

OpenBerg e-Book Reader design and rationale

David Teller

Abstract. Edition of books has changed. Ten years ago, there were only two on-line publishers, both of them non-profit organisations. Nowadays, a number of companies provide one-size-fits-all on-line shops for on-line publishers or on-line self-published authors. On-line publications, however, tend to fall in one of the following categories

- project Gutenberg
- webpages
- proprietary formats
- open formats restricted by patents to proprietary readers.

The OpenBerg project aims at providing free, open-source and open-standard tools for digital publication (a.k.a. e-Publishing), for use by authors, readers and editors. In particular, our main effort is the development of OpenBerg Reader, a cross-platform, multi-format, e-Book reader targetted towards textbooks, exercise books and academic content.

This document describes the key notions of OpenBerg Reader, the choices we made regarding our tools and platforms, as well as the current status of the implementation and our plans for the future. The objective of this document is to serve as a reference for the developers of OpenBerg

Contents

Contents	iii
1 Introduction	1
1.1 The OpenBerg Project	1
1.2 Our objective	3
1.3 Contents of this document	3
2 Tools and design choices	4
2.1 Formats	4
2.2 Platform	6
2.3 Compiler and build system	8
2.4 Licensing	9
3 Modules	10
3.1 Overview	10
3.2 The installer	10
3.3 Cross-platform UI	11
3.4 Platform-specific UI	14
3.5 OpenBerg Core	14
3.6 Extending OpenBerg	22
4 Conclusion	23
4.1 The work so far	23
4.2 One Laptop Per Child	23
4.3 The future	25
Bibliography	27
A The team	29
A.1 Senior members	29
A.2 Junior members	29
A.3 Support sta	29
A.4 Former senior members	29
B RoadMap	30
B.1 Features	30
B.2 Estimated calendar	32
C Build instructions	34
C.1 Obtaining the code – anonymously	34
C.2 Obtaining the code – as a developer	34

iv	<i>OpenBerg e-Book Reader</i>	<i>design and rationale</i>	
	C.3	Cross-platform build system	35
	C.4	Unix build system	38
	D	Licenses	41
	D.1	GNU Public License, version 2	41
	D.2	GNU Lesser General Public License, version 2.1	48

1 Introduction

It's about digital publications and the future of books, nothing less.

During the past ten years, edition has changed more than during the previous century. Not only are people less interested in traditional paper books and journals, new means of self-publication are now within the reach of anyone, even with limited technical skills, through websites, wikis, blogs, PDFs or even attached Word documents. There's even an open-

and the Swing Toolkit as implementation target has the same drawbacks as in the case of OSoft.

Comparison with Plucker Plucker [8] permits under-average pagination, a little degree of library management, and it is tied to PalmOS.

1.2 Our objective

OpenBerg Reader has recently been selected as a potential candidate for inclusion in project One Laptop Per Child [6], along with ThoutReader and EVince. If this happens, OpenBerg Reader will be distributed to millions of children of Third-World countries, and will serve as support for reading textbooks and the Wikipedia.

Our objective is to strive to be effectively selected. In order to do so, we are currently working on adding features to OpenBerg Reader and re-working features to make them usable on low-powered laptops with either small resolutions or grayscale display.

We believe that OpenBerg Reader will be judge acceptable when

- the current memory-related problems of the Mozilla platform are solved
- OpenBerg Reader is feature-complete with respect to Milestone 4 (see Section B.1)
- OpenBerg Reader is well-tested.

1.3 Contents of this document

This role of this document is to serve as a reference regarding the design of OpenBerg. In chapter 2, we present the tools used in the development of OpenBerg. In chapter 3, we detail the modules of OpenBerg. We conclude in chapter 4 by a review of our work, a discussion on the adequation of OpenBerg Reader with OLPC's goals and our plans for the future. Appendix 1 briefly presents the team, Appendix 2 details our Roadmap, Appendix 3 contains build instructions and Appendix 4 displays the licenses used in the project.

2 Tools and design choices

2.1 Formats

Most e-Book viewers are mono-format. Some, as EVince, are multi-format, insofar as they have to reimplement some of their features (e.g.

\LaTeX The \LaTeX format is actually a powerful programming language intended for compilation before display. It offers extremely rich pag-

- Gtk Very good toolkit, cross-platform, supporting internationalisation, open-source, but its display of XHTML + CSS is extremely limited.
- Gtk + WebKit This addition permits XHTML + CSS rendering, but WebKit hasn't reached alpha version yet (and didn't even exist when we started working on OpenBerg).
- wxWidgets wxWidgets (formerly wxWindows) is an open-source cross-platform meta-toolkit able to map to Gtk, MFC, Cocoa and others. Its support for internationalisation is limited and its XHTML component is extremely limited.
- wxWidgets + wxMozilla This addition permits XHTML + CSS ren-

2.4 Licensing

The matter of licensing is an important problem. Indeed, while we wish OpenBerg to be free software and to stay free software, it might be interesting to extend OpenBerg to proprietary formats. In fact, we have been approached by three publishers hoping that they could use OpenBerg on proprietary platforms and/or with proprietary formats, possibly with digital signatures or with so-called Digital Rights Management.

At the moment, this question is not fully resolved. Our plans are to release everything under the Gnu General Public License (see Section D.1), double-licensing the XPidl interfaces under the Gnu Lesser General Public License (see Section D.2). This choice should give us the maximal freedom, while ensuring that our work will stay free software and while giving publishers the possibility of adding new input formats. Note that a consequence of this choice is that publishers will be able to customise the user interface with proprietary extensions through the use of Mozilla's extension mechanism (mostly with ui overlaying techniques) but not by altering the source code of OpenBerg Reader.

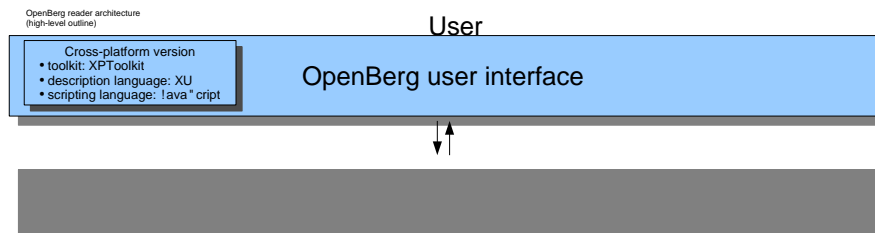


Figure 1: High-level outline of OpenBerg

3.3 Cross-platform UI

3.3.1 Overview

Figure 2 presents an overview of the structure of OpenBerg's cross-platform UI.

The cross-platform UI is composed exclusively of cross-platform code and data, i.e.

- XUL for the description of the user-interface itself
- XBL for the definition of XUL components
- JavaScript to add behaviours to the UI
- CSS for the styling of components
- PNG for icons
- Java/JavaScript string bundles, as `.properties` files, containing the internationalised, human-readable, texts for use in JavaScript code
- DTD for the definition of entities, containing the internationalised, human-readable, texts for use in XUL code.

Modules

13

may be referenced as `chrome://openberg/content/...`

3.4 Platform-specific UI

At the moment, one platform-specific UI is under development, for Windows Mobile, on top of Minimo. This platform is currently at an early stage of specifications.

3.5 OpenBerg Core

3.5.1 Overview

The core is developed exclusively in C++, using Mozilla's XPCOM libraries. Among other things, this means that the code should contain no call to libc. The objective of this being to make OpenBerg as platform-independent as Mozilla itself. Note that this limitation should not be constraining as Mozilla itself is built on top of

-

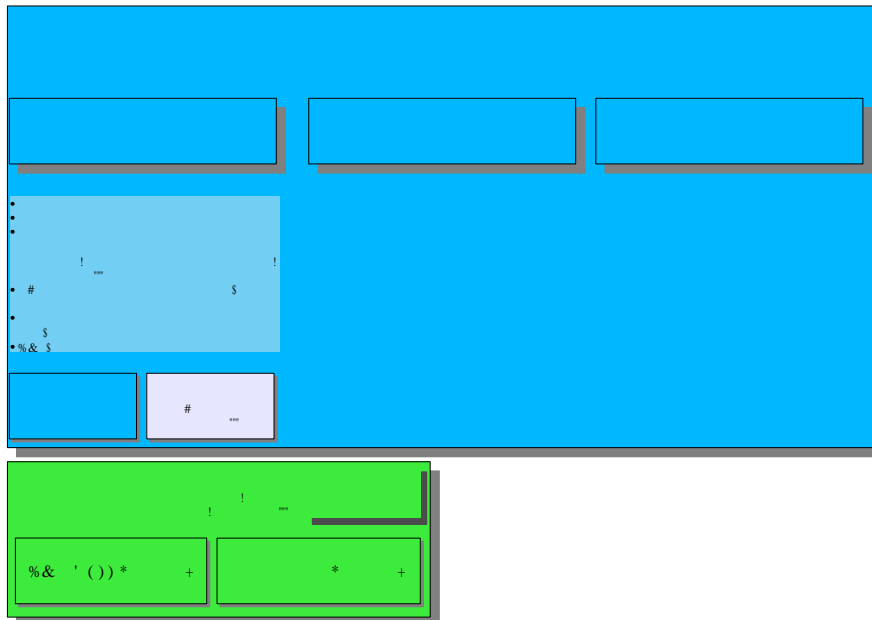


Figure 3: OpenBerg’s core

obl, while Mozilla’s interfaces see their name prefixed with nsI. Examples: oblBook, nsIStreamListener.

Functions In the absence of namespace and/or language-level module features, functions (by oppositions to methods) should be regrouped inside classes, as static methods.

Mozilla guidelines We respect Mozilla’s portability guidelines, as defined in [20]

Mozilla guidelines We respect Mozilla’s coding guidelines are [10]

Everything else When they are not in contradiction with the above, we use Ted Ho’s C++ coding guidelines, as expressed in [11]

3.5.3 Loader

Overview

The loader is the first unit of OpenBerg to be loaded and the last to be unloaded. It is in charge of initialisation of other units, of components and registration of components wrt XPCOM.

Details

From the point of view of the Mozilla platform, the loader is a nsI Module, that is a structure which can dynamically be loaded and unloaded and is associated with an entry and an exit point. Semantically, this is similar to the notion of component.

Indeed, the Loader is itself loaded dynamically whenever the UI attempts to instantiate any of the components of OpenBerg. The task of the loader is to overview

- initialise debugging/error-reporting structures
- adapt the configuration of OpenBerg to the platform (at the moment, FireFox or XULRunner)
- overview instantiation of OpenBerg's components – from the point of view of the Mozilla platform, the Loader contains a nsI Factory per component.
- overview the unloading of OpenBerg.

The loader is defined as class `obOpenbergModule`, by file `openberg.cpp`.

Dependencies

This unit depends on

Library Loader initialises Library and permits instantiation of all its components.

Annotations Loader initialises Annotations and permits instantiation of all its components.

Find Loader initialises Find and permits instantiation of all its components.

Utilities Loader initialises Utilities.

3.5.4 Library

Overview

The Library unit abstracts the format of e-Books, contains the low-level primitives used by the UI to trigger the loading of a book, warn listeners of the completion of book loading, or access the per-book storage.

Detail

Warning 1. *All functions, methods, etc. of the Library unit must be invoked from the UI thread. Ensuring that a call is executed in the UI thread is the task of interface nsIProxy, provided by the Mozilla library.*

Access to books The first role of the Library unit is to permit access to books regardless of the format in which they are stored. To perform this, the Library unit relies on plug-ins implementing the asynchronous loading of book formats and on-the-fly conversion to OEBPS. Components adding input formats – or *Book Loaders* – implement interface `oblBook` and must be registered with respect to the mime types they can interpret. They are then loaded dynamically when a request for the opening of such a book is emitted. Loading of books is asynchronous, i.e. requests should be accompanied with a listener (a.k.a. continuation). At a later point, once the book is ready for consultation, the listener will be triggered by the Library unit. Note that the listener may potentially be triggered from any thread and should therefore usually be protected by a `nsIProxy`.

All functions related to loading are accessed through the *Book Librarian*, itself a component implementing interface `oblBookLibrarian`.

Access to book storage The second role of the Library is to permit access to a per-book storage, or *book repository*. This storage is intended to contain

- book-specific preferences (e.g. stylesheet overrides)
- book-specific annotations (see Section 3.5.5)
- a cache for book metadata (e.g. chapter's titles)

Book repository-related functions are usable as soon as a book is fully loaded and until the closure of `OpenBerg`. They are accessible through the *Persistence Service*, itself a component implementing interface `oblPersistenceService`.

At the moment, these functions make heavy use of Mozilla's preference system. This is expected to change in a future version.

Access to library storage The third role of the Library is to permit access to a library storage, or *main repository*. This storage is intended to contain

- Book information from `http://www.mozilla.org/`

Dependencies

This unit depends on

Utilities Uses miscellaneous functions of Utilities.

3.5.5 Annotations

Overview

The Annotations unit contains the low-level primitives used by the UI to manage annotations to the text, i.e. undoable modifications performed by the user and transparently stored in a different file.

Detail

The Annotations unit is currently being designed and will most likely be the object of a future report by itself.

The role of this unit is to provide the low-level primitives required by the application and deapplication of user annotations to a text. By annotations, we mean features such as

- bookmarking
- underlining
- highlighting
- user-provided hyperlinks
- adding comments
- in the future, voice annotations, etc.

The associated primitives are

- DOM-based transformation of a XML document to add an annotation
- DOM-based transformation of a XML document to remove an annotation
- localisation of a position in the text – independently from the set of annotations added in the mean time
- (de)serialisation of a position in the text.

Let us consider an extract of a book, running as follows:

This is an `important` text.

A user may wish to emphasise the whole sentence, hence resulting in

This is an important text.

or

<

Questions:

- How should a reference to the occurrence of 2 be represented in memory ? Note that this reference should stay valid even if other annotations are added or removed.
- What should happen if the hyperlink is removed while it is highlighted ? Should the reference be invalidated ?
- What should happen if a third hyperlink appears during the highlighting ? What should happen if that third hyperlink appears inside the highlighted area ?
- What should happen if the highlight is removed afterwards – and how can the annotation be stored in a manner which will permit reloading it at the next opening of the chapter and restoring it to the correct position ?

One possible answer to this is to not let search find occurrences of annotations. This, however, is an important limitation.

Note that the order of removal problem appears also without the find feature. Let us consider the following extract

```
This is
<span class=" user:some_annotation ">
  an important
</span>
text .
```

Let us now assume that the user intends to add an overlapping annotation.

```
This is
<span class=" user:some_annotation ">an
  <span class=" user:other_annotation ">important</span>
</span>
<span class=" user:other_annotation " continued=" true">
  text
</span>.
```

Questions:

- what is the correct representation in the case of overlapping annotations ?
- what should happen if the user decides to remove the instance of some_annotati on before that of another_annotati on ?

One possible answer to this is to forbid overlapping (including stacked)

annotations, or to force the order of removal. Both choices, however, will prove disconcerting for the end-user.

Most likely, this implementation will stay in the future versions of OpenBerg Reader as

4 Conclusion

4.1 The work so far

So far, the bulk of the work we have provided was the design and implementation of OpenBerg's architecture, from the choice of the tools to the two build systems to the organisation of modules and of files inside modules. The bulk of the implementation were the prototype cross-platform

Done, in part	<i>User definable "style sheets" to control font size and type face, etc.</i>
Done	<i>Display graphics in various formats; both vector and bitmap formats.</i>
Working on	<i>Display complex formulae.</i>
Done	<i>Link in multimedia elements such as sound or video files.</i>
Working on	<i>Bookmarking, preferably sharable via network.</i>
Working on	<i>Annotations, preferably sharable via network.</i>
?	<i>Link to external programs for coursework such as worksheets, labs, etc. which can be forwarded BY the teacher to the students, and sent back to the teacher once the homework is done.</i>
Done	<i>Human readable, yet efficient markup language.</i>
Done	<i>Compressed files (easily decompressed to access original files).</i>
Done	<i>Simple, intuitive interface.</i>
Done	<i>Localizeable.</i>
Untested	<i>Supports complex-text language rendering (for example, arabic and thai).</i>
Supported	<i>A standardized schema of metadata for categorizing work.</i>
Done, in part	<i>Sorting on any of the fields in the schema, plus system generated data like last page read, etc.</i>
?	<i>Full-text search capability across all titles - MUST be speedy so indexing will be necessary.</i>
Planned	<i>Network access to larger library to allow downloads of additional titles.</i>

Most of these features should be integrated in the Milestone 3 of OpenBerg. A more detailed list of planned features is available in Chapter B.

4.3 The future

The team has recently grown with the addition of

- a XUL developer working on livesearch
- a C++ developer working on annotations
- a newsmaster.

We hope that this new blood will increase the dynamism – and the speed of development – of OpenBerg Reader. Our current objective is to release Milestone 2 of OpenBerg by June 2006, with the addition of

live search and comprehension of Gutenberg e-Books. That day, we will consider officially that OpenBerg has become a useful project.

Bibliography

- [1] Adobe reader official page, . URL <http://www.adobe.com/products/acrobat/readermain.html>.
 - [2] éditions Vigdor's home page, . URL <http://www.vigdor.com>.
 - [3] Microsoft reader official page, . URL <http://www.microsoft.com/reader/default.asp>.
 - [4] Mozilla developer connection: Extensions, . URL <http://developer.mozilla.org/en/docs/Extensions>.
 - [5] The oebps faq. URL <http://www.textartisan.com/oebfaq/oebps.html>.
 - [6] One laptop per child, . URL <http://laptop.media.mit.edu>.
 - [7] Osoft's thout reader official page, . URL <http://www.osoft.com>.
 - [8] Pluckr official page, . URL <http://www.plkr.org>.
 - [9] David Boswell, Brian King, Ian Oeschger, Pete Collins, and Eric Murphy. *Creating Applications with Mozilla*. O'Reilly, 2002. URL <http://books.mozdev.org>.
 - [10] Alec Flett and Akkana Peck. Mozilla coding style guide. Technical report, 1998. URL <http://www.mozilla.org/hacking/mozilla-style-guide.html>.
 - [11] Ted Ho . C++ coding standard. URL <http://www.possibility.com/Cpp/CppCodingStandard.html>.
 - [12] Richard M. Stallman. *Free Software, Free Society: The Selected Essays of Richard M. Stallman.*, chapter The Right to Read. Free Software Foundation, 2002. URL <http://www.gnu.org/philosophy/right-to-read.html>.
- [(Da)27(v)1(id)-0Td[(tl07)-51(l)y9h2//wzin[(tl06ns,(.))Tm510()1(onit.)]TJ/F219)]TJ2

- [15] unknown author. Open ebook publication structure specification. Technical Report 1.2, Open eBook Forum, 2002. URL <http://www.openebook.org/oebps/oebps1.2/index.htm>.
- [16] unknown author. *Adobe Reader SDK FAQ*, chapter Licensing and Distribution. Adobe Solutions Network, 2005. URL http://partners.adobe.com/public/developer/en/acrobat/sdk/pdf/intro_to_sdk/DeveloperFAQ.pdf.
- [17] Jimmy Wales, Larry Sanger, Ben Kovitz, and maybe a million others. *Wikipedia*. The Wikipedia Project, 2001. URL <http://www.wikipedia.org/>.
- [18] John Walker. The digital imprimatur. *Knowledge, Technology, Policy*, 16(3), 2003. URL <http://www.fourmilab.ch/documents/digital-imprimatur/>.
- [19] Karl Wick and maybe a million others. *Wikiversity*. The Wikipedia Project, 2003. URL <http://en.wikibooks.org/wiki/Wikiversity>.
- [20] David Williams, Scott Collins, Christopher Blizzard, and David Baron. C++ portability guide. Technical Report 0.8, 1998. URL <http://www.mozilla.org/hacking/portable-cpp.html>.

A The team

A.1 Senior members

(aka people who have already contributed some code, by order of ancienty)

David Teller aka Yoric, project manager. In this project, does a bit in everything. In real-life, post-doctoral research fellow, working on concurrent programming languages and static analysis of programming languages.

Nick Thomas aka *cf*, cross-platform build system maintainer, Windows port manager.

Vincent Bernardi aka Enzo, Mac OS X port manager.

A.2 Junior members

(aka people who have already contributed some enthousiasm)

Abbey (real name unknown) C++ developer, Windows Mobile port manager.

Giuseppe de Francesco C++ developer, working on bookmarks and annotations.

Claudio Bernardes aka *obs*, XUL developer, XULRunner port manager.

Julien Appert , XUL developer, working on the cross-platform UI.

A.3 Support sta

Christopher Lannoo webmaster.

A.4 Former senior members

Philippe Gervais aka *Phyxie*, CVS manager, in charge of XML preferences. Researcher in the field of physics.

B RoadMap

B.1 Features

This roadmap is not necessarily representative of the time necessary to

RoadMap

(continued from last page)

	M1	M2	M3	M4	Beyond	Current
JavaScript	Works	Works	Works	Works	Works	Works
Java applets	Works ¹	Works ¹	Works ¹	Works ¹	Works ¹	

(continued from last page)

	M1	M2	M3	M4	Beyond	Current
Navigation						
Along the spine	Works	Works	Works	Works	Works	Works
Out-of-order spine		Works	Works	Works	Works	Works
Along tours			Works	Works	Works	
Along guides			Works	Works	Works	
UI						
Full-screen browsing		Works	Works	Works	Works	
Zooming		Works	Works	Works	Works	Works
Tabbed browsing			Works	Works	Works	
Preferences dialog			Works	Works	Works	
Custom style-sheets			Works	Works	Works	
Mouse browsing	Works	Works	Works	Works	Works	
Keyboard browsing		Ongoing	Works	Works	Works	Ongoing
Help system	Ongoing	Ongoing	Works	Works	Works	Ongoing
Scroll-based display	Works	Works	Works	Works	Works	Works
Page-based display					Works	
Ok on Color Display	Works	Works	Works	Works	Works	Works
Ok on 1024x768+ Display	Works	Works	Works	Works	Works	Works
Ok on Grayscale Display		Ongoing	Works	Works	Works	Ongoing
Ok on 640x480 Display		Ongoing	Ongoing	Works	Works	Ongoing
Print / export						
Copy as text	Works	Works	Works	Works	Works	Works
Copy as XHTML	Works	Works	Works	Works	Works	Works
Printing				Works	Works	
Others						
book: protocol		Works	Works	Works	Works	
External tools (not part .919penBergeader)						
Conversion from L ^A T _E X		Works	Works	Works	Works	
Conversion from LIT				Works	Works	

B.2 Estimated calendar

Milestone 2 alpha April 2006.

Milestone 2 beta May 2006.

Milestone 2 release June 2006.

Milestone 3 alpha August 2006.

Milestone 3 beta September 2006.

Milestone 3 release October 2006.

Milestone 4 alpha December 2007.

Milestone 4 beta January 2007.

Milestone 4 release February 2007.

OpenBerg 1.0 March 2007.

C Build instructions

OpenBerg Reader has two different build systems. A fast and mostly painless Unix build system, detailed in Section C.4 and a slow and painful cross-platform build system, based on Mozilla's build system and detailed in Section C.3. If you have the choice, use the Unix build system.

C.1 Obtaining the code – anonymously

1. Get a computer with cvs or install cvs on your computer. It's a tool used to manage source code and share it between developers.
2. Open a shell (or a command line, or a dos session, or whatever is the equivalent on your operating system).
3. In that shell, go to the directory in which you wish to download the sources.

4. Enter the command

```
cvs -d: pserver: anonymous@cvs.sourceforge.net: /cvsroot/openberg login
```

5. When prompted for a password, just press Enter

6. Enter the command

```
cvs -z3 -d: pserver: anonymous@cvs.sourceforge.net: /cvsroot/openberg co -P proj e
```

7. If you receive an error message, chances are that the server is busy, try again a bit later
8. Once the operation completes without error messages, you have succeeded in getting the source of OpenBerg. Congratulations.
9. If you later wish to get updates to that source, open a shell / command line / dos session, go to the same directory and write (on one line)

```
cvs update
```

Note that there are graphical tools which you can use to access cvs without having to write down instructions. In our experience, these tools

2. Get a SourceForge account.
3. Be accepted as a developer of OpenBerg.
4. Open a shell (or a command line, or a dos session, or whatever is the equivalent on your operating system).
5. In that shell, go to the directory in which you wish to download the sources.
6. Enter the command
`export CVS_RSH=ssh`
7. Enter the command
`cvs -z3 -d:ext:<youraccount>@cvs.sourceforge.net:/cvsroot/openberg co -P projects/openber`
8. When prompted for a password, enter your SourceForge password.
9. If you receive an error message, chances are that the server is busy, try again a bit later.
10. Once the operation completes without error messages, you have succeeded in getting the source of OpenBerg. Congratulations.
11. If you later wish to get updates to that source, open a shell / command line / dos session, go to the same directory and write (on one line)
`cvs update`

Note that there are graphical tools which you can use to access cvs without having to write down instructions. In our experience, these tools don't work. But you're free to try.

C.3 Cross-platform build system

This section describes a the cross-platform build mechanism for OpenBerg. This mechanism should work on all platforms which support Mozilla, with the exact same prerequisites as for compiling Mozilla. If you have any choice, you should rather use the method described in Section C.4, as it is incredibly faster (i.e. a matter of minutes to set it up instead of hours, a matter of minutes to compile, instead of fifteen minutes).

If you have any problem with the compilation of OpenBerg, a good place to ask for help is

<http://www.openberg.org/contact.php>.

Please make sure that this is an OpenBerg problem and not a Firefox problem, as we can probably not help you with the compilation of Firefox.

C.3.3 Build OpenBerg's installer

This is an optional but useful step, especially if you already have a Firefox profile. This step is optional but useful, especially if you already have a Firefox profile.

```
export NSPR_LOG_MODULES=openberg:5
```

This will instruct Mozilla to print OpenBerg's error messages to the standard error channel.

If you want to help, to be kept up to date, or for any further inquiry, do not hesitate to visit our website at <http://www.openberg.org>.

There is a list of tasks awaiting taker http://www.openberg.org/how_can_i_help.php

Enjoy !

C.4 Unix build system

What you need to compile OpenBerg under Unix

- Firefox 1.5.x or XULRunner 1.8.0.1

-

C.4.1 Compilation and first installation

To get OpenBerg to compile, you will open file `Makefile.Linux` and change the value of the following variables

`XPIDLXX`

D Licenses

D.1 GNU Public License, version 2

D.1.1 Copyright

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301, USA

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

D.1.3 GNU General Public License

Terms and Conditions For Copying, Distribution and Modification

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - (a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
 - (c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty).

in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a ver-

12. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

End of Terms and Conditions

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what it
does.>
```

```
Copyright (C) <year> <name of author>
```

```
This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License as
published by the Free Software Foundation; either version 2 of the
License, or (at your option) any later version.
```

```
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied war-
ranty of MERCHANTABILITY or FITNESS FOR A PARTICU-
LAR PURPOSE. See the GNU General Public License for more
details.
```

```
You should have received a copy of the GNU General Public
License along with this program; if not, write to the Free Software
Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-
1301, USA.
```

Also add information on how to contact you by electronic and paper

mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) <year> <name of author>  
Gnomovision comes with ABSOLUTELY NO WARRANTY; for  
details type 'show w'.
```

```
This is free software, and you are welcome to redistribute it under  
certain conditions; type 'show c' for details.
```

The hypothetical commands `show w` and `show c` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w` and `show c`; they

a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a

D.2.3 GNU Lesser General Public Licence

Terms and Conditions For Copying, Distribution and Modification

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

A (id)1(luo0(the)-471Libe/dr1(iu6m)-1andr1(iucop)28(y)1(ing.)-317(d)1(istrib)1(uti)1(on)r1(iuan)1(d)-3(i

distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and dis-

tribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- (a) The modified work must itself be a software library.
- (b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.
- (c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.
- (d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-

do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

5. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

- (a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)
- (b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
- (c) Accompany the work with a written offer, valid for at least three

years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

- (d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
- (e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

6. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
 - (a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
 - (b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.
7. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long

system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

11. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
12. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.
13. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

No Warranty

14. Because the program is licensed free of charge, there is no warranty for the program, to the extent permitted by applicable law. Except when otherwise stated in writing the copyright holders and/or other parties provide the program "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of the program is with you. Should the program prove defective, you assume the cost of all necessary

servicing, repair or correction.

15. In no event unless required by applicable law or agreed to in writing will any copyright holder, or any other party who may modify and/or redistribute the program as permitted above, be liable to you for damages, including any general, special, incidental or consequential damages arising out of the use or inability to use the program (including but not limited to loss of data or data being rendered inaccurate or losses sustained by you or third parties or a failure of the program to operate with any other programs), even if such holder or other party has been advised of the possibility of such damages.

End of Terms and Conditions

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it

does (if you can) (for example: "libfoo.so - free software for bar")

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library 'Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That's all there is to it!