

Biloba STX

User's Guide

Version 1.0
7-Apr-2004

Contents

1	Preparation	4
1.1	About This Document	4
1.2	Basic Concepts	4
1.3	Test Environment	5
2	Introduction to Biloba	6
2.1	Your First Document	6
2.2	Paragraphs and Captions	6
2.3	Term Definitions and Quotes	7
2.4	Lists	12
2.5	Inline Formatting	14
2.6	Adding Images	15
2.7	Adding Example Text	17
2.8	Further Reading	17

List of Figures

2.1	Source: A Simple Section	7
2.2	Rendered: A Simple Section	7
2.3	Source: Nested Sections	8
2.4	Rendered: Nested Sections	9
2.5	Source: Term/Definition Pairs	10
2.6	Rendered: Term/Definition Pairs	10
2.7	Source: A Quotation	10
2.8	Rendered: A Quotation	11
2.9	Source: Nested Sections Restructured Using Special Forms	11
2.10	Rendered: Nested Sections Restructured Using Special Forms	11
2.11	Source: An Itemized List	12
2.12	Rendered: An Itemized List	12
2.13	Source: A Numbered List	13
2.14	Rendered: A Numbered List	13
2.15	Source: Two Lists	13
2.16	Rendered: Two Lists	13
2.17	Source: Inline Formatting	14
2.18	Rendered: Inline Formatting	14
2.19	Source: Hyperlinks	14
2.20	Rendered: Hyperlinks	15
2.21	Source: A Horizontal Rule	15
2.22	Rendered: A Horizontal Rule	15
2.23	Source: Adding an Image	16
2.24	Rendered: Adding an Image	16
2.25	Source: Preventing a Caption	16
2.26	Rendered: Preventing a Caption	17
2.27	A Simple C Program	18
2.28	Rendered: Adding a Program Listing	18

Chapter 1

Preparation

1.1 About This Document

This is the *User's Guide* to the Biloba document formatting system. It takes the reader on a guided tour through the process of creating documents with Biloba. Starting with the creation of a simplistic document, more and more features are explained as the reader progresses.

For details about the setup process, Biloba's internals or how to extend the system, please refer to the *Expert's Guide*.

Biloba is a non-interactive document formatting system specifically designed for documents on the Web. It was developed as a project completed as part of the requirements for the BSc. (Hons) Computer Studies by Viktor C. Pavlu under the supervision of Carlton McDonald at the University of Derby in the years 2003-2004.

1.2 Basic Concepts

Biloba takes a document in plain text format and converts it into the output format of your choice. Currently you can choose from the following formats:

- **XHTML** Extensible Hypertext Markup Language. The standard format for documents on the Internet.
- **TEX** A very popular file format in the scientific and academic communities. Can be converted to PDF (and other) files.
- **XML** Extensible Markup Language. A standard format for structured data.

- **debug, raw** Representation of the document's logical structure.

Most users would want their documents converted to XHTML.

1.3 Test Environment

For the purpose of this guide, we will use the *Biloba STX Test Harness* which can be found on the CD. It is a very simple environment that needs no configuration by the user — it can be started right away but needs to be copied from the CD to your local drive as temporary files have to be created which is not possible on the CD.

To start it, copy the directory `'test_harness'` to your local hard disk and execute `'start.bat'` which will start the Web server required to run Biloba. If the server was successfully started, you will see a window displaying the text

`'Copyright (c) 1991-2003 iMatix Corporation'`.

Do not close this window while using the test harness.

Your browser displays the *Biloba STX Test Harness* page (`redirect.html`). Go directly to the *Test Environment*, a page where you can enter STX documents for testing purposes. This is the environment used for this guide.

The way documents are formatted does not differ whether you use this simple pre-configured test environment or any other interface. Biloba can be configured to run embedded in your Web server or as a command line tool. The *Expert's Guide* tells you how to setup Biloba for your needs.

Chapter 2

Introduction to Biloba

2.1 Your First Document

Go to the test environment as described in the previous section.

Push the *Render my Document* button and you will see the document from the text area rendered into HTML. Hit your browser's *back* button to return to the text area where you can edit the source of the document. Modify the text and push the *Render my Document* button again. The rendered document has changed accordingly.

Whenever you push the *Render my Document* button, Biloba is invoked to transform your document. It parses the source in the text area and converts it into the format you have selected in the dropdown box left to the button. Have Biloba render your document into another format to get the idea.

2.2 Paragraphs and Captions

Biloba's Structured Text is organised around the concept of text elements. A text element consists of one or more non-empty lines that have the same indentation on each line. Indentation refers to the number of blanks at the beginning of a line.

A structured document consists of sections which may contain sub-sections which again may contain sub-sections and so on. Every section is introduced by a caption giving the name of the section followed by at least one text element that forms the body of the section.

Every text element after a caption that is indented two characters relative to the caption belongs to the body of that caption. Together, a caption and its body

Figure 2.1: Source: A Simple Section

Caption

This is a part
of /Caption's/ body.

So is this.

Figure 2.2: Rendered: A Simple Section

Caption

This is a part of *Caption's* body.

So is this.

elements form a section.

Here is an example:

"Caption" is the caption for the section and the two paragraphs "This is..." and "So is this." form the body of the section.

Try the example in the test environment to see how Biloba renders them.

Note that captions are not allowed to span multiple lines.

Sections and sub-sections can be nested:

Here we have a section with two paragraphs followed by three sub-sections each of which contains further paragraphs.

2.3 Term Definitions and Quotes

In the previous section we had an example of three sections that only consisted of a word and its definition. The word was put as the caption of the section and the section's body contained its definition.

As terms and their definition are a very common thing in documents, Biloba has a special form to express that something is such a term/definition pair.

By adding the information that something is a term/definition pair instead of a

Figure 2.3: Source: Nested Sections

Language

Ludwig von Wittgenstein once said "The limits of language are the limits of one's world."

Let's have a closer look at language.

Syntax

Syntax describes the structure of valid sentences in a language.

Semantics

Semantics describe the meaning of a syntactically correct sentence.

Pragmatics

Pragmatics is the meaning of a sentence put into the context of background knowledge and experience.

Figure 2.4: Rendered: Nested Sections

Language

Ludwig von Wittgenstein once said "The limits of language are the limits of one's world."

Let's have a closer look at language.

Syntax

Syntax describes the structure of valid sentences in a language.

Semantics

Semantics describe the meaning of a syntactically correct sentence.

Pragmatics

Pragmatics is the meaning of a sentence put into the context of background knowledge and experience.

Figure 2.5: Source: Term/Definition Pairs

```
Syntax -- structure of valid sentences in a language.  
Semantics -- meaning of a syntactically correct sentence.  
Pragmatics -- meaning of a sentence put into the ...
```

Figure 2.6: Rendered: Term/Definition Pairs

```
Syntax  
    structure of valid sentences in a language.  
Semantics  
    meaning of a syntactically correct sentence.  
Pragmatics  
    meaning of a sentence put into the ...
```

common caption followed by a paragraph, the information can be presented in a more succinct way and will therefore be easier to convey. If all term/definition pairs are formatted using this special form, it is also possible to set them off from the rest of the text and thus making it easier for the reader to find a certain definition.

The special form for term/definition pairs looks as follows:

Biloba has special forms for many common text elements in order to give them additional meaning.

Another special form is used for quotes:

Here the left part must be enclosed in quotation marks, otherwise the parser will identify the quote as a very long term that needs to be defined.

Using the special forms we can now re-structure the example from the previous section as follows:

In terms of structure, the result is quite different from the first example. We only have one section instead of three and instead of simple paragraphs we have logically distinct text elements:

Figure 2.7: Source: A Quotation

```
"The limits of language are  
the limits of one's world." --Ludwig von Wittgenstein
```

Figure 2.8: Rendered: A Quotation

*"The limits of language are the limits of
one's world."*
- Ludwig von Wittgenstein

Figure 2.9: Source: Nested Sections Restructured Using Special Forms
Language

"The limits of language are
the limits of one's world." --Ludwig von Wittgenstein

Let's have a closer look at language.

Syntax -- structure of valid sentences in a language.
Semantics -- meaning of a syntactically correct sentence.
Pragmatics -- meaning of a sentence put into the ...

Figure 2.10: Rendered: Nested Sections Restructured Using Special Forms

Language

*"The limits of language are the limits of
one's world."*
- Ludwig von Wittgenstein

Let's have a closer look at language.

Syntax
 structure of valid sentences in a language.
Semantics
 meaning of a syntactically correct sentence.
Pragmatics
 meaning of a sentence put into the ...

Figure 2.11: Source: An Itemized List

- item
- item that spans more
 than one line
- item

Figure 2.12: Rendered: An Itemized List

- ◆ item
- ◆ item that spans more than one line
- ◆ item

- a quote,
- a paragraph, and
- three term/definition pairs

You should try to use the special forms whenever possible to implicitly add this information to your document which otherwise would be lost. It helps to make your documents more accessible to your audience.

2.4 Lists

There are two kinds of lists in Biloba:

- itemized or bullet lists (such as this one), and
- numbered lists

An item of a bullet list is introduced by a ”- ” sequence in front of it.

As we can see in the example, list items are allowed to span multiple lines as long as the lines are well aligned with another.

Numbered list items are introduced by a ”1) ” sequence in front of them.

An empty line ends a list. Consequently, these are two lists, though the author might have intended otherwise:

Figure 2.13: Source: A Numbered List

- 1) item number one
- 2) item number two
- 3) and finally, item number three

Figure 2.14: Rendered: A Numbered List

1. item number one
2. item number two
3. and finally, item number three

Figure 2.15: Source: Two Lists

- 1) item number one
- 2) item number two

- 3) and finally, item number three

Figure 2.16: Rendered: Two Lists

1. item number one
2. item number two

1. and finally, item number three

Figure 2.17: Source: Inline Formatting

```
/emphasis/  
*stronger emphasis*  
  
''sample text''  
  
~text that needs correction~  
_text that replaces corrected text_
```

Figure 2.18: Rendered: Inline Formatting

emphasis

stronger emphasis

sample text

~~text that needs correction~~

text that replaces corrected text

2.5 Inline Formatting

Inline formatting is sometimes added to *emphasize* words that need to be set off from the rest of the sentence. Do not overuse this feature, otherwise the whole effect of emphasis gets lost in chaos.

This is how it is done in Biloba:

Sample text is used to mark something as an example or to signal that this text has to be typed in or is the output of a program.

Text that needs correction and the accompanying text that replaces corrected text are used to denote that something was faulty and should be corrected accordingly.

Figure 2.19: Source: Hyperlinks

```
[http://www.google.com]  
[search google>>http://www.google.com]
```

Figure 2.20: Rendered: Hyperlinks

<http://www.google.com> search google

Figure 2.21: Source: A Horizontal Rule

Text between square brackets is used to create hyperlinks. The text between is interpreted as the target. Optionally you may add text to a link as illustrated in the last row of the example. The given text will be turned into a link to the given location. If no text is given, the location itself is displayed as link text.

A line that only consist of dashes will be rendered as a horizontal separator.

2.6 Adding Images

Adding images to a document is done with

where the first line represents the caption of the image. The line without the hash sign ‘#’ in front gives the path to the image. The ‘#type:image’ line is required so that Biloba knows it should render this figure as an image.

Note that these lines must be indented by two spaces relative to the current level. If the line immediately before this indented text element was a single line, it would be converted into a caption and these lines were interpreted as normal text in a paragraph. Therefore it is essential to ensure that text elements immediately before an image consist of at least two lines. If splitting the line to span two lines is not an option you can add a single ‘.’ character as the only character to be found in the second line. The ‘.’ will not produce any output but ensures that Biloba does not turn the line before into a caption.

Here is an example:

Without the guiding dot, ”Consider these leaves” would become a caption for obvious reasons.

Figure 2.22: Rendered: A Horizontal Rule

.....

Figure 2.23: Source: Adding an Image
#Image of Ginkgo Leaves (Tamara Crupi, September 1996)
#type:image
../biloba.jpg

Figure 2.24: Rendered: Adding an Image



Fig. 1: Image of Ginkgo Leaves (Tamara Crupi, September 1996).

Figure 2.25: Source: Preventing a Caption
Consider these leaves:

```
.  
#Image of Ginkgo Leaves (Tamara Crupi, September 1996)  
#type:image  
../biloba.jpg
```


Figure 2.26: Rendered: Preventing a Caption
Consider these leaves:



Fig. 1: Image of Ginkgo Leaves (Tamara Crupi, September 1996).

2.7 Adding Example Text

You can make Biloba retain text just as you entered it and have it copied into a figure. No further formatting is applied to this text element.

This is achieved in the same way as we did with images, except that you must omit the `#type:image` line.

The first line not starting with a hash sign `#` indicates the beginning of text that should be preserved. It extends until the next line with "normal" indentation.

This is particularly useful when describing external data such as program listings or printouts.

2.8 Further Reading

We have now seen all of Biloba's forms required to create documents. Look at the source of the documents part of the test harness for additional examples. Learning from examples can be very effective.

Figure 2.27: A Simple C Program

```
#include <stdio.h>
#define S "Hello, World\n"

int main(int argc, char **argv)
{
    exit(printf(S) == strlen(S) ? 0 : 1);
}
```

Figure 2.28: Rendered: Adding a Program Listing

```
#include <stdio.h>
#define S "Hello, Worldn"

int main(int argc, char **argv)
{
    exit(printf(S) == strlen(S) ? 0 : 1);
}
```

Fig. 1: A Simple C Program.

If you still have questions you should have a look at the *Expert's Guide* to Biloba which covers features you would not require for everyday use.