aggiunta e la cartella col nome vecchio resta sul cloud. Prestare attenzione. Lo stesso avviene quando viene modificato l'autore di un testo. La cartella che viene rinominata, tuttavia, non è più quella del singolo testo ma quella che contiene le cartelle dei testi dello stesso autore
- Quando vengono modificati i metadati cambiano solo i file .OPF. Verificare che anch'essi siano aggiornati nel caso di piccole modifiche.
- Una volta creati i campi dei metadati, le biblioteche possono essere clonate mantenendo le stesse colonne e le stesse intestazioni. Un buon metodo può essere quello di avere una biblioteca di lavoro e man mano che si controllano e modificano i metadati copiare il libro nella biblioteca che viene sincronizzato, facendo attenzione ad utilizzare la funzione "elimina dopo la copia" che in automatico toglie la voce dalla biblioteca di lavoro. Ovviamente dei backup sono sempre utili.

koha (oai-pmh)

csv

Un file csv è un file che contiene i dati dei libri sotto forma di testo separato da virgole. Questo tipo di file viene solitamente generato da un foglio di calcolo.

Per far sì che l'importazione del file sia semplice, occorre tenere presente alcune nozioni di base.

- le intestazioni delle colonne devono essere consistenti tra tutti i file di una stessa biblioteca (ovvero le intestazioni devono essere uguali)
- le intestazioni devono essere solo nella prima riga, ovvero tutte le altre linee devono essere dati, non altre intestazioni o cambio di intestazione

amusewiki (oai-pmh)

Questo software è lo strumento migliore per la gestione dei testi editabili, delle trascrizioni e l'editing degli OCR. Amusewiki comunica con ALN direttamente tramite gli OAI-PMH. Si può dire tranquillamente dire che per quel che riguarda questo tipo di dati si tratta della fonte privilegiata di informazioni per ALN. Per un elenco completo dei tag e del loro uso, vedasi questa sezione del manuale.

vufind (oai-pmh)

Workflows

Eliminare un testo su Amusewiki

Per eliminare un testo, aprire l'editor ed aggiungere la linea "#DELETED Motivo" in cima, quindi salvare. Una volta approvata la modifica, il testo non sarà più visibile. Per eliminare definitivamente il testo andare dal menù utente sulla voce "testi non ancora pubblicati o rimossi" e premere purga. Questo eliminerà definitivamente il testo dalla collezione amusewiki.

Aggiungere un libro

Lavorare su Amusewiki

Molto utile è l'uso delle collezioni per dividere in case editrici, collane e sottocollane i libri e le antologie che si caricano, o per mantenere distinti fondi d'archivio che hanno anche nella propria suddivisione interna una fonte di informazioni.

Aggiungere una rivista o una collezione di riviste

Titolazione

Per nominare i singoli numeri di un periodico, attualmente da alcuni spazi è utilizzata la seguente forma:

Titolo - anno in numeri romani . numero sequenziale

Il numero sequenziale dovrebbe avere lo 0 per i numeri da 1 a 9 nel caso che la numerazione di quell'annata non sia superiore a 99.

00 da 1 a 9 e 0 da 10 a 99 se la numerazione arriva 999.

Se la numerazione non è divisa in annate ovviamente quel valore viene omesso. Questo perché altrimenti l'ordinamento avviene in questo modo: 1, 10, 11, 12 ... 2, 20, 21, 22 ... ecc. ecc.

Ad esempio:

Wohlstand für Alle - IV.07

L'Adunata dei Refrattari - XXXIX.21

**Avis de Tempêtes - 31/32 **

Lavorare su Amusewiki

Come primo passaggio, creare dal menù "Aggregations" il nome della serie. All'interno della serie, creare l'aggregazione (ovvero il singolo numero che compone la serie). L'aggregazione è una sorta di contenitore vuoto, al cui interno caricare i singoli articoli che compongono il numero della rivista o del giornale.

Questo sistema è valido anche per le antologie, semplicemente basta non assegnare a nessuna serie l'aggregazione che si crea

Mycorrhiza Project

Manuale d'uso

Dall'installazione locale Amusewiki al Bookbuilder, passando per il collettore Mycorrhiza

mycorrhiza.amusewiki.org